Groupe de travail Réseau
Request for Comments : 4271
RFC rendue obsolète : 1771
Catégorie : Sur la voie de la normalisation
Traduction : Claude Brière de L'Isle

Y. Rekhter, éditeur
T. Li, éditeur
S. Hares, éditeur
janvier 2006

Protocole 4 de routeur frontière (BGP-4)

Statut du présent mémoire
La présente RFC spécifie un protocole de normalisation pour la communauté de l'Internet et appelle à des discussions et suggestions pour son amélioration. Prière de se reporter à l'édition en cours des "Internet Official Protocol Standards" (*normes officielles du protocole Internet*) (STD 1) pour connaître l'état de la normalisation et le statut du présent protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent document discute du protocole de routeur frontière (BGP, *Border Gateway Protocol*) qui est un protocole d'acheminement entre systèmes autonomes.

La principale fonction d'un système parlant BGP est d'échanger des informations sur l'accessibilité du réseau avec les autres systèmes BGP. Ces informations d'accessibilité de réseau incluent des informations sur la liste des systèmes autonomes (AS, *Autonomous System*) que traversent les informations d'accessibilité. Ces informations sont suffisantes pour construire un graphe de la connexité des AS en vue de cette accessibilité à partir duquel les boucles d'acheminement peuvent être élaguées, et, au niveau de l'AS, des décisions de politique peuvent être mises en application.

BGP-4 fournit un ensemble de mécanismes pour la prise en charge de l'acheminement inter domaines sans classes (CIDR, *Classless Inter-Domain Routing*). Ces mécanismes incluent la prise en charge de l'annonce d'un ensemble de destinations comme préfixe IP, et éliminent le concept de "classe" de réseau au sein de BGP. BGP-4 introduit aussi des mécanismes qui permettent l'agrégation des chemins, incluant l'agrégation des chemins d'AS.

Le présent document rend obsolète la RFC 1771.

Table des matières

1. Introduction....2
1.1 Définition des termes d'utilisation courante....2
1.2 Spécification des exigences....3
2. Remerciements....3
3. Résumé du fonctionnement....4
3.1 Routes : annonces et mémorisation....5
3.2 Base d'informations d'acheminement....6
4. Formats de message....6
4.1 Format d'en-tête de message....7
4.2 Format du message OPEN....7
4.3 Format du message UPDATE....8
4.4 Format du message KEEPALIVE....11
4.5 Format du message NOTIFICATION....11
5. Attributs de chemin....12
5.1 Utilisation de l'attribut Path....13
6. Traitement d'erreur dans BGP....16
6.1 Traitement d'erreur d'en-tête de message....16
6.2 Traitement d'erreur du message OPEN....17
6.3 Traitement d'erreur du message UPDATE....17
6.4 Traitement d'erreur du message NOTIFICATION....18
6.5 Traitement d'erreur d'expiration du temporisateur de garde....18
6.6 Traitement d'erreur de l'automate à états finis....19
6.7 Cessation....19

6.8 Détection de collision de connexions BGP....19
7. Négociation de la version BGP....19
8. Automate à états finis pour BGP....20
8.1 Événements pour le FSM BGP....21
8.2 Description du FSM....26
9. Traitement du message UPDATE....36
9.1 Processus de décision....37
9.2 Processus Update-Send....41
9.3 Critères de choix de chemin....44
9.4 Générer des chemins BGP....44
10. Temporisateurs BGP....45
11. Considérations pour la sécurité....45
12. Considérations relatives à l'IANA....46
13. Références normatives....47
14. Références pour information....48
Appendice A Comparaison avec la RFC 1771....48
Appendice B Comparaison avec la RFC 1267....49
Appendice C Comparaison avec la RFC 1163....49
Appendice D Comparaison avec la RFC 1105....49
Appendice E Options TCP qui peuvent être utilisées avec BGP....50
Appendice F Recommandations de mise en œuvre....50
F.1 Plusieurs réseaux par message....50
F.2. Réduction du flottement de chemin....50
F.3 Ordre des attributs de chemin....50
F.4 Tri de AS_SET....51
F.5 Contrôle de la négociation de version....51
F.6 Agrégation AS_PATH complexe....51
Adresse des éditeurs....51
Déclaration complète de droits de reproduction....51

1. Introduction

Le protocole de routeur frontière (BGP, *Border Gateway Protocol*) est un protocole d'acheminement entre les systèmes autonomes.

La principale fonction d'un système parlant BGP est d'échanger des informations sur l'accessibilité du réseau avec les autres systèmes BGP. Ces informations d'accessibilité de réseau incluent des informations sur la liste des systèmes autonomes (AS, *Autonomous System*) que traversent les informations d'accessibilité. Ces informations sont suffisantes pour construire un graphe de la connexité des AS en vue de cette accessibilité à partir duquel les boucles d'acheminement peuvent être élaguées, et, au niveau de l'AS, des décisions de politique peuvent être mises en application.

BGP-4 fournit un ensemble de mécanismes pour la prise en charge de l'acheminement inter domaines sans classes (CIDR, *Classless Inter-Domain Routing*) [RFC1518], [RFC1519]. Ces mécanismes incluent la prise en charge de l'annonce d'un ensemble de destinations comme préfixe IP, et éliminent le concept de "classe" de réseau au sein de BGP. BGP-4 introduit aussi des mécanismes qui permettent l'agrégation des chemins, incluant l'agrégation des chemins d'AS.

Les informations d'acheminement échangées via BGP ne prennent en charge que le paradigme de transmission fondée sur la destination, qui suppose qu'un routeur transmet un paquet sur la seule base de l'adresse de destination portée dans l'en-tête IP du paquet. Ceci, à son tour, reflète l'ensemble des décisions de politique qui peuvent (ou non) être mises en application en utilisant BGP. BGP ne peut prendre en charge que les politiques qui se conforment au paradigme de transmission fondée sur la destination.

1.1 Définition des termes d'utilisation courante

Ce paragraphe donne la définition des termes qui ont une signification spécifique pour le protocole BGP et qui sont utilisés tout au long de ce texte.

Adj-RIB-In : contient des informations d'acheminement non traitées qui ont été annoncées au locuteur BGP local par ses homologues.

Adj-RIB-Out : contient les chemins à annoncer à des homologues spécifiques au moyen des messages UPDATE du locuteur BGP local.

Système autonome (AS, *Autonomous System*) : la définition classique d'un système autonome est un ensemble de routeurs sous une seule administration technique, utilisant un protocole de passerelles intérieures (IGP, *Internal Gateway Protocol*) et une métrique commune pour déterminer comment acheminer les paquets au sein de l'AS, et utilisant un protocole d'acheminement inter AS pour déterminer comment acheminer les paquets aux autres AS. Depuis que la définition classique a été développée, il est devenu courant qu'un seul AS utilise plusieurs IGP et, parfois, plusieurs ensembles de métriques au sein d'un AS. L'utilisation du terme "système autonome" souligne le fait que, même quand plusieurs IGP et métriques sont utilisés, l'administration d'un AS apparaît aux autres AS comme ayant un seul plan d'acheminement intérieur cohérent, et présente une image cohérente des destinations qui sont accessibles à travers lui.

Identifiant BGP : entier non signé de 4 octets qui indique l'identifiant BGP de l'envoyeur des messages BGP. Un certain locuteur BGP règle la valeur de son identifiant BGP à une adresse IP allouée à ce locuteur BGP. La valeur de l'identifiant BGP est déterminée au démarrage et est la même pour toute interface locale et tout homologue BGP.

Locuteur BGP : routeur qui met en œuvre BGP.

EBGP (*External BGP*) : connexion BGP entre homologues externes.

Homologue externe : homologue qui est dans un système autonome différent de celui du système local.

Chemin faisable : chemin annoncé qui est disponible pour être utilisé par le receveur.

IBGP *(Internal BGP)* : connexion BGP entre des homologues interne.

Homologue interne : homologue qui est dans le même système autonome que le système local.

IGP *(Interior Gateway Protocol)* : protocole de routeur intérieur ; protocole d'acheminement utilisé pour échanger des informations d'acheminement entre les routeurs au sein d'un seul système autonome.

Loc-RIB : contient les chemins qui ont été choisis par le processus de décision du locuteur BGP local.

NLRI *(Network Layer Accessibility Informations)* : informations d'accessibilité de la couche réseau

Route : unité d'informations qui apparie un ensemble de destinations avec les attributs d'un chemin pour ces destinations. Les ensembles de destinations sont des systèmes dont les adresses IP sont contenues dans un préfixe d'adresse IP porté dans le champ NLRI d'un message UPDATE. La route est l'information rapportée dans le champ Attributs de chemin du même message UPDATE.

RIB *(Routing Information Base)* : base de données d'informations d'acheminement

Route infaisable : route précédemment annoncée comme faisable qui n'est plus disponible à l'utilisation.

1.2 Spécification des exigences

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

2. Remerciements

Le présent document a été à l'origine publié comme [RFC1267] en octobre 1991, dont les auteurs étaient Kirk Lougheed et Yakov Rekhter.

Nous tenons à exprimer nos remerciements à Guy Almes, Len Bosack, et Jeffrey C. Honig pour leurs contributions à la version antérieure (BGP-1) de ce document.

Nous tenons à remercier spécialement Dennis Ferguson de ses nombreuses contributions à la première version de ce document.

Nos remerciements explicites à Bob Braden pour sa relecture de la version antérieure (BGP-2) de ce document, et pour ses commentaires constructifs et précieux.

Nous remercions aussi Bob Hinden, Directeur de l'acheminement du groupe de pilotage de l'ingénierie de l'Internet, et l'équipe de réviseurs qu'il a rassemblé pour relire la version antérieure (BGP-2) de ce document. Cette équipe, consistant en Deborah Estrin, Milo Medin, John Moy, Radia Perlman, Martha Steenstrup, Mike St. Johns, et Paul Tsuchiya, a agi avec une forte combinaison de fermeté, professionnalisme, et courtoisie.

Certaines sections du document ont fait de gros emprunts à IDRP [IS10747], qui est la contrepartie OSI de BGP. Pour cela, le crédit revient au groupe ANSI X3S3.3 présidé par Lyman Chapin et à Charles Kunzinger, qui était l'éditeur IDRP au sein de ce groupe.

Nous remercions aussi Benjamin Abarbanel, Enke Chen, Edward Crabbe, Mike Craren, Vincent Gillet, Eric Gray, Jeffrey Haas, Dimitry Haskin, Stephen Kent, John Krawczyk, David LeRoy, Dan Massey, Jonathan Natale, Dan Pei, Mathew Richardson, John Scudder, John Stewart III, Dave Thaler, Paul Traina, Russ White, Curtis Villamizar, et Alex Zinin de leurs commentaires.

Nous remercions tout spécialement Andrew Lange de son aide dans la préparation de la version finale de ce document.

Finalement, nous remercions tous les membres du groupe de travail IDR de leurs idées et leur soutien apporté à ce document.

3. Résumé du fonctionnement

Le protocole de routeur frontière (BGP, *Border Gateway Protocol*) est un protocole d'acheminement entre les systèmes autonomes. Il est bâti sur l'expérience de EGP (comme défini dans la [RFC904]) et l'usage de EGP dans le cœur de réseau NSFNET (décrit dans les [RFC1092] et [RFC1093]). Pour plus d'informations sur BGP, voir les [RFC1772], [RFC1930], [RFC1997], et [RFC2858].

La principale fonction d'un système de locuteurs BGP est d'échanger des informations d'accessibilité de réseau avec d'autres systèmes BGP. Ces informations d'accessibilité de réseau incluent des informations sur la liste des systèmes autonomes (AS) que traversent ces informations d'accessibilité. Ces informations sont suffisantes pour construire un graphe de la connexité des AS, à partir duquel les boucles d'acheminement peuvent être élaguées, et, au niveau de l'AS, des décisions de politique peuvent être mises en application.

Dans le contexte de ce document, on suppose qu'un locuteur BGP n'annonce à ses homologues que les routes qu'il utilise lui-même. (Dans ce contexte, un locuteur BGP est dit "utiliser" une route BGP si elle est la route BGP préférée et est utilisée pour la transmission). Tous les autres cas sortent du domaine d'application de ce document.

Dans le contexte de ce document, le terme "adresse IP" se réfère à une adresse IPv4 [RFC0791].

Les informations d'acheminement échangées via BGP ne prennent en charge que le paradigme de la transmission fondée sur la destination, qui suppose qu'un routeur transmet un paquet sur la seule base de l'adresse de destination portée dans l'en-tête IP du paquet. Cela, à son tour, reflète l'ensemble des décisions de politique qui peuvent (ou pas) être mises en application à l'aide de BGP. Noter que certaines politiques ne peuvent pas être prises en charge par le paradigme de la transmission fondée sur la destination, et exigent donc des techniques telles que l'acheminement de source (autrement dit l'acheminement explicite). De telles politiques ne peuvent pas non plus être appliquées en utilisant BGP. Par exemple, BGP ne permet pas qu'un AS envoie du trafic à un AS voisin pour le transmettre à une certaine destination (accessible à travers) mais au delà de cet AS voisin, dans l'intention que le trafic prenne un chemin différent de celui pris par le trafic originaire de l'AS voisin (pour cette même destination). Par ailleurs, BGP peut prendre en charge toute politique conforme au paradigme de la transmission fondée sur la destination.

BGP-4 fournit un nouvel ensemble de mécanismes pour prendre en charge l'acheminement inter domaines sans classe (CIDR, *Classless Inter-Domain Routing*) [RFC1518], [RFC1519]. Ces mécanismes incluent de prendre en charge l'annonce d'un ensemble de destinations sous forme d'un préfixe IP et d'éliminer le concept de "classe" de réseau au sein de BGP. BGP-4 introduit aussi des mécanismes qui permettent l'agrégation de routes, incluant l'agrégation de chemins d'AS.

Le présent document utilise tout au long le terme de "système autonome" (AS). La définition classique d'un système autonome est un ensemble de routeurs sous une seule administration technique, utilisant un protocole de passerelle intérieure (IGP, *interior gateway protocol*) et une métrique commune pour déterminer comment acheminer les paquets au sein de l'AS, et en utilisant un protocole d'acheminement inter AS pour déterminer comment acheminer les paquets aux autres AS. Depuis que cette définition classique a été développée, il est devenu courant qu'un seul AS utilise plusieurs IGP et, parfois, plusieurs ensembles de métriques au sein d'un AS. L'utilisation du terme "système autonome" souligne le fait que, même quand plusieurs IGP et métriques sont utilisées, l'administration d'un AS apparaît aux autres AS comme ayant un seul plan cohérent d'acheminement intérieur et présente une image cohérente des destinations qui sont accessibles à travers lui.

BGP utilise TCP [RFC793] comme son protocole de transport. Cela élimine le besoin de mettre en œuvre la fragmentation, retransmission, accusé de réception, et séquencement explicite de mise à jour. BGP écoute sur l'accès TCP 179. Le mécanisme de notification d'erreur utilisé dans BGP suppose que TCP prend en charge une clôture "en douceur" (c'est-à-dire, toutes les données en instance seront livrées avant que la connexion soit close).

Une connexion TCP est formée entre deux systèmes. Ils échangent des messages pour ouvrir la connexion et confirmer ses paramètres.

Le flux de données initial est la portion du tableau d'acheminement BGP qui est permise par la politique d'exportation, appelée la Adj-Ribs-Out (voir en 3.2). Des mises à jour incrémentaires sont envoyées lorsque les tableaux d'acheminement changent. BGP n'exige pas de rafraîchissement périodique du tableau d'acheminement. Pour permettre que les changements de politique locale aient l'effet correct sans avoir à réinitialiser les connexions BGP, un locuteur BGP DEVRAIT soit (a) conserver la version courante des routes qui lui sont annoncées par tous ses homologues pour la durée de la connexion, soit (b) utiliser l'extension de rafraîchissement de chemin [RFC2918].

Les messages KEEPALIVE peuvent être envoyés périodiquement pour assurer que la connexion est active. Les messages NOTIFICATION sont envoyés en réponse aux erreurs ou conditions spéciales. Si une connexion rencontre une condition d'erreur, un message NOTIFICATION est envoyé et la connexion est close.

Un homologue dans un AS différent est appelé un homologue externe, tandis qu'un homologue dans le même AS est appelé un homologue interne. BGP interne et BGP externe sont généralement abrégés respectivement en IBGP et EBGP.

Si un AS particulier a plusieurs locuteurs BGP et fournit un service de transit pour d'autres AS, il faut alors veiller à s'assurer d'une vue cohérente de l'acheminement au sein de l'AS. Une vue cohérente des chemins intérieurs de l'AS est fournie par l'IGP utilisé au sein de l'AS. Pour les besoins du présent document, on suppose qu'une vue cohérente des chemins extérieurs à l'AS est fournie en ayant tous les locuteurs BGP au sein de l'AS qui tiennent IBGP les uns avec les autres.

Le présent document spécifie le comportement de base du protocole BGP. Ce comportement peut être, et est, modifié par des spécifications d'extension. Lorsque le protocole est étendu, le nouveau comportement est pleinement documenté dans les spécifications d'extension.

3.1 Routes : annonces et mémorisation

Pour les besoins du présent protocole, une route est définie comme une unité d'information qui apparie un ensemble de destinations avec les attributs d'un chemin pour ces destinations. L'ensemble des destinations sont des systèmes dont les adresses IP sont contenues dans un préfixe d'adresse IP qui est porté dans le champ Informations d'accessibilité de couche réseau (NLRI, *informations d'accessibilité de couche réseau*) d'un message UPDATE, et les routes sont les informations rapportées dans le champ Attributs de chemin dans le même message UPDATE.

Les routes sont annoncées entre les locuteurs BGP dans les messages UPDATE. Plusieurs routes qui ont les mêmes attributs de chemin peuvent être annoncées dans un seul message UPDATE en incluant plusieurs préfixes dans le champ NLRI du message UPDATE.

Les routes sont mémorisées dans les bases de données d'informations d'acheminement (RIB, *Routing Information Base*) : à savoir, Adj-RIBs-In, Loc-RIB, et Adj-RIBs-Out, comme décrit au paragraphe 3.2.

Si un locuteur BGP choisit d'annoncer une route reçue précédemment, il PEUT y ajouter, ou modifier, les attributs de chemin de la route avant de l'annoncer à un homologue.

BGP fournit des mécanismes par lesquels un locuteur BGP peut informer ses homologues qu'une route annoncée précédemment n'est plus disponible. Il y a trois méthodes par lesquelles un certain locuteur BGP peut indiquer qu'une route a été retirée du service :

a) le préfixe IP qui exprime la destination pour une route annoncée précédemment peut être annoncé dans le champ Routes retirées dans le message UPDATE, marquant ainsi que la route associée n'est plus disponible à l'utilisation,

b) une route de remplacement avec les mêmes NLRI peut être annoncée, ou

c) la connexion du locuteur BGP peut être close, ce qui retire implicitement du service toutes les routes que la paire de locuteurs s'étaient annoncés l'un à l'autre.

Changer les attributs d'une route se fait en annonçant une route de remplacement. La route de remplacement porte les nouveaux attributs (changés) et a le même préfixe d'adresse que la route originale.

3.2 Base d'informations d'acheminement

La base de données d'informations d'acheminement (RIB) au sein d'un locuteur BGP consiste en trois parties distinctes :

a) Adj-RIBs-In : elle mémorise les informations d'acheminement apprises des messages UPDATE entrants qui ont été reçus d'autres locuteurs BGP. leur contenu représente des routes disponibles comme entrée au processus de décision.

b) Loc-RIB : contient les informations d'acheminement locales que le locuteur BGP a choisies en appliquant ses politiques locales aux informations d'acheminement contenues dans sa Adj-RIBs-In. Ce sont les routes qui seront utilisées par le locuteur BGP local. Le prochain bond pour chacune de ces routes DOIT pouvoir être résolu via le tableau d'acheminement du locuteur BGP local.

c) Adj-RIBs-Out : mémorise les informations que le locuteur BGP local a choisies pour les annoncer à ses homologues. Les informations d'acheminement mémorisées dans la Adj-RIBs-Out seront portées dans les messages UPDATE du locuteur BGP local et annoncées à ses homologues.

En résumé, la Adj-RIBs-In contient des informations d'acheminement non traitées qui ont été annoncées au locuteur BGP local par ses homologues ; la Loc-RIB contient les routes qui ont été choisies par le processus de décision du locuteur BGP local ; et la Adj-RIBs-Out organise les routes pour les annoncer à des homologues spécifiques (au moyen des messages UPDATE du locuteur local).

Bien que le modèle conceptuel distingue entre Adj-RIBs-In, Loc-RIB, et Adj-RIBs-Out, cela n'implique ni n'exige qu'une mise en œuvre doive tenir trois copies séparées des informations d'acheminement. Le choix de mise en œuvre (par exemple, trois copies des informations ou une copie avec des pointeurs) n'est pas imposé par le protocole.

Les informations d'acheminement que le locuteur BGP utilise pour transmettre les paquets (ou pour construire le tableau de transmissions utilisé pour la transmission des paquets) sont tenues dans le tableau d'acheminement. Le tableau d'acheminement accumule les routes pour les réseaux directement connectés, les routes statiques, les routes apprises des protocoles d'IGP, et les routes apprises de BGP. Qu'une route BGP spécifique devrait être installée dans le tableau d'acheminement, et qu'une route BGP devrait écraser une route pour la même destination installée par une autre source, est une décision de politique locale, et n'est pas spécifié dans ce document. En plus de la transmission réelle des paquets, le tableau d'acheminement est utilisé pour la résolution des adresses de prochain bond spécifiées dans les mises à jour BGP (voir le paragraphe 5.1.3).

4. Formats de message

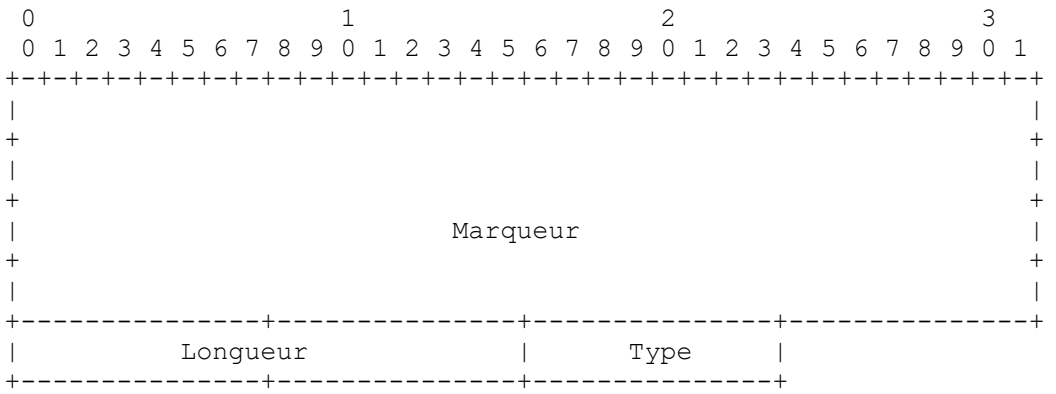
Cette section décrit les formats de message utilisés par BGP.

Les messages BGP sont envoyés sur des connexions TCP. Un message n'est traité qu'après être entièrement reçu. La taille maximum de message est de 4096 octets. Toutes les mises en œuvre sont obligées de prendre en charge cette taille maximum de message. Le plus petit message qui peut être envoyé consiste en un en-tête BGP sans portion de données (19 octets).

Tous les champs multi octets sont dans l'ordre des octets du réseau.

4.1 Format d'en-tête de message

Chaque message a un en-tête de taille fixe. Il peut y avoir, ou pas, une portion de données qui suit l'en-tête, selon le type de message. La disposition de ces champs figure ci-dessous :



Marqueur : ce champ de 16 octets est inclus pour la compatibilité ; il DOIT être réglé tout de uns.

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 le prochain message (son champ Marqueur) dans le flux TCP. La valeur du champ Longueur DOIT toujours être d'au moins 19 et pas supérieure à 4096, et PEUT être plus contrainte, selon le type de message. Le "bourrage" de données supplémentaires après le message n'est pas permis. Donc, le champ Longueur DOIT avoir la plus petite valeur requise, étant donné le reste du message.

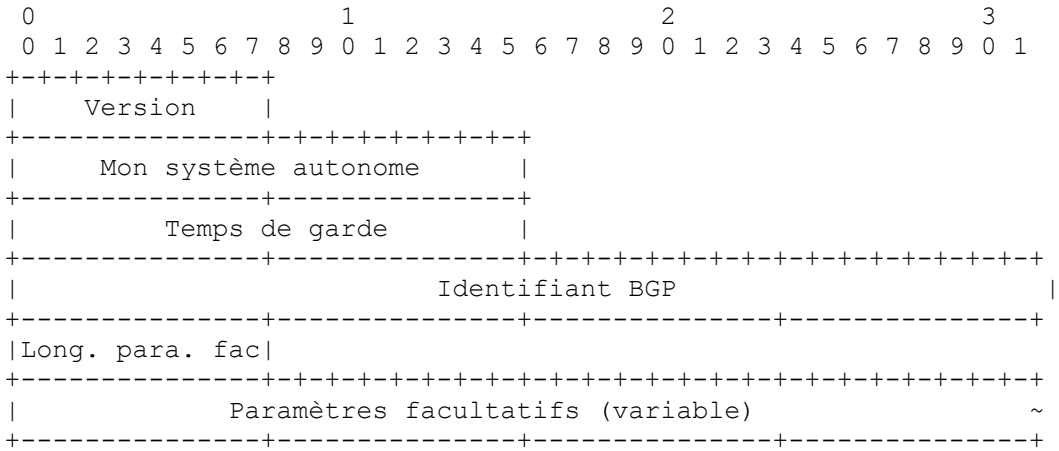
Type : cet entier non signé de 1 octet indique le code de type du message. Les codes de type suivants sont définis :
1 : OPEN *(ouverture)*
2 : UPDATE *(mise à jour)*
3 : NOTIFICATION *(notification)*
4 : KEEPALIVE *(garder en vie)*

La [RFC2918] définit un code de type de plus : 5 - ROUTE-REFRESH.

4.2 Format du message OPEN

Après l'établissement d'une connexion TCP, le premier message envoyé par chaque côté est un message OPEN. Si le message OPEN est acceptable, un message KEEPALIVE confirmant le OPEN est renvoyé.

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



Version : entier non signé de un octet qui indique le numéro de version de protocole du message. Le numéro de version actuel de BGP est 4.

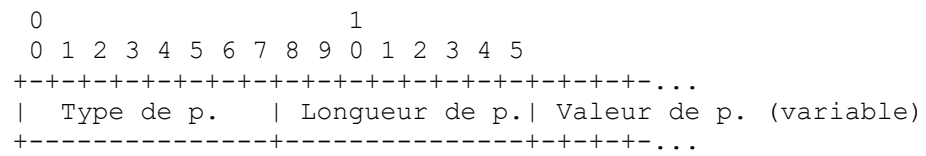
Mon système autonome : cet entier non signé de 2 octets indique le numéro de système autonome de l'envoyeur.

Temps de garde : cet entier non signé de 2 octets indique le nombre de secondes que propose l'envoyeur pour la valeur du temporisateur de garde *(Hold Timer)*. À réception d'un message OPEN, un locuteur BGP DOIT calculer la valeur du temporisateur de garde en utilisant le plus petite de son temps de garde configuré et du temps de garde reçu dans le message OPEN. 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 peuvent s'écouler entre la réception de messages successifs KEEPALIVE et/ou UPDATE provenant de l'envoyeur.

Identifiant BGP : cet entier non signé de 4 octets indique l'identifiant BGP de l'envoyeur. Un certain locuteur BGP règle la valeur de son identifiant BGP à une adresse IP qui est allouée à ce locuteur BGP. La valeur de l'identifiant BGP est déterminée au démarrage et est la même pour chaque interface locale et homologue BGP.

Longueur des paramètres facultatifs : cet entier non signé de 1 octet 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 contient une liste de paramètres facultatifs, dans lequel chaque paramètre est codé comme un triplet <Type de paramètre, Longueur de paramètre, Valeur de paramètre>.



Type de paramètre est un champ d'un octet qui identifie sans ambiguïté les paramètres individuels. Longueur de paramètre est un champ de un octet qui contient la longueur du champ Valeur de paramètre en octets. Valeur de paramètre est un champ de longueur variable qui est interprété conformément à la valeur du champ Type de paramètre.

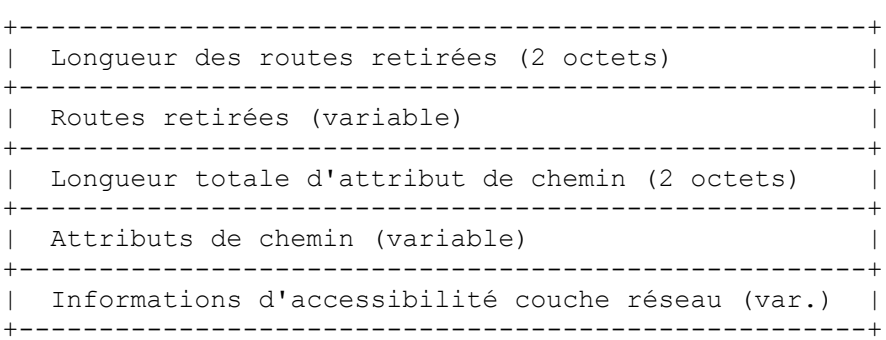
La [RFC3392] définit le paramètre facultatif Capacités.

La longueur minimum du message OPEN est 29 octets (incluant l'en-tête du message).

4.3 Format du message UPDATE

Les messages UPDATE sont utilisés pour transférer les informations d'acheminement entre homologues BGP. Les informations dans le message UPDATE peuvent être utilisées pour construire un graphe qui décrit les relations des divers systèmes autonomes. Par l'application des règles à discuter, les boucles d'informations d'acheminement et certaines autres anomalies peuvent être détectées et retirées de l'acheminement inter AS.

Un message UPDATE est utilisé pour annoncer les routes faisables qui partagent des attributs de chemin communs avec un homologue, ou pour retirer du service plusieurs routes infaisables (voir le paragraphe 3.1). Un message UPDATE PEUT simultanément annoncer une route faisable et retirer du service plusieurs routes infaisables. Le message UPDATE inclut toujours l'en-tête BGP de taille fixe, et inclut aussi d'autres champs, comme montré ci-dessous (noter que certains des champs montrés peuvent ne pas être présents dans tous les messages UPDATE) :



Longueur des routes retirées : cet entier non signé de deux octets indique la longueur totale du champ Routes retirées en octets. Sa valeur permet de déterminer la longueur du champ Informations d'accessibilité de couche réseau, comme spécifié ci-dessous.

Une valeur de 0 indique qu'aucune route n'est retirée du service, et que le champ Routes retirées n'est pas présent dans ce message UPDATE.

Routes retirées : c'est un champ de longueur variable qui contient une liste de préfixes d'adresses IP pour les routes qui sont retirées du service. Chaque préfixe d'adresse IP est codé comme un doublet de la forme <longueur, préfixe>, dont les champs sont décrits ci-dessous :

```
+---------------------------+
|   Longueur (1 octet)      |
+---------------------------+
|   Préfixe (variable)      |
+---------------------------+
```

L'utilisation et la signification de ces champs sont :

a) Longueur : le champ Longueur indique la longueur en bits du préfixe d'adresse IP. Une longueur de zéro indique un préfixe qui correspond à toutes les adresses IP (avec un préfixe qui est lui-même de zéro octet).

b) Préfixe : le champ Préfixe contient un préfixe d'adresse IP, suivi par le nombre minimum de bits en queue nécessaire pour faire tomber la fin du champ sur une limite d'octet. Noter que la valeur des bits en queue est sans importance.

Longueur totale d'attribut de chemin : cet entier non signé de deux octets indique la longueur totale du champ Attributs de chemin en octets. Sa valeur permet de déterminer la longueur du champ Informations d'accessibilité de couche réseau comme spécifié ci-dessous. Une valeur de 0 indique que ni le champ Informations d'accessibilité de couche réseau ni le champ Attributs de chemin ne sont présents dans ce message UPDATE.

Attributs de chemin : séquence de longueur variable d'attributs de chemin présente dans chaque message UPDATE, sauf pour un message UPDATE qui porte seulement les routes retirées. Chaque attribut de chemin est un triplet <type d'attribut, longueur d'attribut, valeur d'attribut> de longueur variable.

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

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Fanions d'att.|Code type attr.|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Le bit de poids fort (bit 0) de l'octet Fanions d'attribut est le bit Facultatif. Il définit si l'attribut est facultatif (réglé à 1) ou bien connu (réglé à 0).

Le second bit de poids fort (bit 1) de l'octet Fanions d'attribut est le bit Transitif. Il définit si un attribut facultatif est transitif (réglé à 1) ou non transitif (réglé à 0). Pour les attributs bien connus, le bit Transitif DOIT être réglé à 1 (voir à la Section 5 une discussion sur les attributs transitifs).

Le troisième bit de poids fort (bit 2) de l'octet Fanions d'attribut est le bit Partiel. Il définit si les informations contenues dans l'attribut transitif facultatif sont partielle (réglé à 1) ou complètes (réglé à 0). Pour les attributs bien connus et pour les attributs non transitifs facultatifs, le bit Partiel DOIT être réglé à 0.

Le quatrième bit de poids fort (bit 3) de l'octet Fanions d'attribut est le bit Longueur étendue. Il définit si la longueur d'attribut fait un octet (réglé à 0) ou deux octets (réglé à 1).

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

L'octet Code de type d'attribut contient le code de type d'attribut. Les codes de type d'attribut actuellement définis sont présentés à la Section 5.

Si le bit Longueur étendue de l'octet Fanions d'attribut est réglé à 0, le troisième octet de l'attribut de chemin contient la longueur des données d'attribut en octets.

Si le bit Longueur étendue de l'octet Fanions d'attribut est réglé à 1, le troisième et le quatrième octet de l'attribut de chemin contiennent la longueur des données de l'attribut en octets.

Les octets restants de l'attribut de chemin représentent la valeur de l'attribut et sont interprétés conformément aux fanions d'attribut et au code de type d'attribut. Les codes de type d'attribut, leurs valeurs et utilisations d'attribut, sont les suivants :

a) ORIGINE (code de type 1) : ORIGINE est un attribut obligatoire bien connu qui définit l'origine des systèmes autonomes. L'octet de données peut prendre les valeurs suivantes :

Valeur	**Signification**
0	IGP - les informations d'accessibilité de couche réseau intérieures à l'AS d'origine
1	EGP - les informations d'accessibilité de couche réseau apprises via EGP [RFC904]
2	INCOMPLETE - informations d'accessibilité de couche réseau apprise par d'autres moyens

L'usage de cet attribut est défini au paragraphe 5.1.1.

b) AS_PATH (Code de type 2) : AS_PATH est un attribut obligatoire bien connu qui se compose d'une séquence de segments de chemin d'AS. Chaque segment de chemin d'AS est représenté par un triplet <type de segment de chemin, longueur de segment de chemin, valeur de segment de chemin>.

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

Valeur	**Type de segment**
1	AS_SET : ensemble non ordonné d'AS qu'une route a traversé dans le message UPDATE.
2	AS_SEQUENCE : ensemble ordonné d'AS qu'une route a traversé dans le message UPDATE.

La longueur de segment de chemin est un champ de longueur de un octet, qui contient le nombre d'AS (pas le nombre d'octets) dans le champ Valeur de segment de chemin.

Le champ Valeur de segment de chemin contient un numéro d'AS ou plus, chacun étant codé comme un champ de longueur de deux octets. L'usage de cet attribut est défini au paragraphe 5.1.2.

c) NEXT_HOP (code de type 3) : c'est un attribut obligatoire bien connu qui définit l'adresse IP (envoi individuel) du routeur qui DEVRAIT être utilisée comme prochain bond pour les destinations dont la liste figure dans le champ Informations d'accessibilité de couche réseau du message UPDATE. L'usage de cet attribut est défini au paragraphe 5.1.3.

d) MULTI_EXIT_DISC (code de type 4) : c'est un attribut facultatif non transitif qui est un entier non signé de quatre octets. La valeur de cet attribut PEUT être utilisée par le processus de décision d'un locuteur BGP pour faire la différence entre plusieurs points d'entrée à un système autonome voisin. L'usage de cet attribut est défini en 5.1.4.

e) LOCAL_PREF (code de type 5) : c'est un attribut bien connu qui est un entier non signé de quatre octets. Un locuteur BGP l'utilise pour informer ses autres homologues internes du degré de préférence du locuteur annonceur pour une route annoncée. L'usage de cet attribut est défini au paragraphe 5.1.5.

f) ATOMIC_AGGREGATE (code de type 6) : c'est un attribut discrétionnaire bien connu de longueur 0. L'usage de cet attribut est défini au paragraphe 5.1.6.

g) AGGREGATOR (code de type 7) : c'est un attribut facultatif transitif de longueur 6. L'attribut contient le dernier numéro d'AS qui a formé la route agrégée (codé sur 2 octets) suivi par l'adresse IP du locuteur BGP qui a formé la route agrégée (codée sur 4 octets). Ce DEVRAIT être la même adresse que celle utilisée pour l'identifiant BGP du locuteur. L'usage de cet attribut est défini au paragraphe 5.1.7.

Informations d'accessibilité de couche réseau : ce champ de longueur variable contient une liste de préfixes d'adresses IP. La longueur, en octets, des informations d'accessibilité de couche réseau n'est pas codée explicitement, mais peut être calculée par : Longueur de message UPDATE - 23 - Longueur totale d'attributs de chemin - Longueur des routes retirées, où "Longueur de message UPDATE" est la valeur codée dans l'en-tête BGP de taille fixe, "Longueur totale d'attributs de chemin", et "Longueur des routes retirées" sont les valeurs codées dans la partie variable du message UPDATE, et 23 est une longueur combinée de l'en-tête BGP de taille fixe, du champ Longueur totale d'attribut de chemin, et du champ Longueur des routes retirées.

Les informations d'accessibilité sont codées comme un ou plusieurs doublets de forme <longueur, préfixe>, dont les champs sont décrits ci-dessous :

```
+-+-+-+-+-+-+-+-+
| Longueur      | (1 octet)
+---------------+
| Préfixe       | (variable)
+---------------+
```

L'usage et la signification de ces champs sont comme suit :

a) Longueur : le champ Longueur indique la longueur en bits du préfixe d'adresse IP. Une longueur de zéro indique un préfixe qui correspond pour toutes les adresses IP (avec le préfixe ayant lui-même zéro octet).

b) Préfixe : le champ Préfixe contient un préfixe d'adresse IP, suivi par assez de bits en queue pour faire tomber la fin du champ sur une limite d'octet. Noter que la valeur des bits en queue est sans importance.

La longueur minimum du message UPDATE est de 23 octets -- 19 octets pour l'en-tête fixe + 2 octets pour la longueur des routes retirées + 2 octets pour la longueur totale d'attributs de chemin (la valeur de la longueur des routes retirées est 0 et la valeur de la longueur totale d'attributs de chemin est 0).

Un message UPDATE peut annoncer, au plus, un ensemble d'attributs de chemin, mais plusieurs destinations, pourvu que les destinations partagent ces attributs. Tous les attributs de chemin contenus dans un certain message UPDATE s'appliquent à toutes les destinations portées dans le champ NLRI du message UPDATE.

Un message UPDATE peut faire la liste de plusieurs routes qui sont à retirer du service. Chacune de ces routes est identifiée par sa destination (exprimée par un préfixe IP) qui identifie sans ambiguïté la route dans le contexte du locuteur BGP - connexion de locuteur BGP à laquelle il a été précédemment annoncé.

Un message UPDATE ne peut annoncer que des routes qui sont à retirer du service, auquel cas le message ne va pas inclure les attributs de chemin ou les informations d'accessibilité de couche réseau. À l'inverse, il ne peut annoncer qu'une route faisable, et dans ce cas le champ Routes retirées n'a pas besoin d'être présent.

Un message UPDATE NE DEVRAIT PAS inclure le même préfixe d'adresse dans les champs Routes retirées et Informations d'accessibilité de couche réseau. Cependant, un locuteur BGP DOIT être capable de traiter les messages UPDATE de cette forme. Un locuteur BGP DEVRAIT traiter un message UPDATE de cette forme même si le Routes retirées ne contient pas le préfixe d'adresse.

4.4 Format du message KEEPALIVE

BGP n'utilise aucun mécanisme de garde en vie fondé sur TCP pour déterminer si les homologues sont joignables. Les messages KEEPALIVE sont plutôt échangés entre les homologues assez souvent pour ne pas laisser le temporisateur de garde arriver à expiration. Un délai maximum raisonnable entre messages KEEPALIVE serait d'un tiers de l'intervalle de temps de garde. Les messages KEEPALIVE NE DOIVENT PAS être envoyés plus fréquemment que un par seconde. Une mise en œuvre PEUT ajuster le taux d'envoi des messages KEEPALIVE en fonction de l'intervalle de temps de garde. Si l'intervalle de temps de garde négocié est zéro, des messages KEEPALIVE périodiques NE DOIVENT PAS être envoyés.

Un message KEEPALIVE consiste seulement en l'en-tête de message et a une longueur de 19 octets.

4.5 Format du message NOTIFICATION

Un message NOTIFICATION est envoyé quand une condition d'erreur est détectée. La connexion BGP est close immédiatement après qu'il a été envoyé.

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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Code d'erreur | Sous code     |   Données (variable)          |
+---------------+---------------+---------------+---------------+
```

Code d'erreur : entier non signé d'un octet qui indique le type de NOTIFICATION. Les codes d'erreur suivants sont définis :

Code d'erreur	Nom symbolique	Référence
1	Erreur d'en-tête de message	paragraphe 6.1
2	Erreur de message OPEN	paragraphe 6.2
3	Erreur de message UPDATE	paragraphe 6.3
4	Temporisateur de garde expiré	paragraphe 6.5
5	Erreur de FSM	paragraphe 6.6
6	Cessation	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 rapportée. Chaque code d'erreur peut avoir un ou plusieurs sous codes d'erreur associés. Si aucun sous code d'erreur approprié n'est défini, une valeur de zéro (non spécifique) est alors utilisée pour le champ Sous code d'erreur.

Sous codes d'erreur d'en-tête de message :
1 - Connexion non synchronisée.
2 - Mauvaise longueur de message.
3 - Mauvais type de message.

Sous codes d'erreur de message OPEN :
1 - Numéro de version non pris en charge.
2 - Mauvais AS homologue.
3 - Mauvais identifiant BGP.
4 - Paramètre facultatif non pris en charge.
5 - [Déconseillé - voir l'Appendice A].
6 - Temps de garde inacceptable.
7 - Capacité non prise en charge

Sous codes d'erreur de message UPDATE :
1 - Liste d'attributs mal formée.
2 - Attribut bien connu non reconnu.
3 - Attribut bien connu manquant.
4 - Erreur des fanions d'attribut.
5 - Erreur de longueur d'attribut.
6 - Attribut ORIGINE invalide.
7 - [Déconseillé - voir l'Appendice A].
8 - Attribut NEXT_HOP invalide.
9 - Erreur d'attribut facultatif.
10 - Champ Réseau invalide.
11 - AS_PATH mal formé.

Données : ce champ de longueur variable est utilisé pour diagnostiquer la raison de la NOTIFICATION. Le contenu du champ Données dépend du code et sous code d'erreur. Voir plus de détails à la Section 6.

Noter que la longueur du champ Données peut être déterminée à partir du champ Longueur de message par la formule :

Longueur de message = 21 + Longueur des données

La longueur minimum du message NOTIFICATION est 21 octets (incluant l'en-tête de message).

5. Attributs de chemin

Cette section discute des attributs de chemin du message UPDATE.

Les attributs de chemin entrent dans quatre catégories :
1. bien connu obligatoire,
2. bien connu discrétionnaire,
3. facultatif transitif,
4. facultatif non transitif.

Les mises en œuvre de BGP DOIVENT reconnaître tous les attributs bien connus. Certains de ces attributs sont obligatoires et DOIVENT être inclus dans tout message UPDATE qui contient des NLRI. D'autres sont discrétionnaires et PEUVENT ou non être envoyés sur un message UPDATE particulier.

Une fois qu'un homologue BGP a mis à jour des attributs bien connus, il DOIT passer ces attributs à ses homologues dans toute mise à jour qu'il transmet.

En plus des attributs bien connus, chaque chemin PEUT contenir un ou plusieurs attributs facultatifs. Il n'est pas exigé ni attendu de toutes les mises en œuvre de BGP qu'elles prennent en charge tous les attributs facultatifs. Le traitement d'un attribut facultatif non reconnu est déterminé par le réglage du bit Transitif dans l'octet Fanions d'attribut. Les chemins avec des attributs facultatifs transitifs non reconnus DEVRAIENT être acceptés. Si un chemin avec un attribut facultatif transitif non reconnu est accepté et passé aux autres homologues BGP, l'attribut facultatif transitif non reconnu de ce chemin DOIT être passé, avec le chemin, aux autres homologues BGP avec le bit Partiel dans l'octet Fanions d'attribut réglé à 1. Si un chemin avec un attribut facultatif transitif reconnu est accepté et passé aux autres homologues BGP et si le bit Partiel dans l'octet Fanions d'attributs est réglé à 1 par certains AS antérieurs, il NE DOIT PAS être remis à 0 par l'AS courant. Les attributs facultatifs non transitifs non reconnus DOIVENT être ignorés en silence et ne pas être passés aux autres homologues BGP.

De nouveaux attributs transitifs facultatifs PEUVENT être rattachés au chemin par le générateur ou par tout autre locuteur BGP sur le chemin. Si ils ne sont pas rattachés par le générateur, le bit Partiel dans l'octet Fanions d'attribut est réglé à 1. Les règles pour le rattachement de nouveaux attributs non transitifs facultatifs vont dépendre de la nature de l'attribut spécifique. La documentation de chaque nouvel attribut non transitif facultatif sera supposée inclure de telles règles (la description de l'attribut MULTI_EXIT_DISC en donne un exemple). Tous les attributs facultatifs (transitifs et non transitifs) PEUVENT être mis à jour (si approprié) par les locuteurs BGP sur le chemin.

L'envoyeur d'un message UPDATE DEVRAIT ordonner les attributs de chemin au sein du message UPDATE en ordre ascendant d type d'attribut. Le receveur d'un message UPDATE DOIT être prêt à traiter les attributs de chemin au sein de messages UPDATE qui ne sont pas dans l'ordre.

Le même attribut (attribut du même type) ne peut pas apparaître plus d'une fois au sein du champ Attributs de chemin d'un message UPDATE particulier.

La catégorie obligatoire se réfère à un attribut qui DOIT être présent dans les échanges IBGP et EBGP si des NLRI sont contenues dans le message UPDATE. Les attributs classés facultatifs pour les besoins du mécanisme d'extension de protocole peuvent être purement discrétionnaires, discrétionnaires, exigés, ou interdits dans certains contextes.

attribut	**EBGP**	**IBGP**
ORIGIN	obligatoire	obligatoire
AS_PATH	obligatoire	obligatoire
NEXT_HOP	obligatoire	obligatoire
MULTI_EXIT_DISC	discrétionnaire	discrétionnaire
LOCAL_PREF	voir § 5.1.5	exigé
ATOMIC_AGGREGATE	voir § 5.1.6 et 9.1.4	
AGGREGATOR	discrétionnaire	discrétionnaire

5.1 Utilisation de l’attribut Path

L'utilisation de chaque attribut de chemin BGP est décrite dans les sous paragraphes qui suivent.

5.1.1 ORIGIN

ORIGIN est un attribut obligatoire bien connu. L'attribut ORIGIN est généré par le locuteur qui génère les informations d'acheminement associées. Sa valeur NE DEVRAIT PAS être changée par un autre locuteur.

5.1.2 AS_PATH

AS_PATH est un attribut obligatoire bien connu. Cet attribut identifie les systèmes autonomes à travers lesquels sont passées les informations d'acheminement portées dans ce message UPDATE. Les composants de cette liste peuvent être des AS_SET ou des AS_SEQUENCE.

Quand un locuteur BGP propage une route qu'il a apprise du message UPDATE d'un autre locuteur BGP, il modifie l'attribut AS_PATH de la route sur la base de la localisation du locuteur BGP auquel la route va être envoyée :

a) Quand un certain locuteur BGP annonce la route à un homologue interne, le locuteur qui annonce NE DEVRA PAS modifier l'attribut AS_PATH associé à la route.
b) Quand un certain locuteur BGP annonce la route à un homologue externe, le locuteur qui annonce met à jour l'attribut AS_PATH comme suit :
   1) si le premier segment de chemin de l'AS_PATH est du type AS_SEQUENCE, le système local ajoute son propre numéro d'AS comme dernier élément de la séquence (le met dans la position la plus à gauche par rapport à la position des octets dans le message de protocole). Si l'ajout doit causer un débordement du segment AS_PATH (c'est-à-dire, plus de 255 AS) il DEVRAIT ajouter un nouveau segment de type AS_SEQUENCE et ajouter son propre numéro d'AS à ce nouveau segment.
   2) si le premier segment de chemin de l'AS_PATH est du type AS_SET, le système local ajoute un nouveau segment de chemin de type AS_SEQUENCE à l'AS_PATH, incluant son propre numéro d'AS dans ce segment.
   3) si le AS_PATH est vide, le système local crée un segment de chemin de type AS_SEQUENCE, place son propre AS dans ce segment, et place ce segment dans le AS_PATH.

Quand un locuteur BGP génère une route, alors :

a) le locuteur générateur inclut son propre numéro d'AS dans un segment de chemin, de type AS_SEQUENCE, dans l'attribut AS_PATH de tous les messages UPDATE envoyés à un homologue externe. Dans ce cas, le numéro d'AS du système autonome du locuteur générateur va être la seule entrée du segment de chemin, et ce segment de chemin va être le seul segment dans l'attribut AS_PATH.
b) le locuteur générateur inclut un attribut AS_PATH vide dans tous les messages UPDATE envoyés à ses homologues internes. (Un attribut AS_PATH vide a un champ Longueur qui contient la valeur zéro).

Chaque fois que la modification de l'attribut AS_PATH appelle à l'inclusion ou l'ajout du numéro d'AS du système local, le système local PEUT inclure/ajouter plus d'une instance de son propre numéro d'AS dans l'attribut AS_PATH. C'est contrôlé via la configuration locale.

5.1.3 NEXT_HOP

NEXT_HOP est un attribut obligatoire bien connu qui définit l'adresse IP du routeur qui DEVRAIT être utilisé comme prochain bond pour les destinations mentionnées dans le message UPDATE. L'attribut NEXT_HOP est calculé comme suit :

1) Lors de l'envoi d'un message à un homologue interne, si la route n'est pas générée en local, le locuteur BGP NE DEVRAIT PAS modifier l'attribut NEXT_HOP sauf si il a été explicitement configuré à annoncer sa propre adresse IP comme prochain bond. Lorsque il annonce une route générée en local à un homologue interne, le locuteur BGP DEVRAIT utiliser l'adresse d'interface du routeur par lequel le réseau annoncé est accessible pour le locuteur comme prochain bond. Si la route est directement connectée au locuteur, ou si l'adresse d'interface du routeur à travers lequel le réseau annoncé est accessible pour le locuteur est l'adresse de l'homologue interne, alors le locuteur BGP DEVRAIT utiliser sa propre adresse IP pour l'attribut NEXT_HOP (l'adresse de l'interface qui est utilisé pour atteindre l'homologue).

2) Lors de l'envoi d'un message à un homologue externe, X, et quand l'homologue est à un bond IP du locuteur :
   - Si la route annoncée a été apprise d'un homologue interne ou est générée en local, le locuteur BGP peut utiliser une adresse d'interface du routeur homologue interne (ou le routeur interne) à travers lequel le réseau annoncé est accessible pour le locuteur pour l'attribut NEXT_HOP, pourvu que l'homologue X partage un sous réseau commun avec cette adresse. C'est une forme d'attribut NEXT_HOP "tiers".
   - Autrement, si la route annoncée a été apprise d'un homologue externe, le locuteur peut utiliser une adresse IP de tout routeur adjacent (connu par l'attribut NEXT_HOP reçu) que le locuteur lui-même utilise pour le calcul de routes locales dans l'attribut NEXT_HOP, pourvu que cet homologue X partage un sous réseau commun avec cette adresse. C'est une seconde forme d'attribut NEXT_HOP "tiers".
   - Autrement, si l'homologue externe auquel la route est annoncée partage un sous réseau commun avec une des interfaces du locuteur BGP qui annonce, le locuteur PEUT utiliser l'adresse IP associée à une telle interface dans l'attribut NEXT_HOP. C'est ce qu'on appelle un attribut NEXT_HOP de "première partie".
   - Par défaut (si aucune des conditions ci-dessus ne s'applique) le locuteur BGP DEVRAIT utiliser l'adresse IP de l'interface que le locuteur utilise pour établir la connexion BGP avec l'homologue X dans l'attribut NEXT_HOP.

3) Lors de l'envoi d'un message à un homologue externe X, et lorsque l'homologue est à plusieurs bonds IP du locuteur (autrement dit "EBGP multi bonds") :

- Le locuteur PEUT être configuré à propager l'attribut NEXT_HOP. Dans ce cas, en annonçant une route que le locuteur a apprise d'un de ses homologues, l'attribut NEXT_HOP de la route annoncée est exactement le même que l'attribut NEXT_HOP de la route apprise (le locuteur ne modifie pas l'attribut NEXT_HOP).
- Par défaut, le locuteur BGP DEVRAIT utiliser l'adresse IP de l'interface que le locuteur utilise dans l'attribut NEXT_HOP pour établir la connexion BGP avec l'homologue X.

Normalement, l'attribut NEXT_HOP est choisi de façon telle que soit pris le plus court chemin disponible. Un locuteur BGP DOIT être capable de prendre en charge l'annonce de désactivation des attributs NEXT_HOP "tiers" afin de traiter les supports imparfaitement pontés.

Une route générée par un locuteur BGP NE DEVRA PAS être annoncée à un homologue en utilisant une adresse de cet homologue comme NEXT_HOP. Un locuteur BGP NE DEVRA PAS installer une route avec lui-même comme prochain bond.

L'attribut NEXT_HOP est utilisé par le locuteur BGP pour déterminer l'interface de sortie réelle et l'adresse du prochain bond immédiat qui DEVRAIENT être utilisées pour transmettre les paquets en transit aux destinations associées.

L'adresse du prochain bond immédiat est déterminée en effectuant une opération de recherche de route récurrente pour l'adresse IP dans l'attribut NEXT_HOP, en utilisant le contenu du tableau d'acheminement, en choisissant une entrée si plusieurs entrées de coût égal existent. L'entrée de tableau d'acheminement qui résout l'adresse IP dans l'attribut NEXT_HOP va toujours spécifier l'interface sortante. Si l'entrée spécifie un sous réseau rattaché, mais ne spécifie pas une adresse de prochain bond, alors l'adresse qui est dans l'attribut NEXT_HOP DEVRAIT être utilisé comme adresse de prochain bond immédiat. Si l'entrée spécifie aussi l'adresse de prochain bond, cette adresse DEVRAIT être utilisée comme adresse de prochain bond immédiat pour la transmission de paquets.

5.1.4 MULTI_EXIT_DISC

MULTI_EXIT_DISC est un attribut facultatif non transitif qui est destiné à être utilisé sur des liaisons externes (inter AS) pour différencier plusieurs points de sortie ou d'entrée au même AS du voisinage. La valeur de l'attribut MULTI_EXIT_DISC est un nombre non signé de quatre octets, appelé une métrique. Toutes choses égales par ailleurs, le point de sortie avec la plus faible métrique DEVRAIT être préféré. S'il est reçu sur EBGP, l'attribut MULTI_EXIT_DISC PEUT être propagé sur IBGP aux autres locuteurs BGP au sein du même AS (voir aussi au paragraphe 9.1.2.2). L'attribut MULTI_EXIT_DISC reçu d'un AS du voisinage NE DOIT PAS être propagé aux autres AS du voisinage.

Un locuteur BGP DOIT mettre en œuvre un mécanisme (fondé sur la configuration locale) qui permette que l'attribut MULTI_EXIT_DISC soit retiré d'une route. Si un locuteur BGP est configuré à retirer l'attribut MULTI_EXIT_DISC d'une route, ce retrait DOIT être fait avant de déterminer le degré de préférence des routes et avant d'effectuer le choix de route (phases 1 et 2 du processus de décision).

Une mise en œuvre PEUT aussi (sur la base de la configuration locale) altérer la valeur de l'attribut MULTI_EXIT_DISC reçu sur EBGP. Si un locuteur BGP est configuré à altérer la valeur de l'attribut MULTI_EXIT_DISC reçu sur EBGP, cette altération de valeur DOIT être faite avant de déterminer le degré de préférence de la route et avant d'effectuer le choix de la route (phases 1 et 2 du processus de décision). Voir au paragraphe 9.1.2.2 les restrictions nécessaires à cela.

5.1.5 LOCAL_PREF

LOCAL_PREF est un attribut bien connu qui DEVRA être inclus dans tous les messages UPDATE qu'un certain locuteur BGP envoie aux autres homologues internes. Un locuteur BGP DEVRA calculer le degré de préférence pour chaque route externe sur la base de la politique configurée localement, et inclure le degré de préférence quand il annonce une route à ses homologues internes. Le plus haut degré de préférence DOIT être préféré. Un locuteur BGP utilise le degré de préférence appris via LOCAL_PREF dans son processus de décision (voir au paragraphe 9.1.1).

Un locuteur BGP NE DOIT PAS inclure cet attribut dans les messages UPDATE qu'il envoie à ses homologues externes, sauf dans le cas de confédérations BGP [RFC3065]. Si il est contenu dans un message UPDATE qui est reçu d'un homologue externe, cet attribut DOIT alors être ignoré par le locuteur receveur, sauf dans le cas de confédérations BGP [RFC3065].

5.1.6 ATOMIC_AGGREGATE

ATOMIC_AGGREGATE est un attribut discrétionnaire bien connu.

Lorsque un locuteur BGP agrège plusieurs routes pour les besoins d'une annonce à un homologue particulier, le AS_PATH de la route agrégée inclut normalement un AS_SET formé à partir de l'ensemble des AS qui ont servi à former la route agrégée. Dans de nombreux cas, l'administrateur du réseau peut déterminer si l'agrégat peut être annoncé en toute sécurité sans le AS_SET, et sans former de routes en boucles.

Si un agrégat exclut au moins certains des numéros d'AS présents dans le AS_PATH des routes qui ont été agrégées par suite de l'abandon de l'AS_SET, la route agrégée, quand elle est annoncée à l'homologue, DEVRAIT inclure l'attribut ATOMIC_AGGREGATE.

Un locuteur BGP qui reçoit une route avec un attribut ATOMIC_AGGREGATE NE DEVRAIT PAS retirer l'attribut quand il propage la route aux autres locuteurs.

Un locuteur BGP qui reçoit une route avec l'attribut ATOMIC_AGGREGATE NE DOIT PAS faire de NLRI plus spécifiques de cette route (comme défini au paragraphe 9.1.4) quand il annonce cette route aux autres locuteurs BGP.

Un locuteur BGP qui reçoit une route avec l'attribut ATOMIC_AGGREGATE doit être conscient du fait que le chemin réel pour les destinations, comme spécifié dans les NLRI de la route, tout en ayant la propriété de ne pas faire de boucle, peut n'être pas le chemin spécifié dans l'attribut AS_PATH de la route.

5.1.7 AGGREGATOR

AGGREGATOR est un attribut facultatif transitif, qui PEUT être inclus dans les mises à jour qui sont formées par agrégation (voir au paragraphe 9.2.2.2). Un locuteur BGP qui effectue une agrégation de routes PEUT ajouter l'attribut AGGREGATOR, qui DEVRA contenir son propre numéro d'AS et son adresse IP. L'adresse IP DEVRAIT être la même que l'identifiant BGP du locuteur.

6. Traitement d'erreur dans BGP

Cette section décrit les actions à entreprendre quand des erreurs sont détectées lors du traitement de messages BGP.

Quand une des conditions décrites ici est détectée, un message NOTIFICATION, avec les champs Code d'erreur, Sous code d'erreur, et Données indiqués, est envoyé, et la connexion BGP est close (sauf si il est explicitement déclaré qu'aucun message NOTIFICATION ne doit être envoyé et que la connexion BGP ne doit pas être close). Si aucun sous code d'erreur n'est spécifié, un zéro DOIT alors être utilisé.

La phrase "la connexion BGP est close" signifie que la connexion TCP a été close, que la Adj-RIB-In associée a été supprimée, et toutes les ressources pour cette connexion BGP ont été désallouées. Les entrées dans la Loc-RIB associée à l'homologue distant sont marquées comme invalides. Le système local recalcule ses meilleures routes pour les destinations des routes marquées comme invalides. Avant que les routes invalides soient supprimées du système, il annonce à ses homologues soit les retraits pour les routes marquées comme invalides, soit les nouvelles meilleures routes, avant que les routes invalides soient supprimées du système.

Sauf spécification explicite contraire, le champ Données du message NOTIFICATION, qui est envoyé pour indiquer une erreur, est vide.

6.1 Traitement d'erreur d'en-tête de message

Toutes les erreurs détectées lors du traitement de l'en-tête de message DOIVENT être indiquées par l'envoi du message NOTIFICATION avec le code d'erreur "Erreur d'en-tête de message". Le sous code d'erreur précise la nature spécifique de l'erreur.

La valeur attendue du champ Marqueur de l'en-tête de message est toute de uns. Si le champ Marqueur de l'en-tête de message n'est pas comme attendu, une erreur de synchronisation s'est produite et le sous code d'erreur DOIT être réglé à "Connexion non synchronisée".

Si au moins une des conditions suivantes est vraie :
- le champ Longueur de l'en-tête de message fait moins de 19 ou plus de 4096, ou
- le champ Longueur d'un message OPEN est moins que la longueur minimum du message OPEN, ou
- le champ Longueur d'un message UPDATE est moins que la longueur minimum du message UPDATE, ou
- le champ Longueur d'un message KEEPALIVE n'est pas égal à 19,

- le champ Longueur d'un message NOTIFICATION est moins que la longueur minimum de message NOTIFICATION, le sous code d'erreur DOIT alors être réglé à "Mauvaise longueur de message". Le champ Données DOIT contenir le champ Longueur erroné.

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

6.2 Traitement d'erreur du message OPEN

Toutes les erreurs détectées lors du traitement du message OPEN DOIVENT être indiquées par l'envoi du message NOTIFICATION avec le code d'erreur "Erreur de message OPEN". Le sous code d'erreur précise la nature spécifique de l'erreur.

Si le numéro de version du champ Version du message OPEN reçu n'est pas pris en charge, le sous code d'erreur DOIT être réglé à "Numéro de version non pris en charge". Le champ Données est un entier non signé de deux octets, qui indique le plus grand numéro de version pris en charge en local inférieur à la version de l'offre de l'homologue BGP distant (comme indiqué dans le message OPEN reçu) ou si le plus petit numéro de version pris en charge localement est supérieur à la version de l'offre de l'homologue BGP distant, alors le plus petit numéro de version pris en charge en local.

Si le champ Système autonome du message OPEN est inacceptable, le sous code d'erreur DOIT être réglé à "Mauvais AS homologue". La détermination des numéros de système autonome acceptables sort du domaine d'application de ce protocole.

Si le champ Temps de garde du message OPEN 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 de 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 BGP du message OPEN est syntaxiquement incorrect, le sous code d'erreur DOIT alors être réglé à "Mauvais identifiant BGP". La correction syntaxique signifie que le champ Identifiant BGP représente une adresse IP d'hôte en envoi individuel valide.

Si un des paramètres facultatifs dans le message OPEN n'est pas reconnu, le sous code d'erreur DOIT alors être réglé à "Paramètres facultatifs non pris en charge".

Si un des paramètres facultatifs dans le message OPEN est reconnu, mais est mal formé, le sous code d'erreur DOIT alors être réglé à 0 (Non spécifique).

6.3 Traitement d'erreur du message UPDATE

Toutes les erreurs détectées lors du traitement du message UPDATE DOIVENT être indiquées par l'envoi du message NOTIFICATION avec le code d'erreur "Erreur de message UPDATE". Le sous code d'erreur précise la nature spécifique de l'erreur.

La vérification d'erreur d'un message UPDATE commence par l'examen des attributs de chemin. Si la longueur des routes retirées ou la longueur totale d'attribut est trop grande (c'est-à-dire, si Longueur des routes retirées + Longueur totale d'attribut + 23 excède la longueur du message) le sous code d'erreur DOIT alors être réglé à "Liste d'attributs mal formée".

Si un des attributs reconnus 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 de fanions d'attribut". Le champ Données DOIT contenir l'attribut erroné (type, longueur, et valeur).

Si un attribut reconnu a une longueur d'attribut qui entre 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 DOIT contenir l'attribut erroné (type, longueur, et valeur).

Si un des attributs obligatoires bien connus n'est pas présent, le sous code d'erreur DOIT alors être réglé à "Attribut bien connu manquant". Le champ Données DOIT contenir le code de type d'attribut de l'attribut bien connu manquant.

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

Si l'attribut ORIGIN a une valeur indéfinie, le sous code d'erreur DOIT alors être réglé à "Attribut d'origine invalide". Le champ Données DOIT contenir l'attribut non reconnu (type, longueur, et valeur).

Si le champ Attribut NEXT_HOP est syntaxiquement incorrect, le sous code d'erreur DOIT alors être réglé à "Attribut NEXT_HOP invalide". Le champ Données DOIT contenir l'attribut incorrect (type, longueur, et valeur). Correction syntaxique signifie que l'attribut NEXT_HOP représente une adresse IP d'hôte valide.

L'adresse IP dans le NEXT_HOP DOIT satisfaire les critères suivants pour être considérée comme sémantiquement correcte :

a) elle NE DOIT PAS être l'adresse IP du locuteur receveur ;
b) dans le cas d'un EBGP, où l'envoyeur et le receveur sont à un bond IP l'un de l'autre, soit l'adresse IP dans le NEXT_HOP DOIT être l'adresse IP de l'envoyeur qui est utilisée pour établir la connexion BGP, soit l'interface associée à l'adresse IP du NEXT_HOP DOIT partager un sous réseau commun avec le locuteur BGP receveur.

Si l'attribut NEXT_HOP est sémantiquement incorrect, l'erreur DEVRAIT être enregistrée, et la route DEVRAIT être ignorée. Dans ce cas, un message NOTIFICATION NE DEVRAIT PAS être envoyé, et la connexion NE DEVRAIT PAS être close.

La correction syntaxique de l'attribut AS_PATH est vérifiée. Si le chemin est syntaxiquement incorrect, le sous code d'erreur DOIT alors être réglé à "AS_PATH mal formé".

Si le message UPDATE est reçu d'un homologue externe, le système local PEUT vérifier si l'AS le plus à gauche (par rapport à la position des octets dans le message de protocole) dans l'attribut AS_PATH est égal au numéro de système autonome de l'homologue qui a envoyé le message. Si la vérification détermine que ce n'est pas le cas, le sous code d'erreur DOIT être réglé à "AS_PATH mal formé".

Si un attribut facultatif est reconnu, la valeur de cet attribut DOIT alors être vérifiée. Si une erreur est détectée, le sous code d'erreur DOIT être réglé à "Erreur d'attribut facultatif". Le champ Données DOIT contenir l'attribut (type, longueur, et valeur).

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

La correction syntaxique du champ NLRI dans le message UPDATE est vérifiée.Si le champ est syntaxiquement incorrect, le sous code d'erreur DOIT alors être réglé à "Champ réseau invalide".

Si un préfixe dans le champ NLRI est sémantiquement incorrect (par exemple, une adresse IP de diffusion groupée inattendue) une erreur DEVRAIT être enregistrée en local, et le préfixe DEVRAIT être ignoré.

Un message UPDATE qui contient des attributs de chemin corrects, mais pas de NLRI, DEVRA être traité comme un message UPDATE valide.

6.4 Traitement d'erreur du message NOTIFICATION

Si un homologue envoie un message NOTIFICATION, et si le receveur du message détecte une erreur dans ce message, le receveur ne peut pas utiliser un message NOTIFICATION pour rapporter cette erreur à l'homologue. Toute erreur de cette sorte (par exemple, un code d'erreur ou sous code d'erreur non reconnu) DEVRAIT être noté, enregistré en local, et porté à l'attention de l'administration de l'homologue. Les moyens de faire cela sont cependant en dehors du domaine d'application du présent document.

6.5 Traitement d'erreur d'expiration du temporisateur de garde

Si un système ne reçoit pas de message KEEPALIVE, UPDATE, et/ou NOTIFICATION successifs au sein de la période spécifiée dans le champ Temps de garde du message OPEN, le message NOTIFICATION avec le code d'erreur "Temporisateur de garde expiré" est alors envoyé et la connexion BGP est close.

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

Toute erreur détectée par l'automate à états finis de BGP (par exemple, la réception d'un événement inattendu) est indiquée par l'envoi du message NOTIFICATION avec le code d'erreur "Erreur d'automate à états finis".

6.7 Cessation

En l'absence de toute erreur fatale (qui sont indiquées dans cette section) un homologue BGP PEUT choisir, à tout moment, de clore sa connexion BGP par l'envoi du message NOTIFICATION avec le code d'erreur "Cessation". Cependant, le message de NOTIFICATION Cessation NE DOIT PAS être utilisé quand existe une erreur fatale indiquée dans cette section.

Un locuteur BGP PEUT prendre en charge la capacité d'imposer une limite supérieure, configurée en local, au nombre de préfixes d'adresse que le locuteur veut accepter d'un voisin. Quand la limite supérieure est atteinte, le locuteur, sous le contrôle de la configuration locale, (a) élimine les nouveaux préfixes d'adresse provenant du voisin (tout en conservant la connexion BGP avec le voisin), ou (b) termine la connexion BGP avec le voisin. Si le locuteur BGP décide de terminer sa connexion BGP avec un voisin parce que le nombre de préfixes d'adresse reçus du voisin excède la limite supérieure configurée en local, le locuteur DOIT envoyer au voisin un message NOTIFICATION avec le code d'erreur "Cessation". Le locuteur PEUT aussi enregistrer cela en local.

6.8 Détection de collision de connexions BGP

Si deux locuteurs BGP essayent d'établir une connexion BGP simultanément l'un avec l'autre, deux connexions parallèles vont être formées. Si l'adresse IP de source utilisée par une de ces connexions est la même que l'adresse IP de destination utilisée par l'autre, et si l'adresse IP de destination utilisée par la première connexion est la même que l'adresse IP de source utilisée par l'autre, une collision de connexion se produit. Dans le cas d'une collision de connexion, une des connexions DOIT être close.

Sur la base de la valeur de l'identifiant BGP, une convention est établie pour détecter quelle connexion BGP doit être préservée quand une collision se produit. La convention est de comparer les identifiants BGP des homologues impliqués dans la collision et de ne garder que la connexion initiée par le locuteur BGP avec l'identifiant BGP de plus forte valeur.

À réception d'un message OPEN, le système local DOIT examiner toutes ses connexions qui sont dans l'état OuvertConfirmé. Un locuteur BGP PEUT aussi examiner les connexions dans un état OuvertEnvoyé si il connaît l'identifiant BGP de l'homologue par des moyens hors protocole. Si, parmi ces connexions, il y a une connexion à un locuteur BGP distant dont l'identifiant BGP est égal à celui du message OPEN, et si cette connexion entre en collision avec la connexion sur laquelle le message OPEN est reçu, le système local effectue alors la procédure de résolution de collision suivante :

1) L'identifiant BGP du système local est comparé à l'identifiant BGP du système distant (comme spécifié dans le message OPEN). La comparaison des identifiants BGP est faite en les convertissant dans l'ordre d'octets de l'hôte et en les traitant comme des entiers non signés de quatre octets.
2) Si la valeur de l'identifiant BGP local est inférieure à celle de l'identifiant distant, le système local clôt la connexion BGP déjà existante (celle qui est déjà dans l'état OuvertConfirmé) et accepte la connexion BGP initiée par le système distant.
3) Autrement, le système local clôt la connexion BGP nouvellement créée (celle associée au message OPEN nouvellement reçu) et continue d'utiliser celle existante (qui est déjà dans l'état OuvertConfirmé).

Sauf si c'est permis via la configuration, une collision de connexion avec une connexion BGP existante dans l'état Établi cause la clôture de la connexion nouvellement créée.

Noter qu'une collision de connexion ne peut être détectée avec des connexions dans les états Repos, Connecté, ou Actif.

La clôture de la connexion BGP (qui résulte de la procédure de résolution de collision) est accomplie par l'envoi du message NOTIFICATION avec le code d'erreur "Cessation".

7. Négociation de la version BGP

Les locuteurs BGP PEUVENT négocier la version du protocole en faisant plusieurs tentatives à l'ouverture d'une connexion BGP, en commençant par le plus haut numéro de version que chaque locuteur BGP supporte. Si une tentative d'ouverture échoue avec un code d'erreur "Erreur de message OPEN", et un sous code d'erreur "Numéro de version non accepté", le locuteur BGP a 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 le numéro de version qu'il prend en charge. Si les deux homologues prennent en charge une ou plusieurs versions communes, cela leur permettra de déterminer

rapidement la version commune la plus élevée. Afin de prendre en charge la négociation de version de BGP, les futures versions de BGP DEVRONT conserver le format des messages OPEN et NOTIFICATION.

8. Automate à états finis pour BGP

Les structures de données et l'automate à états finis (FSM, *Finite State Machine*) décrits dans ce document sont conceptuels et n'ont pas à être mis en œuvre précisément comme décrit ici, pour autant que la mise en œuvre prenne en charge la fonctionnalité décrite et qu'elle présente le même comportement externe visible.

Cette section spécifie le fonctionnement de BGP en termes d'automate à états finis. La section comporte deux parties :
1) Description des événements de l'automate à états finis (paragraphe 8.1)
2) Description du FSM (paragraphe 8.2)

Les attributs de session exigés (obligatoires) pour chaque connexion sont :
1) State *(état)*
2) ConnectRetryCounter *(compteur d'essais de connexion)*
3) ConnectRetryTimer *(temporisateur d'essai de connexion)*
4) ConnectRetryTime *(intervalle d'essai de connexion)*
5) HoldTimer *(temporisateur de garde)*
6) HoldTime *(temps de garde)*
7) KeepaliveTimer *(temporisateur de garde en vie)*
8) KeepaliveTime *(durée de garde en vie)*

L'attribut d'état de session indique l'état actuel du FSM BGP. Le ConnectRetryCounter indique le nombre de fois qu'un homologue BGP a essayé d'établir une session avec un homologue.

Les attributs obligatoires relatifs aux temporisateurs sont décrits à la Section 10. Chaque temporisateur a un "temporisateur" et une "heure" (la valeur initiale).

La liste des attributs de session facultatifs figure ci-dessous. Ces attributs facultatifs peuvent être pris en charge par connexion ou par système local :
1) AcceptConnectionsUnconfiguredPeers *(accepter des connexions d'homologues non configurés)*
2) AllowAutomaticStart *(permettre le démarrage automatique)*
3) AllowAutomaticStop *(permettre l'arrêt automatique)*
4) CollisionDetectEstablishedState *(détection de collision dans l'état établi)*
5) DampPeerOscillations *(amortissement des oscillations de l'homologue)*
6) DelayOpen *(retard d'ouverture)*
7) DelayOpenTime *(heure de retard d'ouverture)*
8) DelayOpenTimer *(temporisateur de retard d'ouverture)*
9) IdleHoldTime *(temps de garde en repos)*
10) IdleHoldTimer *(temporisateur de garde en repos)*
11) PassiveTcpEstablishment *(établissement TCP passif)*
12) SendNOTIFICATIONwithoutOPEN *(envoi de notification sans message OPEN)*
13) TrackTcpState *(retraçage de l'état TCP)*

Les attributs de session facultatifs prennent en charge différentes caractéristiques des fonction de BGP qui ont des implications pour les transitions d'état du FSM BGP. Deux groupes d'attributs qui se rapportent aux temporisateurs sont :
groupe 1 : DelayOpen, DelayOpenTime, DelayOpenTimer
groupe 2 : DampPeerOscillations, IdleHoldTime, IdleHoldTimer

Le premier paramètre (DelayOpen, DampPeerOscillations) est un attribut facultatif qui indique que la fonction de temporisateur est active. La valeur "Time" spécifie la valeur initiale pour le "Timer" (DelayOpenTime, IdleHoldTime). "Timer" spécifie le temporisateur réel.

Prière de se référer au paragraphe 8.1.1 pour l'explication de l'interaction entre ces attributs facultatifs et les événements signalés à l'automate à états. Le paragraphe 8.2.1.3 fournit aussi un bref survol des différents types d'attributs facultatifs (fanions ou temporisateurs).

8.1 Événements pour le FSM BGP

8.1.1 Événements facultatifs liés aux attributs de session facultatifs

Les entrées au FSM BGP sont des événements. Les événements peuvent être obligatoires ou facultatifs. Certains événements facultatifs sont liés à des attributs de session facultatifs. Les attributs de session facultatifs activent plusieurs groupes de fonctions de FSM.

Les liens entre une fonction de FSM, les événements, et les attributs de session facultatifs sont décrits ci-dessous.

Groupe 1 : événements administratifs automatiques (démarrage/arrêt)
Attributs de session facultatifs : AllowAutomaticStart, AllowAutomaticStop, DampPeerOscillations, IdleHoldTime, IdleHoldTimer

Option 1 : AllowAutomaticStart
Description : une connexion d'homologue BGP peut être démarrée et arrêtée par une commande administrative. Cette commande administrative peut être manuelle, sur la base de l'intervention de l'opérateur, ou sous le contrôle d'une logique spécifique d'une mise en œuvre de BGP. Le terme "automatique" se réfère à un démarrage produit au FSM de la connexion d'homologue BGP quand une telle logique détermine que la connexion d'homologue BGP devrait être redémarrée.
L'attribut AllowAutomaticStart spécifie que cette connexion BGP prend en charge le démarrage automatique de la connexion BGP.
Si la mise en œuvre BGP prend en charge AllowAutomaticStart, l'homologue peut être redémarré de façon répétée. Trois autres options contrôlent le débit auquel se produit le redémarrage automatique : DampPeerOscillations, IdleHoldTime, et the IdleHoldTimer.
L'option DampPeerOscillations spécifie que la mise en œuvre engage une logique supplémentaire pour atténuer les oscillations des homologues BGP en présence de séquences de démarrages et arrêts automatiques. IdleHoldTime spécifie la durée pendant laquelle l'homologue BGP est conservé dans l'état Repos avant de permettre le prochain redémarrage automatique. IdleHoldTimer est le temporisateur qui maintient l'homologue dans l'état Repos.
Un exemple de logique de DampPeerOscillations est une augmentation de la valeur de IdleHoldTime si la connexité d'un homologue BGP oscille (connecté/déconnecté) de façon répétée dans une certaine période de temps. Pour engager cette logique, un homologue pourrait connecter et déconnecter 10 fois en 5 minutes. La valeur de IdleHoldTime pourrait être réglée de 0 à 120 secondes.
Valeurs : VRAI ou FAUX

Option 2 : AllowAutomaticStop
Description : cet attribut de session facultatif d'homologue BGP indique que la connexion BGP permet un arrêt "automatique" de la connexion BGP. Arrêt "automatique" est défini comme un arrêt sous le contrôle d'une logique spécifique de la mise en œuvre. La logique spécifique de la mise en œuvre sort du domaine d'application de la présente spécification.
Valeurs : VRAI ou FAUX

Option 3 : DampPeerOscillations
Description : l'attribut de session facultatif DampPeerOscillations indique que la connexion BGP utilise une logique qui amortit les oscillations de l'homologue BGP dans l'état Repos.
Valeurs : VRAI ou FAUX

Option 4 : IdleHoldTime
Description : IdleHoldTime est la valeur qui est réglée dans le IdleHoldTimer.
Valeurs : durée en secondes

Option 5 : IdleHoldTimer
Description : IdleHoldTimer aide à contrôler les oscillations de l'homologue BGP. Le IdleHoldTimer est utilisé pour garder l'homologue BGP dans l'état Repos pour une durée particulière. L'événement IdleHoldTimer_Expires est décrit au paragraphe 8.1.3.
Valeurs : durée en secondes

Groupe 2 : homologues non configurés
Attributs de session facultatifs : AcceptConnectionsUnconfiguredPeers

Option 1 : AcceptConnectionsUnconfiguredPeers

Description : le FSM BGP permet facultativement l'acceptation de connexions avec un homologue BGP provenant de voisins qui ne sont pas pré configurés. L'attribut de session facultatif "AcceptConnectionsUnconfiguredPeers" permet au FSM de prendre en charge les transitions d'état qui permettent à la mise en œuvre d'accepter ou rejeter ces homologues non configurés.
AcceptConnectionsUnconfiguredPeers a des implications sur la sécurité. Prière de se référer au document sur les vulnérabilités de BGP [RFC4272] pour les détails.
Valeurs : VRAI ou FAUX

Groupe 3 : traitement de TCP
Attributs de session facultatifs : PassiveTcpEstablishment, TrackTcpState

Option 1 : PassiveTcpEstablishment
Description : cette option indique que le FSM BGP va attendre passivement que l'homologue BGP distant établisse la connexion TCP pour BGP.
Valeurs : VRAI ou FAUX

Option 2 : TrackTcpState
Description : le FSM BGP retrace normalement le résultat final d'une tentative de connexion TCP plutôt que les messages TCP individuels. Facultativement, le FSM BGP peut prendre en charge des interactions supplémentaires avec la négociation de connexion TCP. Les interactions avec les événements TCP peuvent augmenter la quantité d'enregistrements qu'exige la connexion d'homologue BGP et le nombre de changements du FSM.
Valeurs : VRAI ou FAUX

Groupe 4 : traitement de message BGP
Attributs de session facultatifs : DelayOpen, DelayOpenTime, DelayOpenTimer, SendNOTIFICATIONwithoutOPEN, CollisionDetectEstablishedState

Option 1 : DelayOpen
Description : l'attribut de session facultatif DelayOpen permet aux mises en œuvre d'être configurées à retarder l'envoi d'un message OPEN d'un délai spécifique (DelayOpenTime). Le retard donne à l'homologue BGP distant le temps d'envoyer le premier message OPEN.
Valeurs : VRAI ou FAUX

Option 2 : DelayOpenTime
Description : c'est la valeur initiale réglée dans le DelayOpenTimer.
Valeur : durée en secondes

Option 3 : DelayOpenTimer
Description : l'attribut de session facultatif DelayOpenTimer est utilisé pour retarder l'envoi d'un message OPEN sur une connexion. L'événement DelayOpenTimer_Expires (événement 12) est décrit au paragraphe 8.1.3.
Valeur : durée en secondes

Option 4 : SendNOTIFICATIONwithoutOPEN
Description : SendNOTIFICATIONwithoutOPEN permet à un homologue d'envoyer une NOTIFICATION sans envoyer d'abord un message OPEN. Sans cet attribut de session facultatif, la connexion BGP suppose qu'un message OPEN doit être envoyé par l'homologue avant que l'homologue envoie un message NOTIFICATION.
Valeurs : VRAI ou FAUX

Option 5 : CollisionDetectEstablishedState
Description : Normalement, une collision détectée (voir le paragraphe 6.8) va être ignorée dans l'état Établi. Cet attribut de session facultatif indique que cette connexion BGP traite les collisions dans l'état Établi.
Valeurs : VRAI ou FAUX

Note : les attributs de session facultatifs précisent la description du FSM BGP pour les caractéristiques existantes de mises en œuvre de BGP. Les attributs de session facultatifs peuvent être prédéfinis pour une mise en œuvre et non lisibles via les interfaces de gestion pour des mises en œuvre existantes correctes. Lorsque de plus récentes MIB BGP (version 2 et au delà) sont prises en charge, ces champs vont être accessibles via une interface de gestion.

8.1.2 Événements administratifs

Un événement administratif est un événement dans lequel une interface d'opérateur et un moteur de politique BGP signalent à l'automate à états finis de BGP de commencer ou arrêter l'automate BGP. Les indications de base de démarrage

et d'arrêt sont augmentées d'attributs de connexion facultatifs qui signalent un certain type de mécanisme de démarrage ou d'arrêt au FSM BGP. Un exemple de cette combinaison est l'événement 5, AutomaticStart_with_PassiveTcpEstablishment. Avec cet événement, la mise en œuvre BGP signale au FSM BGP que la mise en œuvre utilise un démarrage automatique avec l'option d'utiliser un établissement TCP passif. L'établissement TCP passif signale que ce FSM BGP va attendre que le côté distant démarre l'établissement de TCP.

Noter que seuls l'événement 1 (ManualStart) et l'événement 2 (ManualStop) sont des événements administratifs obligatoires. Tous les autres événements administratifs sont facultatifs (événements 3 à 8). Chaque événement ci-dessous a un nom, une définition, un statut (obligatoire ou facultatif), et les attributs facultatifs de session qui DEVRAIENT être établis à chaque étape. Lors de la génération des événements 1 à 8 pour le FSM BGP, les conditions spécifiées dans la section "Statut d'attribut facultatif" sont vérifiées. Si une de ces conditions n'est pas satisfaite, le système local devrait alors enregistrer une erreur de FSM.

Les réglages des attributs de session facultatifs peuvent être implicites dans certaines mises en œuvre, et donc peuvent n'être pas réglés explicitement par une action externe d'opérateur. Le paragraphe 8.2.1.5 décrit ces réglages implicites des attributs de session facultatifs. Les états administratifs décrits ci dessous peuvent aussi être implicites dans certaines mises en œuvre et n'être pas directement configurables par un opérateur externe.

Événement 1 : ManualStart
Définition : l'administrateur de système local démarre manuellement la connexion homologue.
Statut : obligatoire
Facultatif
Attribut
Statut : l'attribut PassiveTcpEstablishment DEVRAIT être réglé à FAUX.

Événement 2 : ManualStop
Définition : l'administrateur de système local arrête manuellement la connexion homologue.
Statut : obligatoire
Facultatif
Attribut
Statut : pas d'interaction avec un attribut facultatif.

Événement 3: AutomaticStart
Définition : le système local démarre automatiquement la connexion BGP.
Statut : facultatif, selon le système local
Facultatif
Attribut
Statut : 1) l'attribut AllowAutomaticStart DEVRAIT être réglé à VRAI si cet événement se produit.
2) si l'attribut de session facultatif PassiveTcpEstablishment est pris en charge, il DEVRAIT être réglé à FAUX.
3) si DampPeerOscillations est pris en charge, il DEVRAIT être réglé à FAUX quand cet événement arrive.

Événement 4 : ManualStart_with_PassiveTcpEstablishment
Définition : l'administrateur de système local démarre manuellement la connexion homologue, mais a PassiveTcpEstablishment activé. L'attribut facultatif PassiveTcpEstablishment indique que l'homologue va écouter avant d'établir la connexion.
Statut : facultatif, selon le système local
Facultatif
Attribut
Statut : 1) l'attribut PassiveTcpEstablishment DEVRAIT être réglé à VRAI si cet événement se produit.
2) l'attribut DampPeerOscillations DEVRAIT être réglé à FAUX quad cet événement se produit.

Événement 5 : AutomaticStart_with_PassiveTcpEstablishment
Définition : le système local démarre automatiquement la connexion BGP avec PassiveTcpEstablishment activé. L'attribut facultatif PassiveTcpEstablishment indique que l'homologue va écouter avant d'établir une connexion.
Statut : facultatif, selon le système local
Facultatif
Attribut
Statut : 1) l'attribut AllowAutomaticStart DEVRAIT être réglé à VRAI.
2) l'attribut PassiveTcpEstablishment DEVRAIT être réglé à VRAI.
3) si l'attribut DampPeerOscillations est pris en charge, DampPeerOscillations DEVRAIT être réglé à FAUX.

Événement 6 : AutomaticStart_with_DampPeerOscillations

Définition : le système local démarre automatiquement la connexion d'homologue BGP avec l'atténuation des oscillations d'homologue activée. La méthode exacte d'atténuation persistante des oscillations d'homologue est déterminée par la mise en œuvre et sort du domaine d'application du présent document.
Statut : facultatif, selon le système local.
Facultatif
Attribut
Statut : 1) l'attribut AllowAutomaticStart DEVRAIT être réglé à VRAI.
2) l'attribut DampPeerOscillations DEVRAIT être réglé à VRAI.
3) l'attribut PassiveTcpEstablishment DEVRAIT être réglé à FAUX.

Événement 7 : AutomaticStart_with_DampPeerOscillations_et_PassiveTcpEstablishment
Définition : le système local démarre automatiquement la connexion d'homologue BGP avec l'atténuation d'oscillation d'homologue activée et PassiveTcpEstablishment activé. La méthode exacte d'atténuation persistantes des oscillations d'homologue est déterminée par la mise en œuvre et sort du domaine d'application du présent document.
Statut : facultatif, selon le système local
Facultatif
Attributs
Statut : 1) l'attribut AllowAutomaticStart DEVRAIT être réglé à VRAI.
2) l'attribut DampPeerOscillations DEVRAIT être réglé à VRAI.
3) l'attribut PassiveTcpEstablishment DEVRAIT être réglé à VRAI.

Événement 8 : AutomaticStop
Définition : le système local arrête automatiquement la connexion BGP. Un exemple d'événement d'arrêt automatique est quand le nombre de préfixes pour un certain homologue excède un seuil et que le système local déconnecte automatiquement l'homologue.
Statut : facultatif, selon le système local
Facultatif
Attribut
Statut : 1) l'attribut AllowAutomaticStop DEVRAIT être VRAI.

8.1.3 Événements de temporisateur

Événement 9 : ConnectRetryTimer_Expires
Définition : événement généré quand le ConnectRetryTimer arrive à expiration.
Statut : obligatoire

Événement 10 : HoldTimer_Expires
Définition : événement généré quand le HoldTimer arrive à expiration.
Statut : obligatoire

Événement 11 : KeepaliveTimer_Expires
Définition : événement généré quand le KeepaliveTimer arrive à expiration.
Statut : obligatoire

Événement 12 : DelayOpenTimer_Expires
Définition : événement généré quand le DelayOpenTimer arrive à expiration.
Statut : facultatif
Facultatif
Attribut
Statut : si cet événement se produit :
1) l'attribut DelayOpen DEVRAIT être réglé à VRAI,
2) l'attribut DelayOpenTime DEVRAIT être pris en charge,
3) DelayOpenTimer DEVRAIT être pris en charge.

Événement 13 : IdleHoldTimer_Expires
Définition : événement généré quand le IdleHoldTimer arrive à expiration, ce qui indique que la connexion BGP a fini d'attendre la période de retard pour empêcher les oscillations de l'homologue BGP. Le IdleHoldTimer n'est utilisé que quand la fonction d'atténuation d'oscillations persistantes de l'homologue est activée par le réglage de l'attribut DampPeerOscillations à VRAI. Les mises en œuvre qui n'appliquent pas la fonction d'atténuation d'oscillations persistantes de l'homologue peuvent n’avoir pas le IdleHoldTimer.
Statut : facultatif

Facultatif
Attribut
Statut : si cet événement se produit :
1) l'attribut DampPeerOscillations DEVRAIT être réglé à VRAI.
2) IdleHoldTimer DEVRAIT être juste arrivé à expiration.

8.1.4 Événements fondés sur la connexion TCP

Événement 14 : TcpConnection_Valid
Définition : événement qui indique que le système local a reçu une demande de connexion TCP avec une adresse IP de source, un accès TCP, une adresse IP de destination, et un accès TCP valides. La définition d'une adresse IP de source et de destination invalides est déterminée par la mise en œuvre. L'accès de destination BGP DEVRAIT être l'accès 179, comme défini par l'IANA. La demande de connexion TCP est notée par le système local à réception d'un TCP SYN.
Statut : facultatif
Facultatif
Attribut
Statut : 1) L'attribut TrackTcpState DEVRAIT être réglé à VRAI si cet événement se produit.

Événement 15 : Tcp_CR_Invalid
Définition : événement qui indique la réception par le système local d'une demande de connexion TCP avec une adresse de source ou un numéro d'accès invalide, ou une adresse de destination ou numéro d'accès invalide. Le numéro d'accès de destination BGP DEVRAIT être 179, comme défini par l'IANA. Une demande de connexion TCP se produit quand le système local reçoit un TCP SYN.
Statut : facultatif
Facultatif
Attribut
Statut : l'attribut TrackTcpState devrait être réglé à VRAI si cet événement se produit.

Événement 16 : Tcp_CR_Acked
Définition : événement indiquant la demande du système local d'établir une connexion TCP avec l'homologue distant. La connexion TCP du système local a envoyé un TCP SYN, reçu un message TCP SYN/ACK, et envoyé un TCP ACK.
Statut : obligatoire

Événement 17 : TcpConnectionConfirmed
Définition : événement indiquant que le système local a reçu une confirmation que la connexion TCP a été établie par le site distant. Le moteur TCP de l'homologue distant a envoyé un TCP SYN. Le moteur TCP de l'homologue local a envoyé un message SYN, un ACK et a maintenant reçu un ACK final.
Statut : obligatoire

Événement 18 : TcpConnectionFails
Définition : événement indiquant que le système local a reçu une notification d'échec de connexion TCP. La machine TCP de l'homologue BGP distant pourrait avoir envoyé un FIN. L'homologue local va répondre avec un FIN-ACK. Une autre possibilité est que l'homologue local ait indiqué une fin de temporisation de la connexion TCP et y ait mit fin.
Statut : obligatoire

8.1.5 Événements fondés sur la connexion BGP

Événement 19 : BGPOpen
Définition : événement généré quand un message OPEN valide a été reçu.
Statut : obligatoire
Facultatif
Attribut
Statut : 1) L'attribut facultatif DelayOpen DEVRAIT être réglé à FAUX.
2) Le DelayOpenTimer NE DEVRAIT PAS être en cours.

Événement 20 : BGPOpen avec le temporisateur DelayOpenTimer en cours
Définition : événement généré quand un message OPEN valide a été reçu d'un homologue qui a réussi à établir une connexion de transport et qu'il retarde actuellement l'envoi d'un message OPEN BGP.
Statut : facultatif

Facultatif
Attribut
Statut : 1) l'attribut DelayOpen DEVRAIT être réglé à VRAI.
2) DelayOpenTimer DEVRAIT être en cours.

Événement 21 : BGPHeaderErr
Définition : événement généré quand un en-tête de message BGP reçu n'est pas valide.
Statut : obligatoire

Événement 22 : BGPOpenMsgErr
Définition : événement généré quand un message OPEN a été reçu avec des erreurs.
Statut : obligatoire

Événement 23 : OpenCollisionDump
Définition : événement généré administrativement quand une collision de connexions a été détectée pendant le traitement d'un message OPEN entrant et que cette connexion a été choisie pour être déconnectée. Voir au paragraphe 6.8 plus d'informations sur la détection des collisions. L'événement 23 est une action administrative générée par la logique de mise en œuvre qui détermine si cette connexion doit être abandonnée selon les règles du paragraphe 6.8. Cet événement peut se produire si le FSM est mis en œuvre comme deux automates à états reliés.
Statut : facultatif
Facultatif
Attribut
Statut : si l'automate à états va traiter cet événement dans l'état Établi, l'attribut facultatif CollisionDetectEstablishedState DEVRAIT être réglé à VRAI.
Note : L'événement OpenCollisionDump peut se produire dans les état Repos, Connecté, Actif, OuvertEnvoyé, et OuvertConfirmé sans qu'aucun attribut facultatif soit établi.

Événement 24 : NotifMsgVerErr
Définition : événement généré quand un message NOTIFICATION avec "erreur de version" est reçu.
Statut : obligatoire

Événement 25 : NotifMsg
Définition : événement généré quand un message NOTIFICATION est reçu et que le code d'erreur est tout sauf "erreur de version".
Statut : obligatoire

Événement 26 : KeepAliveMsg
Définition : événement généré quand un message KEEPALIVE est reçu.
Statut : obligatoire

Événement 27 : UpdateMsg
Définition : événement généré quand un message UPDATE valide est reçu.
Statut : obligatoire

Événement 28 : UpdateMsgErr
Définition : événement généré quand un message UPDATE invalide est reçu.
Statut : obligatoire

8.2 Description du FSM

8.2.1 Définition du FSM

BGP DOIT tenir un FSM séparé pour chaque homologue configuré. Chaque homologue BGP apparié dans une connexion potentielle va tenter de se connecter à l'autre, sauf si il est configuré à rester dans l'état Repos, ou si il est configuré à rester passif. Pour les besoins de cette discussion, le côté actif, ou qui se connecte, de la connexion TCP (le côté d'une connexion TCP qui envoie le premier paquet TCP SYN) est appelé sortant. Le côté passif ou qui écoute (l'envoyeur du premier SYN/ACK) est appelé une connexion entrante. (Voir au paragraphe 8.2.1.1 des informations sur les termes actif et passif utilisés ci-dessous.)

Une mise en œuvre BGP DOIT se connecter et écouter sur l'accès TCP 179 pour les connexions entrantes en plus d'essayer de se connecter aux homologues. Pour chaque connexion entrante, un automate à états DOIT être instancié. Il existe une période dans laquelle l'identité de l'homologue à l'autre extrémité d'une connexion entrante est connue, mais l'identifiant

BGP n'est pas connu. Durant cette période, une connexion entrante et une connexion sortante peuvent toutes deux exister pour la même paire d'homologues configurée. C'est ce qu'on appelle une collision de connexions (voir le paragraphe 6.8).

Une mise en œuvre BGP va avoir, au plus, un FSM pour chaque appariement configuré, plus un FSM pour chaque connexion TCP entrante pour laquelle l'homologue n'a pas encore été identifié. Chaque FSM correspond à exactement une connexion TCP.

Il peut y avoir plus d'une connexion entre une paire d'homologues si les connexions sont configurées à utiliser une paire d'adresses IP différentes. C'est ce qu'on appelle des "appariements configurés" multiples au même homologue.

8.2.1.1 Termes "actif" et "passif"

Les termes "actif" et "passif" ont été depuis près de dix ans dans le vocabulaire des opérateurs de l'Internet et leur utilité a été prouvée. Les mots actif et passif ont une signification légèrement différente lorsque ils sont appliqués à une connexion TCP ou à un homologue. Il y a seulement un côté actif et un côté passif pour toute connexion TCP, selon la définition ci-dessus et l'automate à états ci-dessous. Quand un locuteur BGP est configuré comme actif, il peut finir sur l'un ou l'autre côté, actif ou passif de la connexion qui sera établie. Une fois la connexion TCP achevée, savoir quel côté était actif ou passif n'a pas d'importance. La seule différence est sur le côté de la connexion TCP qui a le numéro d'accès 179.

8.2.1.2 FSM et détection de collision

Il y a un FSM par connexion BGP. Quand la collision de connexions se produit avant d'avoir pu déterminer à quel homologue une connexion est associée, il peut y avoir deux connexions pour un homologue. Après la résolution de la collision de connexions (voir au paragraphe 6.8) le FSM pour la connexion qui est close DEVRAIT être fermé.

8.2.1.3 FSM et attributs de session facultatifs

Les attributs de session facultatifs spécifient des attributs qui agissent comme des fanions (VRAI ou FAUX) ou comme des temporisateurs facultatifs. Pour les attributs facultatifs qui agissent comme des fanions, si l'attribut de session facultatif peut être réglé à VRAI sur le système, les actions du FSM BGP correspondant doivent être prises en charge. Par exemple, si les options suivantes peuvent être établies dans une mise en œuvre BGP : AutoStart et PassiveTcpEstablishment, les événements 3, 4 et 5 doivent alors être pris en charge. Si un attribut de session facultatif ne peut pas être réglé à VRAI, les événements qui prennent en charge cet ensemble d'options n'ont pas à être pris en charge.

Chacun des temporisateurs facultatifs (DelayOpenTimer et IdleHoldTimer) a un groupe d'attributs qui sont :
- le fanion indiquant la prise en charge,
- l'heure réglée dans le temporisateur,
- le temporisateur.

Les deux temporisateurs facultatifs présentent ce format :
DelayOpenTimer *(temporisateur de retard d'ouverture)* : DelayOpen, DelayOpenTime, DelayOpenTimer
IdleHoldTimer *(temporisateur de garde en repos)* : DampPeerOscillations, IdleHoldTime, IdleHoldTimer

Si le fanion qui indique la prise en charge d'un temporisateur facultatif (DelayOpen ou DampPeerOscillations) ne peut pas être réglé à VRAI, les temporisateurs et événements qui prennent en charge cette option n'ont pas à être pris en charge.

8.2.1.4 Numéros d'événements de FSM

Les numéros d'événement (1 à 28) utilisés dans cette description d'automate à états aident à spécifier le comportement de l'automate à états BGP. Les mises en œuvre PEUVENT utiliser ces numéros pour fournir des informations de gestion de réseau. La forme exacte d'un FSM ou les événements de FSM sont spécifiques de chaque mise en œuvre.

8.2.1.5 Actions du FSM qui dépendent de la mise en œuvre

À certains points, le FSM BGP spécifie que l'initialisation de BGP va se produire ou que des ressources BGP vont être supprimées. L'initialisation du FSM BGP et les ressources associées dépendent de la portion de politique de la mise en œuvre BGP. Les détails de ces actions sortent du domaine d'application du document de FSM.

8.2.2 Automate à états finis

État Repos *(Idle)* : Initialement, le FSM de l'homologue BGP est dans l'état Repos. (À partir d'ici, le terme "FSM de l'homologue BGP" sera abrégé en FSM BGP.)

Dans cet état, le FSM BGP refuse toutes les connexions BGP entrantes pour cet homologue. Aucune ressource n'est allouée à l'homologue. En réponse à un événement ManualStart (Événement 1) ou à un événement AutomaticStart (Événement 3) le système local :
- initialise toutes les ressources BGP pour la connexion homologue,
- règle le ConnectRetryCounter *(compteur d'essais de connexion)* à zéro,
- démarre le ConnectRetryTimer *(temporisateur d'essai de connexion)* avec la valeur initiale,
- initie une connexion TCP avec l'autre homologue BGP,
- écoute si une connexion peut être initiée par l'homologue BGP distant, et
- change son état en Connecter.

Les événements ManualStop (Événement 2) et AutomaticStop (Événement 8) sont ignorés dans l'état Repos.

En réponse à un événement ManualStart_with_PassiveTcpEstablishment (Événement 4) ou AutomaticStart_with_PassiveTcpEstablishment (Événement 5), le système local :
- initialise toutes les ressources BGP,
- règle le ConnectRetryCounter à zéro,
- démarre le ConnectRetryTimer avec la valeur initiale,
- écoute si une connexion peut être initiée par l'homologue BGP distant, et
- change son état en Actif.

La valeur exacte du temporisateur ConnectRetryTimer est une affaire locale, mais elle DEVRAIT être suffisamment grande pour permetrre l'initialisation de TCP.

Si l'attribut DampPeerOscillations est réglé à VRAI, les trois événements supplémentaires suivants peuvent se produire au sein de l'état Repos :
- AutomaticStart_avec_DampPeerOscillations (Événement 6),
- AutomaticStart_avec_DampPeerOscillations_et_PassiveTcpEstablishment (Événement 7),
- IdleHoldTimer_Expires (Événement 13).

À réception d'un de ces trois événements, le système local va utiliser ces événements pour empêcher les oscillations de l'homologue. La méthode pour empêcher l'oscillation persistante de l'homologue sort du domaine d'application du présent document.

Tout autre événement (Événements 9 à 12, 15 à 28) reçu dans l'état Repose ne cause pas de changement de l'état du système local.

État Connecter : dans cet état, le FSM BGP FSM attend que la connexion TCP soit achevée.

Les événements de démarrage (Événements 1, 3 à 7) sont ignorés dans l'état Connecter.

En réponse à un événement ManualStop (événement 2), le système local :
- abandonne la connexion TCP,
- libère toutes les ressources BGP,
- règle le compteur ConnectRetryCounter à zéro,
- arrête le temporisateur ConnectRetryTimer et le règle à zéro, et
- change son état en Repos.

En réponse à l'événement ConnectRetryTimer_Expires (événement 9) le système local :
- abandonne la connexion TCP,
- redémarre le temporisateur ConnectRetryTimer,
- arrête le temporisateur DelayOpenTimer et le remet à zéro,
- initie une connexion TCP avec l'autre homologue BGP,
- continue d'écouter si une connexion peut être initiée par l'homologue BGP distant, et
- reste dans l'état Connecter.

Si l'événement DelayOpenTimer_Expires (événement 12) se produit dans l'état Connecter, le système local :
- envoie un message OPEN à son homologue,
- règle le temporisateur HoldTimer à une grande valeur, et
- change son état pour OuvertEnvoyé.

Si le FSM BGP reçoit un événement TcpConnection_Valid (événement 14) la connexion TCP est traitée, et la connexion reste dans l'état Connecter.

Si le FSM BGP reçoit un événement Tcp_CR_Invalid (événement 15) le système local rejette la connexion TCP, et la connexion reste dans l'état Connecter.

Si la connexion TCP réussit (événement 16 ou événement 17) le système local vérifie l'attribut DelayOpen avant de poursuivre. Si l'attribut DelayOpen est réglé à VRAI, le système local :
- arrête le temporisateur ConnectRetryTimer (si il court) et le règle à zéro,
- règle le temporisateur DelayOpenTimer à la valeur initiale, et
- reste dans l'état Connecter.

Si l'attribut DelayOpen est réglé à FAUX, le système local :
- arrête le temporisateur ConnectRetryTimer (si il court) et le règle à zéro,
- achève l'initialisation de BGP,
- envoie un message OPEN à son homologue,
- règle le temporisateur HoldTimer à une grande valeur, et
- change son état en OuvertEnvoyé.

Une valeur de HoldTimer de 4 minutes est suggérée.

Si la connexion TCP échoue (événement 18) le système local vérifie le temporisateur DelayOpenTimer. Si DelayOpenTimer court, le système local :
- redémarre le temporisateur ConnectRetryTimer avec la valeur initiale,
- arrête le temporisateur DelayOpenTimer et remet sa valeur à zéro,
- continue d'écouter si une connexion peut être initiée par l'homologue BGP distant, et
- change son état en Actif.

Si le temporisateur DelayOpenTimer ne court pas, le système local :
- arrête le temporisateur ConnectRetryTimer à zéro,
- abandonne la connexion TCP,
- libère toutes les ressources BGP, et
- change son état en Repos.

Si un message OPEN est reçu pendant que le temporisateur DelayOpenTimer court (événement 20) le système local :
- arrête le temporisateur ConnectRetryTimer (si il court) et le règle à zéro,
- achève l'initialisation de BGP,
- arrête et libère le temporisateur DelayOpenTimer (met sa valeur à zéro),
- envoie un message OPEN,
- envoie un message KEEPALIVE ;
si la valeur initiale de HoldTimer n'est pas zéro,
- démarre le KeepaliveTimer avec la valeur initiale et
- remet le HoldTimer à la valeur négociée ;
autrement, si la valeur initiale du HoldTimer est zéro,
- relance le KeepaliveTimer et
- remet la valeur du HoldTimer à zéro,
- et change son état en OuvertConfirmé.

Si la valeur du champ Système autonome est la même que le numéro du système autonome local, régler le statut de la connexion à "connexion interne" ; autrement il va être "externe".

Si la vérification de l'en-tête de message BGP (événement 21) ou la vérification du message OPEN détecte une erreur (événement 22) (voir le paragraphe 6.2) le système local :
- (facultativement) si l'attribut SendNOTIFICATIONwithoutOPEN est réglé à VRAI, le système local envoie alors d'abord un message NOTIFICATION avec le code d'erreur approprié, et ensuite
- arrête le temporisateur ConnectRetryTimer (si il court) et le règle à zéro,
- libère toutes les ressources BGP,
- abandonne la connexion TCP,
- incrémente le ConnectRetryCounter de 1,
- (facultativement) effectue une atténuation d'oscillation d'homologue si l'attribut DampPeerOscillations est réglé à VRAI,
- et change son état en Repos.

Si un message NOTIFICATION est reçu avec une erreur de version (événement 24) le système local vérifie le temporisateur DelayOpenTimer. Si il court, le système local :
- arrête le temporisateur ConnectRetryTimer (si il court) et le règle à zéro,
- arrête et remet à zéro le temporisateur DelayOpenTimer,
- libère toutes les autres ressources BGP,
- abandonne la connexion TCP, et
- change son état en Repos.

Si le temporisateur DelayOpenTimer ne court pas, le système local :
- arrête le temporisateur ConnectRetryTimer et le règle à zéro,
- libère toutes les ressources BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue l'atténuation d'oscillation d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

En réponse à tout autre événement (événements 8, 10, 11, 13, 19, 23, 25 à 28) le système local :
- si le temporisateur ConnectRetryTimer court, l'arrête et le remet à zéro,
- si le temporisateur DelayOpenTimer court, l'arrête et le remet à zéro,
- libère toutes les ressources BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue l'atténuation d'oscillation d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

État Actif : dans cet état, le FSM BGP essaye d'acquérir un homologue en écoutant, et acceptant, une connexion TCP.

Les événements de démarrage (événements 1, 3 à 7) sont ignorés dans l'état Actif.

En réponse à un événement ManualStop (événement 2), le système local :
- Si le temporisateur DelayOpenTimer court et si l'attribut de session SendNOTIFICATIONwithoutOPEN est établi, le système local envoie une NOTIFICATION avec Cessation,
- libère toutes les ressources BGP incluant d'arrêter le temporisateur DelayOpenTimer,
- abandonne la connexion TCP,
- règle le compteur ConnectRetryCounter à zéro,
- arrête le temporisateur ConnectRetryTimer et le règle à zéro, et
- change son état en Repos.

En réponse à un événement ConnectRetryTimer_Expires (événement 9) le système local :
- redémarre le temporisateur ConnectRetryTimer (avec la valeur initiale),
- initie une connexion TCP avec l'autre homologue BGP,
- continue d'écouter si une connexion TCP peut être initiée par l'homologue BGP distant, et
- change son état en Connecter.

Si le système local reçoit un événement DelayOpenTimer_Expires (événement 12) il doit :
- régler le temporisateur ConnectRetryTimer à zéro,
- arrêter et régler à zéro le temporisateur DelayOpenTimer,
- achever l'initialisation de BGP,
- envoyer le message OPEN à son homologue distant,
- régler son temporisateur de garde à une grande valeur, et
- change son état en OuvertEnvoyé.

Une valeur de HoldTimer de quatre minutes est aussi suggérée pour cette transition d'état.

Si le système local reçoit un événement TcpConnection_Valid (événement 14) il traite les fanions de connexion TCP et reste dans l'état Actif.

Si le système local reçoit un événement Tcp_CR_Invalid (événement 15) il doit rejeter la connexion TCP et rester dans l'état Actif.

En réponse à la réussite d'une connexion TCP (événement 16 ou Événement 17) le système local vérifie l'attribut facultatif DelayOpen avant de la traiter.

Si l'attribut DelayOpen est réglé à VRAI, le système local :
- arrête le temporisateur ConnectRetryTimer et le règle à zéro,

- règle le temporisateur DelayOpenTimer à la valeur initiale (DelayOpenTime), et
- rese dans l'état Actif.

Si l'attribut DelayOpen est réglé à FAUX, le système local :
- règle le temporisateur ConnectRetryTimer à zéro,
- achève l'initialisation de BGP,
- envoie le message OPEN à son homologue,
- règle son temporisateur HoldTimer à une grande valeur, et
- change son état en OuvertEnvoyé.

Une valeur de HoldTimer de quatre minutes est suggérée comme "grande valeur" pour HoldTimer.

Si le système local reçoit un événement TcpConnectionFails (événement 18) il va :
- redémarrer le temporisateur ConnectRetryTimer (avec la valeur initiale),
- arrêter et mettre à zéro le temporisateur DelayOpenTimer,
- libérer toutes les ressources de BGP,
- incrémenter le compteur ConnectRetryCounter de 1,
- effectuer facultativement l'atténuation d'oscillation d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- changer son état en Repos.

Si un message OPEN est reçu et si le temporisateur DelayOpenTimer court (événement 20) le système local :
- arrête le temporisateur ConnectRetryTimer (si il court) et le règle à zéro,
- arrête et règle à zéro le temporisateur DelayOpenTimer,
- achève l'initialisation de BGP,
- envoie un message OPEN,
- envoie un message KEEPALIVE ;

si la valeur du temporisateur HoldTimer n'est pas zéro,
- démarre le temporisateur KeepaliveTimer à la valeur initiale,
- remet le temporisateur HoldTimer à la valeur négociée ;

autrement, si le temporisateur HoldTimer est à zéro,
- remet le temporisateur KeepaliveTimer à zéro,
- remet le temporisateur HoldTimer à zéro, et
- change son état en OuvertConfirmé.

Si la valeur du champ Système autonome est la même que celle du numéro du système autonome local, régler le statut de la connexion à "connexion interne"; autrement ce sera externe.

Si la vérification d'en-tête de message BGP (événement 21) ou du message OPEN détecte une erreur (événement 22) (voir le paragraphe 6.2) le système local :
- envoie (facultativement) un message NOTIFICATION avec le code d'erreur approprié si l'attribut SendNOTIFICATIONwithoutOPEN est réglé à VRAI,
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si un message NOTIFICATION est reçu avec une erreur de version (événement 24) le système local vérifie le temporisateur DelayOpenTimer. Si il court, le système local :
- arrête le temporisateur ConnectRetryTimer (si il court) et le règle à zéro,
- arrête et remet à zéro le temporisateur DelayOpenTimer,
- libère toutes les ressources BGP,
- abandonne la connexion TCP, et
- change son état en Repos.

Si le temporisateur DelayOpenTimer ne court pas, le système local :
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

En réponse à tout autre événement (événements 8, 10, 11, 13, 19, 23, 25 à 28) le système local :
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de un,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

État OuvertEnvoyé : dans cet état, le FSM BGP attend un message OPEN de son homologue.
Les événements de démarrage (événements 1, 3 à 7) sont ignorés dans l'état OuvertEnvoyé.
Si un événement ManualStop (événement 2) est produit dans l'état OuvertEnvoyé, le système local :
- envoie la NOTIFICATION avec une Cessation,
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources BGP,
- abandonne la connexion TCP,
- règle le compteur ConnectRetryCounter à zéro, et
- change son état en Repos.

Si un événement AutomaticStop (événement 8) est produit dans l'état OuvertEnvoyé, le système local :
- envoie la NOTIFICATION avec une Cessation,
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si le temporisateur de garde arrive à expiration (événement 10, HoldTimer_Expires) le système local :
- envoie un message NOTIFICATION avec le code d'erreur "Temporisateur de garde expiré",
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si un événement TcpConnection_Valid (événement 14), Tcp_CR_Acked (événement 16), ou TcpConnectionConfirmed (événement 17) est reçu, une seconde connexion TCP peut être en cours. Cette seconde connexion TCP est retracée selon le processus de collision de connexions (paragraphe 6.8) jusqu'à ce qu'un message OPEN soit reçu.

Une demande de connexion TCP pour un accès invalide (Tcp_CR_Invalid (événement 15)) est ignorée.

Si un événement TcpConnectionFails (événement 18) est reçu, le système local :
- clôt la connexion BGP,
- redémarre le temporisateur ConnectRetryTimer,
- continue d'écouter si une connexion peut être initiée par l'homologue BGP distant, et
- change son état en Actif.

Quand un message OPEN est reçu, la correction de tous les champs est vérifiée. Si il n'y a pas d'erreur dans le message OPEN (événement 19) le système local :
- remet le temporisateur DelayOpenTimer à zéro,
- règle le temporisateur BGP ConnectRetryTimer à zéro,
- envoie un message KEEPALIVE, et
- établit un temporisateur KeepaliveTimer (via le texte ci-dessous),
- établit le temporisateur HoldTimer conformément à la valeur négociée (voir le paragraphe 4.2),
- change son état en OuvertConfirmé.

Si la valeur de garde négociée est zéro, les temporisateurs HoldTimer et KeepaliveTimer ne sont pas démarrés. Si la valeur du champ Système autonome est la même que celle du numéro de système autonome local, la connexion est alors une connexion "interne" ; autrement, c'est une connexion "externe". (Cela va avoir un impact sur le traitement de UPDATE comme décrit plus loin.)

Si la vérification de l'en-tête de message BGP (événement 21) ou du message OPEN détecte une erreur (événement 22) (voir le paragraphe 6.2) le système local :
- envoie un message NOTIFICATION avec le code d'erreur approprié,
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Les mécanismes de détection de collision (paragraphe 6.8) n'ont pas besoin d'être appliqués quand un message OPEN valide est reçu (événement 19 ou 20). Se référer au paragraphe 6.8 pour les détails de la comparaison. Un événement CollisionDetectDump se produit quand la mise en œuvre BGP détermine, par des moyens qui sortent du domaine d'application du présent document, qu'une collision de connexions s'est produite.

Si il est déterminé que la connexion qui doit être close est dans l'état OuvertEnvoyé, un événement OpenCollisionDump (événement 23) est signalé à l'automate. Si un tel événement est reçu dans l'état OuvertEnvoyé, le système local :
- envoie une NOTIFICATION avec une Cessation,
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si un message NOTIFICATION est reçu avec une erreur de version (événement 24) le système local :
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP, et
- change son état en Repos.

En réponse à tout autre événement (événements 9, 11 à 13, 20, 25 à 28) le système local :
- envoie la NOTIFICATION avec le code d'erreur "Erreur de code d'automate à états finis",
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

État OuvertConfirmé : dans cet état, BGP attend un message KEEPALIVE ou NOTIFICATION.
Tout événement de démarrage (événements 1, 3 à 7) est ignoré dans l'état OuvertConfirmé.
En réponse à un événement ManualStop (événement 2) initié par l'opérateur, le système local :
- envoie le message NOTIFICATION avec une Cessation,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- règle le temporisateur ConnectRetryCounter à zéro,
- règle le temporisateur ConnectRetryTimer à zéro, et
- change don état en Repos.

En réponse à l'événement AutomaticStop initié par le système (événement 8) le système local :
- envoie le message NOTIFICATION avec une Cessation,
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si l'événement HoldTimer_Expires (événement 10) se produit avant qu'un message KEEPALIVE soit reçu, le système local :
- envoie le message NOTIFICATION avec le code d'erreur "Temporisateur de garde expiré",
- règle le temporisateur ConnectRetryTimer à zéro,

- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si le système local reçoit un événement KeepaliveTimer_Expires (événement 11) le système local :
- envoie un message KEEPALIVE,
- redémarre le temporisateur KeepaliveTimer, et
- reste dans l'état OuvertConfirmé.

Dans le cas d'un événement TcpConnection_Valid (événement 14), ou en cas de succès d'une connexion TCP (événements 16 ou 17) dans l'état Confirmé, le système local a besoin de tracer la seconde connexion.

Si une connexion TCP est tentée avec un accès invalide (événement 15) le système local va ignorer la seconde tentative de connexion.

Si le système local reçoit un événement TcpConnectionFails (événement 18) du TCP sous-jacent ou un message NOTIFICATION (événement 25) le système local :
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si le système local reçoit un message NOTIFICATION avec une erreur de version (NotifMsgVerErr (événement 24)) le système local :
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP, et
- change son état en Repos.

Si le système local reçoit un message OPEN valide (BGPOpen (événement 19)) la fonction de détection de collision est traitée selon le paragraphe 6.8. Si cette connexion doit être abandonnée à cause d'une collision de connexions, le système local :
- envoie une NOTIFICATION avec une Cessation,
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP (envoie un TCP FIN),
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si un message OPEN est reçu, la correction de tous les champs est vérifiée. Si la vérification de l'en-tête de message BGP (BGPHeaderErr (événement 21)) ou du message OPEN détecte une erreur (voir le paragraphe 6.2) (BGPOpenMsgErr (événement 22)) le système local :
- envoie un message NOTIFICATION avec le code d'erreur approprié,
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si, durant le traitement d'un autre message OPEN, la mise en œuvre BGP détermine, par des moyens qui sortent du domaine d'application du présent document, qu'une collision de connexions s'est produite et que cette connexion doit être close, le système local va produire un événement OpenCollisionDump (événement 23). Quand le système local reçoit un événement OpenCollisionDump (événement 23) le système local :
- envoie une NOTIFICATION avec une Cessation,
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,

- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si le système local reçoit un message KEEPALIVE (KeepAliveMsg (événement 26)) le système local :
- redémarre le temporisateur HoldTimer et
- change son état en Établi.

En réponse à tout autre événement (événements 9, 12, 13, 20, 27, 28) le système local :
- envoie une NOTIFICATION avec un code de "Erreur d'automate à états finis",
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

État Établi : dans l'état Établi, le FSM BGP peut échanger des messages UPDATE, NOTIFICATION, et KEEPALIVE avec son homologue.
Tout événement de démarrage (événements 1, 3 à 7) est ignoré dans l'état Établi.
En réponse à un événement ManualStop (initié par un opérateur) (événement 2) le système local :
- envoie le message NOTIFICATION avec une Cessation,
- règle le temporisateur ConnectRetryTimer à zéro,
- supprime toutes les routes associées à cette connexion,
- libère les ressources de BGP,
- abandonne la connexion TCP,
- règle le compteur ConnectRetryCounter à zéro, et
- change son état en Repos.

En réponse à un événement AutomaticStop (événement 8) le système local :
- envoie une NOTIFICATION avec une Cessation,
- règle le temporisateur ConnectRetryTimer à zéro,
- supprime toutes les routes associées à cette connexion,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Une raison d'un événement AutomaticStop est qu'un BGP reçoit un message UPDATE avec un nombre de préfixes pour un certain homologue tel que le total des préfixes reçus excède le nombre maximum de préfixes configuré. Le système local déconnecte automatiquement l'homologue.

Si l'événement HoldTimer_Expires se produit (événement 10) le système local :
- envoie un message NOTIFICATION avec le code d'erreur "Temporisateur de garde expiré",
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si l'événement KeepaliveTimer_Expires se produit (événement 11) le système local :
- envoie un message KEEPALIVE, et
- redémarre son temporisateur KeepaliveTimer, sauf si la valeur de HoldTime négociée est zéro.

Chaque fois que le système local envoie un message KEEPALIVE ou UPDATE, il redémarre son temporisateur KeepaliveTimer, sauf si la valeur de HoldTime négociée est zéro.
Un événement TcpConnection_Valid (événement 14) reçu pour un accès valide, va causer le traçage de la seconde connexion. Une connexion TCP invalide (événement Tcp_CR_Invalid (événement 15)) va être ignorée.
En réponse à l'indication que l'établissement de la connexion TCP est réussi (événement 16 ou 17) la seconde connexion DEVRA être tracée jusque à ce qu'il envoie un message OPEN.

Si un message OPEN valide (BGPOpen (événement 19)) est reçu, et si l'attribut facultatif CollisionDetectEstablishedState est VRAI, le message OPEN va être vérifié pour voir si il entre en collision (paragraphe 6.8) avec une autre connexion. Si la mise en œuvre BGP détermine que cette connexion doit être terminée, elle va traiter un événement OpenCollisionDump (événement 23). Si cette connexion doit être terminée, le système local :
- envoie une NOTIFICATION avec une Cessation,
- règle le temporisateur ConnectRetryTimer à zéro,
- supprime toutes les routes associées à cette connexion,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

Si le système local reçoit un message NOTIFICATION (événement 24 ou 25) ou un TcpConnectionFails (événement 18) du TCP sous-jacent, le système local :
- règle le temporisateur ConnectRetryTimer à zéro,
- supprime toutes les routes associées à cette connexion,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- change son état en Repos.

Si le système local reçoit un message KEEPALIVE (événement 26) il va :
- redémarrer son temporisateur HoldTimer, si la valeur négociée de HoldTime n'est pas zéro, et
- rester dans l'état Établi.

Si le système local reçoit un message UPDATE (événement 27) il va :
- traiter le message,
- redémarrer son temporisateur HoldTimer, si la valeur négociée de HoldTime n'est pas zéro, et
- rester dans l'état Établi.

Si le système local reçoit un message UPDATE, et si la procédure de traitement d'erreur de message UPDATE (voir le paragraphe 6.3) détecte une erreur (événement 28) le système local :
- envoie un message NOTIFICATION avec une erreur de mise à jour,
- règle le temporisateur ConnectRetryTimer à zéro,
- supprime toutes les routes associées à cette connexion,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

En réponse à tout autre événement (événements 9, 12, 13, 20 à 22) le système local :
- envoie un message NOTIFICATION avec le code d'erreur "Erreur d'automate à états finis",
- supprime toutes les routes associées à cette connexion,
- règle le temporisateur ConnectRetryTimer à zéro,
- libère toutes les ressources de BGP,
- abandonne la connexion TCP,
- incrémente le compteur ConnectRetryCounter de 1,
- effectue (facultativement) l'atténuation d'oscillations d'homologue si l'attribut DampPeerOscillations est réglé à VRAI, et
- change son état en Repos.

9. Traitement du message UPDATE

Un message UPDATE ne peut être reçu que dans l'état Établi. Recevoir un message UPDATE dans tout autre état est une erreur. Quand un message UPDATE est reçu, la validité de chaque champ est vérifiée, comme spécifié au paragraphe 6.3.

Si un attribut facultatif non transitif n'est pas reconnu, il est ignoré en silence. Si un attribut facultatif transitif n'est pas reconnu, le bit Partiel (le troisième bit de poids fort) dans l'octet des fanions d'attribut est réglé à 1, et l'attribut est conservé pour propagation aux autres locuteurs BGP.

Si un attribut facultatif est reconnu et a une valeur valide, alors, selon le type d'attribut facultatif, il est traité en local, conservé, et mis à jour, si nécessaire, pour une possible propagation aux autres locuteurs BGP.

Si le message UPDATE contient un champ Routes retirées non vide, les routes annoncées précédemment, dont les destinations (exprimées comme des préfixes IP) sont contenues dans ce champ, DEVRONT être retirées de la Adj-RIB-In. Ce locuteur BGP DEVRA faire fonctionner son processus de décision parce que la route annoncée précédemment n'est plus disponible à l'utilisation.

Si le message UPDATE contient une route faisable, la Adj-RIB-In va être mise à jour avec cette route comme suit : si les NLRI de la nouvelle route sont identiques à celles que la route avait actuellement mémorisée dans la Adj-RIB-In, la nouvelle route DEVRA alors remplacer la route plus ancienne dans la Adj-RIB-In, retirant donc implicitement la route plus ancienne du service. Autrement, si la Adj-RIB-In n'a pas de route avec des NLRI identiques à celles de la nouvelle route, la nouvelle route DEVRA être placée dans la Adj-RIB-In.

Une fois que le locuteur BGP a mis à jour la Adj-RIB-In, il DEVRA faire fonctionner son processus de décision.

9.1 Processus de décision

Le processus de décision choisit des routes à annoncer ensuite en appliquant les politiques de la base de données d'information de politique (PIB, *Policy Information Base*) locale aux routes mémorisées dans sa Adj-RIBs-In. Le résultat du processus de décision est l'ensemble des routes qui vont être annoncées aux homologues ; les routes choisies vont être mémorisées dans la Adj-RIBs-Out du locuteur local, conformément à la politique.

Le processus de décision BGP décrit ici est conceptuel, et n'a pas à être mis en œuvre précisément comme décrit, pour autant que les mises en œuvre prennent en charge la fonctionnalité décrite et qu'elles présentent le même comportement visible extérieurement.

Le processus de choix est formalisé en définissant une fonction qui prend l'attribut d'une certaine route comme argument et retourne soit (a) un entier non négatif notant le degré de préférence de la route, soit (b) une valeur notant que cette route est inéligible à l'installation dans la Loc-RIB et va être exclue de la prochaine phase du choix de routes.

La fonction qui calcule le degré de préférence pour une certaine route NE DEVRA PAS utiliser comme entrée ce qui suit : l'existence d'autres routes, la non existence d'autres routes, ou les attributs de chemin d'autres routes. Le choix de route consiste alors en l'application individuelle de la fonction du degré de préférence à chaque route faisable, suivie par le choix d'une de celles qui ont le plus haut degré de préférence.

Le processus de décision opère sur les routes contenues dans la Adj-RIBs-In, et est chargé de :
- choisir les routes qui vont être utilisées en local par le locuteur,
- choisir les routes qui vont être annoncées aux autres homologues BGP,
- l'agrégation de routes et la réduction des informations de routes.

Le processus de décision a lieu en trois phases distinctes, déclenchées chacune par un événement différent :

a) La phase 1 est chargée de calculer le degré de préférence pour chaque route reçue d'un homologue.

b) La phase 2 est invoquée à l'achèvement de la phase 1. Elle est chargée du choix de la meilleure route parmi toutes celles disponibles pour chaque destination distincte, et d'installer chaque route choisie dans la Loc-RIB.

c) La phase 3 est invoquée après la modification de la Loc-RIB. Elle est chargée de disséminer les routes de la Loc-RIB à chaque homologue, conformément aux politiques contenues dans la PIB. L'agrégation de routes et la réduction des informations peuvent facultativement être effectuées au sein de cette phase.

9.1.1 Phase 1 : calcul du degré de préférence

La phase 1 de la fonction de décision est invoquée chaque fois que le locuteur BGP local reçoit d'un homologue un message UPDATE qui annonce une nouvelle route, un remplacement de route, ou des retraits de routes.

La phase 1 de la fonction de décision est un processus séparé, qui s'achève quand il n'y a plus d'autres tâches.

La phase 1 de la fonction de décision verrouille une Adj-RIB-In avant d'opérer sur toute route contenue en son sein, et la déverrouille après avoir opéré sur toutes les routes nouvelles ou infaisables contenues en son sein.

Pour chaque route nouvellement reçue ou faisable de remplacement, le locuteur BGP local détermine un degré de préférence comme suit :

Si la route est apprise d'un homologue interne, soit la valeur de l'attribut LOCAL_PREF est prise comme le degré de préférence, soit le système local calcule le degré de préférence de la route sur la base d'informations préconfigurées de politique. Noter que cette dernière solution peut résulter en la formation de boucles d'acheminement persistantes.

Si la route est apprise d'un homologue externe, le locuteur BGP local calcule alors le degré de préférence sur la base d'informations de politique préconfigurées. Si la valeur retournée indique que la route est inéligible, la route NE PEUT PAS servir d'entrée à la prochaine phase de choix de route ; autrement, la valeur retournée DOIT être utilisée comme valeur de LOCAL_PREF dans toute réannonce de IBGP.

La nature exacte de ces informations de politique, et le calcul impliqué, sont une affaire locale.

9.1.2 Phase 2 : choix du chemin

La phase 2 de la fonction de décision est invoquée à l'achèvement de la phase 1. La fonction de phase 2 est un processus séparé qui s'achève quand ses tâches sont terminées. Le processus de phase 2 considère toutes les routes qui sont éligibles dans la Adj-RIBs-In.

La phase 2 de la fonction de décision est bloquée quand la phase 3 de la fonction de décision est en cours. La fonction de phase 2 bloque toutes les Adj-RIBs-In avant de commencer sa fonction, et les débloque quand elle a fini.

Si l'attribut NEXT_HOP d'une route BGP décrit une adresse qui ne peut pas se résoudre, ou si elle va devenir non résoluble si la route a été installée dans le tableau d'acheminement, la route BGP DOIT être exclue de la phase 2 de la fonction de décision.

Si l'attribut AS_PATH d'une route BGP contient une boucle d'AS, la route BGP devrait être exclue de phase 2 de la fonction de décision. La détection de boucle d'AS est faite en examinant le chemin d'AS complet (comme spécifié dans l'attribut AS_PATH) et en vérifiant que le numéro de système autonome du système local n'apparaît pas dans le chemin d'AS. Le fonctionnement d'un locuteur BGP qui est configuré à accepter des routes ayant son propre numéro de système autonome dans le chemin d'AS sortent du domaine d'application du présent document.

Il est critique que les locuteurs BGP au sein d'un AS ne prennent pas de décisions contradictoires à l'égard du choix de route qui causeraient la formation de boucles d'acheminement.

Pour chaque ensemble de destinations dans lequel existe une route faisable dans la Adj-RIBs-In, le locuteur BGP local identifie la route qui a :
a) le plus haut degré de préférence parmi toutes les routes du même ensemble de destinations, ou
b) est la seule route pour cette destination, ou
c) est choisie par suite des règles de départage de phase 2 spécifiées au paragraphe 9.1.2.2.

Le locuteur local DEVRA alors installer cette route dans la Loc-RIB, remplaçant toute route pour la même destination actuellement détenue dans la Loc-RIB. Quand la nouvelle route BGP est installée dans le tableau d'acheminement, il faut veiller à s'assurer que les routes existantes pour la même destination qui sont maintenant considérées comme invalides sont retirées du tableau d'acheminement. Il dépend de la politique configurée chez le locuteur BGP que la nouvelle route BGP remplace une route non BGP existante dans le tableau d'acheminement.

Le locuteur local DOIT déterminer l'adresse du prochain bond immédiat à partir de l'attribut NEXT_HOP de la route choisie (voir le paragraphe 5.1.3). Si le prochain bond immédiat ou le coût IGP pour le NEXT_HOP (où le NEXT_HOP est résolu par une route IGP) change, la phase 2 du choix de route DOIT être effectuée à nouveau.

Noter que même si des routes BGP n'ont pas à être installées dans le tableau d'acheminement avec le ou les prochains bonds immédiats, les mises en œuvre DOIVENT veiller à ce que, avant que des paquets soient transmis sur la route BGP, son adresse de NEXT_HOP associée soit résolue comme l'adresse de prochain bond immédiat (directement connecté) et que cette adresse (ou ces adresses) soient finalement utilisées pour la transmission réelle de paquet.

Les routes non résolubles DEVRONT être retirées de la Loc-RIB et du tableau d'acheminement. Cependant, les routes non résolubles correspondantes DEVRAIENT être conservées dans la Adj-RIBs-In (au cas où elles deviendraient résolubles).

9.1.2.1 Condition de résolution de chemin

Comme indiqué au paragraphe 9.1.2, les locuteurs BGP DEVRAIENT exclure les routes non résolubles de la phase 2 de décision. Cela assure que seules des routes valides sont installées dans la Loc-RIB et le tableau d'acheminement.

La condition de résolubilité est définie comme suit :

1) Une route Rte1, faisant référence à la seule adresse de réseau intermédiaire, est considérée comme résoluble si le tableau d'acheminement contient au moins une route Rte2 résoluble qui correspond à l'adresse de réseau intermédiaire de Rte1 et n'est pas résolue de façon récurrente (directement ou indirectement) à travers Rte1. Si plusieurs routes qui correspondent sont disponibles, seule la route avec la plus longue correspondance DEVRAIT être prise en compte.

2) Les routes qui font référence à des interfaces (avec ou sans adresses intermédiaires) sont considérées comme résolubles si l'état de l'interface référencée est actif et si le traitement IP est activé sur cette interface.

Les routes BGP ne se réfèrent pas aux interfaces, mais peuvent être résolues par les routes dans le tableau d'acheminement qui peuvent être de deux types (celles qui spécifient des interfaces ou celles qui ne le font pas). Les routes IGP et les routes pour les réseaux directement connectés sont supposées spécifier l'interface sortante. Les routes statiques peuvent spécifier l'interface sortante, l'adresse intermédiaire, ou les deux.

Noter qu'une route BGP est considérée comme non résoluble dans une situation où le tableau d'acheminement du locuteur BGP ne contient aucune route correspondant au prochain bond de la route BGP. Les routes mutuellement récurrentes (routes qui se résolvent l'une l'autre ou elles-mêmes) échouent aussi à la vérification de résolubilité.

Il est aussi important que les mises en œuvre ne prennent pas en compte les routes faisables qui deviendraient non résolubles si elles étaient installées dans le tableau d'acheminement, même si leur prochain bond est résoluble en utilisant le contenu actuel du tableau d'acheminement (un exemple de telles routes serait celui de routes mutuellement récurrentes). Cette vérification assure qu'un locuteur BGP n'installe pas dans le tableau d'acheminement de routes qui vont être retirées et non utilisées par le locuteur. Donc, en plus de la stabilité du tableau d'acheminement local, cette vérification améliore aussi le comportement du protocole dans le réseau.

Chaque fois qu'un locuteur BGP identifie une route qui échoue à la vérification de résolubilité à cause de la récurrence mutuelle, un message d'erreur DEVRAIT être enregistré.

9.1.2.2 Départage (phase 2)

Dans son Adj-RIBs-In, un locuteur BGP peut avoir plusieurs routes pour la même destination qui ont le même degré de préférence. Le locuteur local peut choisir une seule de ces routes pour l'inclure dans la Loc-RIB associée. Le locuteur local considère toutes les routes ayant le même degré de préférence, celles reçue des homologues internes, et celles reçues des homologues externes.

La procédure de départage suivante suppose que, pour chaque route candidate, tous les locuteurs BGP au sein d'un système autonome peuvent certifier le coût d'un chemin (distance intérieure) à l'adresse décrite par l'attribut NEXT_HOP de la route, et suivre le même algorithme de choix de route.

L'algorithme de départage commence par considérer toutes les routes également préférables pour la même destination, et choisir ensuite les routes qui vont être supprimées de la prise en compte. L'algorithme se termine aussitôt qu'une seule route reste en considération. Les critères DOIVENT être appliqués dans l'ordre spécifié.

Plusieurs des critères sont décrits en utilisant un pseudo-code. Noter que le pseudo-code montré a été choisi pour sa clarté, et non pour son efficacité. Il n'est pas destiné à spécifier une mise en œuvre particulière. Les mises en œuvre de BGP PEUVENT utiliser tout algorithme qui produit les mêmes résultats que ceux décrits ici.

a) Retirer du champ de considération toutes les routes qui ne sont pas parmi celles qui ont le plus petit nombre de numéros d'AS présents dans leur attribut AS_PATH. Noter que quand on compte ce nombre, un AS_SET compte pour 1, quel que soit le nombre d'AS de l'ensemble.

b) Retirer du champ de considération toutes les routes qui n'ont pas le plus faible numéro d'origine dans leur attribut Origine.

c) Retirer du champ de considération les routes qui ont des attributs MULTI_EXIT_DISC moins préférés. MULTI_EXIT_DISC n'est comparable qu'entre des routes apprises du même AS voisin (l'AS voisin est déterminé par l'attribut AS_PATH). Les routes qui n'ont pas l'attribut MULTI_EXIT_DISC sont considérées comme ayant la plus petite valeur possible de MULTI_EXIT_DISC.

Ceci est aussi décrit dans la procédure suivante :

```
pour m = toutes les routes encore considérées
   pour n = toutes les routes encore considérées
      si (ASvoisin(m) == ASvoisin(n)) et (MED(n) < MED(m))
         retirer la route m de considération
```

Dans le pseudo-code ci-dessus, MED(n) est une fonction qui retourne la valeur de l'attribut MULTI_EXIT_DISC de la route n. Si la route n n'a pas d'attribut MULTI_EXIT_DISC, la fonction retourne la plus basse valeur possible de MULTI_EXIT_DISC (c'est-à-dire, 0).

De même, ASvoisin(n) est une fonction qui retourne l'AS voisin à partir duquel la route a été reçue. Si la route est apprise via IBGP, et si l'autre locuteur IBGP n'a pas généré la route, il est l'AS voisin duquel l'autre locuteur IBGP a appris la route. Si la route est apprise via IBGP, et si l'autre locuteur IBGP a soit (a) généré la route, soit (b) créé la route par agrégation et si l'attribut AS_PATH de la route agrégée est soit vide, soit commence par un AS_SET, c'est l'AS local.

Si un attribut MULTI_EXIT_DISC est retiré avant de réannoncer une route dans IBGP, la comparaison sur la base de l'attribut EBGP MULTI_EXIT_DISC reçu PEUT encore être effectuée. Si une mise en œuvre choisit de retirer MULTI_EXIT_DISC, la comparaison facultative sur MULTI_EXIT_DISC, si elle est effectuée, DOIT alors être effectuée seulement parmi les routes apprises par EBGP. La meilleure route apprise par EBGP peut alors être comparée avec les routes apprises par IBGP après la suppression de l'attribut MULTI_EXIT_DISC. Si MULTI_EXIT_DISC est retiré d'un sous ensemble de routes apprises par EBGP, et si la "meilleure" route choisie apprise par EBGP n'a pas son attribut MULTI_EXIT_DISC retiré, le MULTI_EXIT_DISC doit alors être utilisé dans la comparaison avec les routes apprises par IBGP. Pour les routes apprises par IBGP, l'attribut MULTI_EXIT_DISC DOIT être utilisé dans les comparaisons de routes qui atteignent cette étape du processus de décision. Inclure l'attribut MULTI_EXIT_DISC d'une route apprise par EBGP dans la comparaison avec une route apprise par IBGP, puis retirer l'attribut MULTI_EXIT_DISC, et annoncer la route s'est révélé causer des routes en boucle.

d) Si au moins une des routes candidates a été reçue via EBGP, retirer de la prise en considération toutes les routes qui ont été reçues via IBGP.

e) Retirer de la prise en considération toutes les routes qui ont un coût intérieur moins préféré. Le coût intérieur d'une route est déterminé en calculant la métrique du prochain bond pour la route en utilisant le tableau d'acheminement. Si le bond NEXT_HOP pour une route est accessible, mais si aucun coût ne peut être déterminé, cette étape devrait être sautée (équivalent à considérer que toutes les routes ont un coût égal).

   Ceci est aussi décrit dans la procédure suivante :

```
pour m = toutes les routes encore considérées
   pour n = toutes les routes encore considérées
      si (coût(n) est inférieur à coût(m))
         retirer m de la prise en considération
```

   Dans le pseudo-code ci-dessus, coût(n) est une fonction qui retourne le coût du chemin (distance intérieure) de l'adresse donnée dans l'attribut NEXT_HOP de la route.

f) Retirer de la prise en considération toutes les routes autres que la route qui a été annoncée par le locuteur BGP avec la plus faible valeur d'identifiant BGP.

g) Préférer la route reçue de la plus basse adresse d'homologue.

9.1.3 Phase 3 : dissémination de chemin

La phase 3 de la fonction de décision est invoquée à l'achèvement de la phase 2, ou lorsque un des événements suivants se produit :

a) quand les routes dans la Loc-RIB pour les destinations locales ont changé,
b) quand des routes générées en local apprise par des moyens hors BGP ont changé,
c) quand une nouvelle connexion de locuteur BGP a été établie.

La phase 3 de la fonction est un processus séparé qui s'achève quand ses tâches sont terminées. La phase 3 de la fonction de décision d'acheminement est bloquée pendant qu'est en cours la phase 2 de la fonction de décision.

Toutes les routes dans la Loc-RIB sont traitées dans la Adj-RIBs-Out conformément à la politique configurée. Cette politique PEUT exclure l'installation d'une route de la Loc-RIB dans une Adj-RIB-Out particulière. Une route NE DEVRA PAS être installée dans la Adj-Rib-Out sauf si la destination, et le prochain bond décrit par cette route, peuvent être transmis de façon appropriée par le tableau d'acheminement. Si une route dans la Loc-RIB est exclue d'une Adj-RIB-Out particulière, La route annoncée précédemment dans cette Adj-RIB-Out DOIT être retirée du service au moyen d'un message UPDATE (voir le paragraphe 9.2).

Des techniques d'agrégation de routes et de réduction d'informations (voir le paragraphe 9.2.2.1) peuvent facultativement être appliquées.

Toute politique locale qui résulte en l'ajout de routes à une Adj-RIB-Out sans aussi être ajoutées au tableau d'acheminement du locuteur BGP locale sort du domaine d'application du présent document.

Quand la mise à jour de la Adj-RIBs-Out et du tableau d'acheminement est achevée, le locuteur BGP local effectue le processus Mise-à-jour-Envoi *(Update-Send)* du paragraphe 9.2.

9.1.4 Chemins en chevauchement

Un locuteur BGP peut transmettre des routes avec des informations d'accessibilité de couche réseau (NLRI) qui chevauchent celle d'un autre locuteur BGP. Le chevauchement des NLRI se produit quand un ensemble de destinations est identifié dans plusieurs routes qui ne correspondent pas. Parce que BGP code les NLRI en utilisant des préfixes IP, le chevauchement va toujours présenter des relations de sous ensembles. Une route qui décrit un plus petit ensemble de destinations (un plus long préfixe) est dite être plus spécifique qu'une route décrivant un plus grand ensemble de destinations (un plus court préfixe) ; de même, une route décrivant une plus grand ensemble de destinations est dite être moins spécifique qu'une route décrivant un plus petit ensemble de destinations.

La relation de préséance décompose effectivement les routes moins spécifiques en deux parties :
- un ensemble de destinations décrites seulement par la route moins spécifique, et
- un ensemble de destinations décrites par le chevauchement de routes moins spécifique et plus spécifiques.

L'ensemble de destinations décrites par le chevauchement représente une portion de la route moins spécifique qui est faisable, mais n'est pas utilisée actuellement. Si une route plus spécifique est ensuite retirée, l'ensemble de destinations décrites par le chevauchement sera encore accessible en utilisant une route moins spécifique.

Si un locuteur BGP reçoit des routes qui se chevauchent, le processus de décision DOIT considérer à la fois les routes sur la base de la politique d'acceptation configurée. Si une route moins spécifique et une route plus spécifique sont toutes deux acceptées, le processus de décision DOIT alors installer dans la Loc-RIB soit les deux routes, plus spécifique et moins spécifique, soit agréger les deux routes et installer dans la Loc-RIB la route agrégée, pourvu que les deux routes aient la même valeur de l'attribut NEXT_HOP.

Si un locuteur BGP choisit d'agréger, il DEVRAIT alors soit inclure tous les AS utilisés pour former l'agrégat dans un AS_SET, soit ajouter l'attribut ATOMIC_AGGREGATE à la route. Cet attribut est principalement informatif. Avec l'élimination des protocoles d'acheminement IP qui ne prennent pas en charge l'acheminement sans classes, et l'élimination des mises en œuvre de routeur et d'hôtes qui ne prennent pas en charge l'acheminement sans classes, il n'est plus nécessaire de désagréger. Les routes NE DEVRAIENT PAS être désagrégées. En particulier, une route qui porte l'attribut ATOMIC_AGGREGATE NE DOIT PAS être désagrégée. C'est-à-dire que les NLRI de cette route ne peuvent pas être plus spécifiques. La transmission sur une telle route ne garantit pas que les paquets IP vont bien ne traverser que les AS figurant sur la liste de l'attribut AS_PATH de la route.

9.2 Processus Update-Send

Le processus Update-Send *(Mise-à-jour-Envoi)* est chargé d'annoncer les messages UPDATE à tous les homologues. Par exemple, il distribue les routes choisies par le processus de décision aux autres locuteurs BGP, qui peuvent être situés dans le même système autonome ou dans un système autonome voisin.

Quand un locuteur BGP reçoit un message UPDATE d'un homologue interne, le locuteur BGP receveur NE DEVRA PAS redistribuer les informations d'acheminement contenues dans ce message UPDATE aux autres homologues internes (sauf si le locuteur agit comme réflecteur de route BGP [RFC2796]).

Au titre de la phase 3 du processus de choix de route, le locuteur BGP a mis à jour sa Adj-RIBs-Out. Toutes les nouvelles routes installées et toutes les nouvelles routes infaisables pour lesquelles il n'y a pas de route de remplacement DEVRONT être annoncées à ses homologues au moyen d'un message UPDATE.

Un locuteur BGP NE DEVRAIT PAS annoncer une certaine route BGP faisable à partir de sa Adj-RIB-Out si cela va produire un message UPDATE contenant la même route BGP qu'annoncée précédemment.

Toute route marquée comme infaisable dans la Loc-RIB DEVRA être retirée. Les changements des destinations accessibles au sein de son propre système autonome DEVRONT aussi être annoncés dans un message UPDATE.

Si, à cause de limites sur la taille maximum d'un message UPDATE (voir la Section 4) une seule route ne tient pas dans le message, le locuteur BGP NE DOIT PAS annoncer la route à ses homologues et PEUT choisir d'enregistrer une erreur en local.

9.2.1 Contrôle des frais généraux du trafic d'acheminement

Le protocole BGP contraint la quantité de trafic d'acheminement (c'est-à-dire de messages UPDATE) afin de limiter à la fois la bande passante de liaison nécessaire pour annoncer les messages UPDATE et la puissance de traitement nécessaire pour que le processus de décision digère les informations contenues dans les messages UPDATE.

9.2.1.1 Fréquence des annonces de chemin

Le paramètre MinRouteAdvertisementIntervalTimer *(temporisateur d'intervalle minimum entre les annonces d'acheminement)* détermine la quantité de temps minimum qui doit d'écouler entre les annonces et/ou retraits de routes pour une destination particulière par un locuteur BGP à un homologue. Cette procédure de limitation de taux s'applique sur la base de la destination, bien que la valeur de MinRouteAdvertisementIntervalTimer soit établie pour chaque homologue BGP.

Deux messages UPDATE envoyés par un locuteur BGP à un homologue qui annoncent des routes faisables et/ou le retrait de routes infaisables à un ensemble commun de destinations DOIVENT être séparés par au moins MinRouteAdvertisementIntervalTimer. Ceci ne peut être réalisé qu'en tenant un temporisateur séparé pour chaque ensemble commun de destinations. Ceci constituerait des frais généraux non garantis. Toute technique qui assure que l'intervalle entre deux messages UPDATE envoyés d'un locuteur BGP à un homologue qui annoncent des routes faisables et/ou un retrait de routes infaisables à un ensemble commun de destinations devra être au moins de MinRouteAdvertisementIntervalTimer, et devra aussi assurer qu'une limite supérieure constante de l'intervalle est acceptable.

Comme une convergence rapide est nécessaire au sein d'un système autonome, soit (a) le MinRouteAdvertisementIntervalTimer utilisé pour les homologues internes DEVRAIT être plus court que le MinRouteAdvertisementIntervalTimer utilisé pour les homologues externes, soit (b) la procédure décrite dans ce paragraphe NE DEVRAIT PAS s'appliquer aux routes envoyées aux homologues internes.

Cette procédure ne limite pas le taux de choix de route, mais seulement le taux d'annonces de routes. Si de nouvelles routes sont choisies plusieurs fois pendant l'attente de l'expiration de MinRouteAdvertisementIntervalTimer, la dernière route choisie DEVRA être annoncée à la fin de MinRouteAdvertisementIntervalTimer.

9.2.1.2 Fréquence des générations de chemin

Le paramètre MinASOriginationIntervalTimer détermine la quantité minimum de temps qui doit s'écouler entre des annonces successives de messages UPDATE qui rapportent des changements au sein des propres systèmes autonomes du locuteur BGP qui annonce.

9.2.2 Organisation efficace des informations d'acheminement

Ayant choisi les informations d'acheminement qu'il veut annoncer, un locuteur BGP peut se donner plusieurs méthodes pour organiser ces informations d'une manière efficace.

9.2.2.1 Réduction des informations

La réduction des informations peut impliquer une réduction de la granularité du contrôle de politique - après que les informations sont réduites, les mêmes politiques vont s'appliquer à toutes les destinations et chemins dans la classe d'équivalence.

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

a) Informations d'accessibilité de couche réseau (NLRI) : les adresses IP de destination peuvent être représentées comme des préfixes d'adresse IP. Dans les cas où il y a une correspondance entre la structure d'adresse et les systèmes sous le contrôle d'un administrateur de système autonome, il va être possible de réduire la taille des NLRI portées dans les messages UPDATE.

b) AS_PATH : les systèmes autonomes peuvent être représentés comme des séquences d'AS ordonnées ou des ensembles non ordonnés d'AS. Les AS_SET sont utilisés dans l'algorithme d'agrégation de routes décrit au paragraphe 9.2.2.2. Ils réduisent la taille des informations de AS_PATH en faisant la liste de chaque numéro d'AS une seule fois, sans considération du nombre de fois qu'il peut être apparu dans les multiples AS_PATH qui ont été agrégés.

Un AS_SET implique que les destinations mentionnées dans les NLRI peuvent être atteintes par des chemins qui traversent au moins certains des systèmes autonomes constituants. Les AS_SET fournissent des informations suffisantes pour éviter des boucles d'informations d'acheminement ; cependant, leur utilisation peut élaguer des chemins faisables potentiels parce que de tels chemins ne figurent plus individuellement sous la forme des AS_SEQUENCE. En pratique, cela ne devrait pas poser de problème parce qu'une fois qu'un paquet IP arrive à la bordure d'un groupe de systèmes autonomes, le locuteur BGP va probablement avoir des systèmes autonomes plus détaillés et peut distinguer les chemins individuels des destinations.

9.2.2.2 Agrégation des informations d'acheminement

L'agrégation est le processus de combinaison des caractéristiques de plusieurs routes différentes d'une façon telle qu'une seule route puisse être annoncée. L'agrégation peut survenir au titre du processus de décision pour réduire la quantité d'informations d'acheminement qui vont être placées dans la Adj-RIBs-Out.

L'agrégation réduit la quantité d'informations qu'un locuteur BGP doit mémoriser et échanger avec les autres locuteurs BGP. Les routes peuvent être agrégées en appliquant la procédure suivante, séparément, aux attributs de chemin de même type et aux informations d'accessibilité de couche réseau.

Les routes qui ont des attributs MULTI_EXIT_DISC différents NE DEVRONT PAS être agrégées.

Si la route agrégée a un AS_SET comme premier élément de son attribut AS_PATH, le routeur qui a généré la route NE DEVRAIT alors PAS annoncer l'attribut MULTI_EXIT_DISC avec cette route.

Les attributs de chemin qui ont des codes de type différents ne peuvent pas être agrégés. Les attributs de chemin de même code de type peuvent être agrégés, selon les règles suivantes :

NEXT_HOP : quand on agrège des routes qui ont des attributs NEXT_HOP différents, l'attribut NEXT_HOP de la route agrégée DEVRA identifier une interface sur le locuteur BGP qui effectue l'agrégation.

attribut ORIGIN : si au moins une route parmi les routes qui sont agrégées a un attribut ORIGIN avec la valeur INCOMPLETE, la route agrégée DOIT alors avoir l'attribut ORIGIN avec la valeur INCOMPLETE. Autrement, si au moins une route parmi les routes qui sont agrégées a ORIGIN avec la valeur EGP, la route agrégée DOIT alors avoir l'attribut ORIGIN avec la valeur EGP. Dans tous les autre cas, la valeur de l'attribut ORIGIN de la route agrégée est IGP.

attribut AS_PATH : si les routes à agréger ont des attributs AS_PATH identiques, la route agrégée a alors le même attribut AS_PATH que chaque route individuelle.

Pour les besoins de l'agrégation des attributs AS_PATH, on modélise chaque AS au sein de l'attribut AS_PATH comme un doublet <type, valeur>, où "type" identifie un type de segment de chemin auquel l'AS appartient (par exemple, AS_SEQUENCE, AS_SET) et "valeur" identifie le numéro d'AS. Si les routes à agréger ont des attributs AS_PATH différents, l'attribut AS_PATH agrégé DEVRA alors satisfaire à toutes les conditions suivantes :

- tous les doublets de type AS_SEQUENCE dans le AS_PATH agrégé DEVRONT apparaître dans tous les AS_PATH dans l'ensemble initial de routes à agréger.

- tous les doublets de type AS_SET dans le AS_PATH agrégé DEVRONT apparaître dans au moins un des AS_PATH de l'ensemble initial (ils peuvent apparaître comme des types AS_SET ou AS_SEQUENCE).

- pour tout doublet X de type AS_SEQUENCE dans le AS_PATH agrégé, qui précède le doublet Y dans le AS_PATH agrégé, X précède Y dans chaque AS_PATH dans l'ensemble initial, qui contient Y, sans considération du type de Y.

- Aucun doublet de type AS_SET avec la même valeur NE DEVRA apparaître plus d'une fois dans le AS_PATH agrégé.

- Plusieurs doublets de type AS_SEQUENCE avec la même valeur ne peuvent apparaître dans le AS_PATH agrégé que quand ils sont adjacents à un autre doublet de mêmes type et valeur.

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

- Déterminer la plus longue séquence de doublets en tête (comme défini ci-dessus) commune à tous les attributs AS_PATH des routes à agréger. Faire de cette séquence la séquence de tête de l'attribut AS_PATH agrégé.

- Régler le type du reste des doublets provenant des attributs AS_PATH des routes à agréger à AS_SET, et l'ajouter à l'attribut AS_PATH agrégé.

- Si le AS_PATH agrégé a plus d'un doublet avec la même valeur (sans considération du type du doublet) les éliminer tous sauf un en supprimant les doublets du type AS_SET de l'attribut AS_PATH agrégé.

- Pour chaque paire de doublets adjacents dans le AS_PATH agrégé, si les deux doublets ont le même type, les fusionner, pour autant que cela ne cause pas la génération d'un segment de plus de 255 octets.

L'Appendice F.6 présente un autre algorithme qui satisfait aux conditions et permet des configurations de politique plus complexes.

ATOMIC_AGGREGATE : si au moins une des routes à agréger a l'attribut de chemin ATOMIC_AGGREGATE, la route agrégée DEVRA avoir aussi cet attribut.

AGGREGATOR : aucun attribut AGGREGATOR provenant des routes à agréger NE DOIT être inclus dans la route agrégée. Le locuteur BGP qui effectue l'agrégation de routes PEUT attacher un nouvel attribut AGGREGATOR (voir le paragraphe 5.1.7).

9.3 Critères de choix de chemin

Généralement, les règles supplémentaires pour comparer les routes sur plusieurs solutions de remplacement sortent du domaine d'application du présent document. Il y a deux exceptions :

- Si l'AS local apparaît sur le chemin d'AS de la nouvelle route considérée, alors cette nouvelle route ne peut pas être vue comme meilleure que toute autre (pourvu que le locuteur soit configuré à accepter de telles routes). Si une telle route était utilisée, il pourrait en résulter un acheminement en boucle.

- Afin de réaliser une opération répartie réussie, seules les routes ayant une probabilité de stabilité peuvent être choisies. Donc, un AS DEVRAIT éviter d'utiliser des routes non stables, et il NE DEVRAIT PAS faire de changements rapides et spontanés de son choix de route. Quantifier les termes "non stable" et "rapide" (dans la phrase précédente) demandera de l'expérience, mais le principe est clair. Les routes qui sont instables peuvent être "pénalisées" (par exemple, en utilisant les procédures décrites dans la [RFC2439]).

9.4 Générer des chemins BGP

Un locuteur BGP peut générer des routes BGP en injectant des informations d'acheminement acquises par d'autres moyens (par exemple, via un IGP) dans BGP. Un locuteur BGP qui génère des routes BGP alloue le degré de préférence (par exemple, selon la configuration locale) à ces routes en les passant à travers le processus de décision (voir le paragraphe 9.1). Ces routes PEUVENT aussi être distribuées aux autres locuteurs BGP au sein de l'AS local au titre du

processus de mise à jour (paragraphe 9.2). La décision de distribuer des routes non acquises par BGP au sein d'un AS via BGP dépend de l'environnement au sein de l'AS (par exemple, type d'IGP) et DEVRAIT être contrôlé via la configuration.

10. Temporisateurs BGP

BGP emploie cinq temporisateurs : ConnectRetryTimer (Section 8), HoldTimer (paragraphe 4.2), KeepaliveTimer (Section 8), MinASOriginationIntervalTimer (paragraphe 9.2.1.2), et MinRouteAdvertisementIntervalTimer (paragraphe 9.2.1.1).

Deux temporisateurs facultatifs, DelayOpenTimer, IdleHoldTimer, PEUVENT être pris en charge par BGP (Section 8). La Section 8 décrit leur utilisation. Le fonctionnement de ces temporisateurs facultatifs sort du domaine d'application du présent document.

ConnectRetryTime est un attribut obligatoire de FSM qui mémorise la valeur initiale de ConnectRetryTimer. La valeur par défaut suggérée pour ConnectRetryTime est 120 secondes.

HoldTime est un attribut obligatoire de FSM qui mémorise la valeur initiale de HoldTimer. La valeur par défaut suggérée pour HoldTime est 90 secondes.

Durant certaines portions de l'automate à états (voir la Section 8) HoldTimer est réglé à une grande valeur. La valeur par défaut suggérée est 4 minutes.

KeepaliveTime est un attribut obligatoire de FSM qui mémorise la valeur initiale de KeepaliveTimer. La valeur par défaut suggérée pour KeepaliveTime est 1/3 de HoldTime.

La valeur par défaut suggérée pour MinASOriginationIntervalTimer est 15 secondes.

La valeur par défaut suggérée pour MinRouteAdvertisementIntervalTimer sur les connexions EBGP est 30 secondes.

La valeur par défaut suggérée pour MinRouteAdvertisementIntervalTimer sur les connexions IBGP est 5 secondes.

Une mise en œuvre de BGP DOIT permettre que HoldTimer soit configurable homologue par homologue, et PEUT permettre que les autres temporisateurs soient configurables.

Pour minimiser la probabilité que la distribution des messages BGP par un certain locuteur BGP fasse des pointes, une gigue DEVRAIT être appliquée aux temporisateurs associés à MinASOriginationIntervalTimer, KeepaliveTimer, MinRouteAdvertisementIntervalTimer, et ConnectRetryTimer. Un locuteur BGP PEUT 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 n'a pas besoin d'être configurée homologue par homologue.

La quantité de gigue par défaut suggérée DEVRA être déterminée en multipliant la valeur de base du temporisateur approprié par un facteur aléatoire, uniformément distribué dans la gamme de 0,75 à 1,0. Une nouvelle valeur aléatoire DEVRAIT être prise chaque fois que le temporisateur est lancé. La gamme des valeurs aléatoires de la gigue PEUT être configurable.

11. Considérations pour la sécurité

Une mise en œuvre de BGP DOIT prendre en charge le mécanisme d'authentification spécifié dans la [RFC2385]. L'authentification fournir par ce mécanisme pourrait être faite homologue par homologue.

BGP utilise TCP pour le transport fiable de son trafic entre routeurs homologues. Pour fournir l'intégrité en mode connexion et l'authentification de l'origine des données sur une base point à point, BGP spécifie l'utilisation du mécanisme défini dans la RFC 2385. Ces services sont destinés à détecter et rejeter les attaques d'écoute active contre les connexions TCP inter routeurs. En l'absence de l'utilisation de mécanismes qui ont un effet sur ces services de sécurité, des attaquants peuvent perturber ces connexions TCP et/ou se faire passer pour le routeur homologue légitime. Parce que le mécanisme défini dans la RFC ne fournit pas l'authentification de l'entité homologue, ces connexions peuvent être soumises à certaines formes d'attaques en répétition qui ne vont pas être détectées à la couche TCP. De telles attaques peuvent résulter en la livraison (à partir de TCP) de messages BGP "cassés" ou "falsifiés".

Le mécanisme défini dans la RFC 2385 augmente la somme de contrôle TCP normale d'un code d'authentification de message (MAC) de 16 octets qui est calculé sur les mêmes données que la somme de contrôle TCP. Ce MAC se fonde sur une fonction de hachage unidirectionnelle (MD5) et l'utilisation d'une clé secrète. La clé est partagée entre les routeurs homologues et est utilisée pour générer les valeurs de MAC qui ne sont pas directement calculées par un attaquant qui n'a pas accès à la clé. Une mise en œuvre conforme doit prendre en charge ce mécanisme, et doit permettre à un administrateur de réseau de l'activer homologue par homologue.

La RFC 2385 ne spécifie pas de moyens de gérer (par exemple, générer, distribuer, et remplacer) les clés utilisées pour calculer le MAC. La [RFC3562] (un document d'information) fournit des lignes directrices dans ce domaine, et fournit les raisons de la prise en charge de ces directives. Elle note qu'une clé distincte devrait être utilisée pour la communication avec chaque homologue protégé. Si la même clé est utilisée pour plusieurs homologues, les services de sécurité offerts peuvent être dégradés, par exemple, du fait du risque accru de compromission d'un routeur qui affecte de façon négative les autres routeurs.

Les clés utilisées pour le calcul de MAC devraient être changées périodiquement, pour minimiser l'impact d'une clé compromise ou d'une attaque réussie de cryptanalyse. La RFC 3562 suggère une période cryptographique (l'intervalle durant lequel une clé est employée) de, au plus, 90 jours. Des changements de clé plus fréquents réduisent la probabilité que des attaques en répétition (comme décrit plus haut) soient faisables. Cependant, en l'absence d'un mécanisme standard pour effectuer de tels changements de façon coordonnée entre homologues, on ne peut pas supposer que les mises en œuvre de BGP-4 qui se conforment à la présente RFC vont prendre en charge des changements de clé fréquents.

Évidemment, chaque clé devrait aussi être choisie pour être difficile à deviner pour un attaquant. Les techniques spécifiées dans la RFC 1750 pour la génération de nombres aléatoires fournissent des directives pour la génération de valeurs qui pourraient être utilisées comme clés. La RFC 2385 invite les mises en œuvre à prendre en charge des clés "composées d'une chaîne de caractères ASCII imprimables de 80 octets ou moins". La RFC 3562 suggère que les clés utilisées dans ce contexte soient de 12 à 24 octets de bits aléatoires (ou pseudo aléatoires). Ceci est parfaitement cohérent avec les suggestions d'algorithmes de MAC analogiques, qui emploient normalement des clés dans la gamme de 16 à 20 octets. Pour fournir suffisamment de bits aléatoires à l'extrémité basse de cette gamme, la RFC 3562 observe aussi qu'une chaîne de texte ACSII normale serait proche de la limite supérieure de longueur de clé spécifiée dans la RFC 2385.

L'analyse des vulnérabilités de BGP est discutée dans la [RFC4272].

12. Considérations relatives à l'IANA

Tous les messages BGP contiennent un type de message de huit bits, pour lequel l'IANA a créé et tient un registre intitulé "Types de messages BGP". Le présent document définit les types de message suivants :

Nom	Valeur	Définition
OPEN	1	voir le paragraphe 4.2
UPDATE	2	voir le paragraphe 4.3
NOTIFICATION	3	voir le paragraphe 4.5
KEEPALIVE	4	voir le paragraphe 4.4

Les allocations futures devront être faites en utilisant le processus d'action de normalisation défini dans la [RFC2434], ou le processus d'allocation précoce de l'IANA défini dans la [RFC4020]. Les allocations consistent en un nom et la valeur.

Les messages UPDATE BGP peuvent porter un ou plusieurs attributs Path, où chaque attribut contient un code de type d'attribut de huit bits. L'IANA tient déjà un tel registre, intitulé "Attributs Path de BGP". Le présent document définit les code de type d'attributs Path suivants :

Nom	Valeur	Définition
ORIGIN	1	voir le paragraphe 5.1.1
AS_PATH	2	voir le paragraphe 5.1.2
NEXT_HOP	3	voir le paragraphe 5.1.3
MULTI_EXIT_DISC	4	voir le paragraphe 5.1.4
LOCAL_PREF	5	voir le paragraphe 5.1.5
ATOMIC_AGGREGATE	6	voir le paragraphe 5.1.6
AGGREGATOR	7	voir le paragraphe 5.1.7

Les allocations futures devront être faites en utilisant le processus d'action de normalisation défini dans la [RFC2434], ou le processus d'allocation précoce de l'IANA défini dans la [RFC4020]. Les allocations consistent en un nom et la valeur.

Le message NOTIFICATION de BGP porte un code d'erreur de 8 bits, pour lequel l'IANA a créé et tient un registre intitulé "Codes d'erreur BGP". Le présent document définit les codes d'erreur suivants :

Nom	Valeur	Définition
Erreur d'en-tête de message	1	paragraphe 6.1
Erreur de message OPEN	2	paragraphe 6.2
Erreur de message UPDATE	3	paragraphe 6.3
Expiration du temporisateur de garde	4	paragraphe 6.5
Erreur de l'automate à états finis	5	paragraphe 6.6
Cessation	6	paragraphe 6.7

Les allocations futures devront être faites en utilisant le processus d'action de normalisation défini dans la [RFC2434], ou le processus d'allocation précoce de l'IANA défini dans la [RFC4020]. Les allocations consistent en un nom et la valeur.

Le message NOTIFICATION BGP porte un sous code d'erreur de huit bits, où chaque sous code doit être défini au sein du contexte d'un code d'erreur particulier, et donc doit être unique au sein de ce seul contexte.

L'IANA a créé et tient un ensemble de registres, "Sous codes d'erreur", avec un registre séparé pour chaque code d'erreur BGP. Les futures allocations devront être faites en utilisant le processus d'action de normalisation défini dans la [RFC2434], ou le processus d'allocation précoce de l'IANA défini dans la [RFC4020]. Les allocations consistent en un nom et la valeur.

Le présent document définit les sous codes d'erreur d'en-tête de message suivants :

Nom	Valeur	Définition
Connexion non synchronisé	1	voir le paragraphe 6.1
Mauvaise longueur de message	2	voir le paragraphe 6.1
Mauvais type de message	3	voir le paragraphe 6.1

Le présent document définit les sous codes d'erreur de message OPEN suivants :

Nom	Valeur	Définition
Numéro de version non pris en charge	1	voir le paragraphe 6.2
Mauvais AS homologue	2	voir le paragraphe 6.2
Mauvais identifiant BGP	3	voir le paragraphe 6.2
Paramètre facultatif non pris en charge	4	voir le paragraphe 6.2
[Déconseillé]	5	voir l'Appendice A
Temps de garde non acceptable	6	voir le paragraphe 6.2

Le présent document définit les sous codes d'erreur de message UPDATE suivants :

Nom	Valeur	Définition
Liste d'attributs mal formée	1	voir le paragraphe 6.3
Attribut bien formé non reconnu	2	voir le paragraphe 6.3
Attribut bien formé manquant	3	voir le paragraphe 6.3
Erreur de fanions d'attribut	4	voir le paragraphe 6.3
Erreur de longueur d'attribut	5	voir le paragraphe 6.3
Attribut ORIGIN invalide	6	voir le paragraphe 6.3
[Déconseillé]	7	voir l'Appendice A
Attribut NEXT_HOP invalide	8	voir le paragraphe 6.3
Erreur d'attribut facultatif	9	voir le paragraphe 6.3
Champ réseau invalide	10	voir le paragraphe 6.3
AS_PATH mal formé	11	voir le paragraphe 6.3

13. Références normatives

[IS10747] Norme internationale ISO/CEI 10747, "Systèmes de traitement de l'information - Échange de télécommunications et d'informations entre systèmes - Protocole pour l'échange d'informations d'acheminement inter domaine entre des systèmes intermédiaires pour la prise en charge de la transmission de PDU ISO", 1993.

[RFC0791] J. Postel, éd., "Protocole Internet - Spécification du protocole du programme Internet", STD 5, septembre 1981.

[RFC0793] J. Postel (éd.), "Protocole de commande de transmission – Spécification du protocole du programme Internet DARPA", (STD 7), septembre 1981.

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

[RFC2385] A. Heffernan, "Protection des sessions de BGP via l'option de signature MD5 de TCP", août 1998. *(P.S.) (Remplacée par* RFC5925)

[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)*

14. Références pour information

[RFC0904] D. Mills, "Spécification formelle du protocole de passerelle extérieure", avril 1984. *(Historique)*

[RFC1092] J. Rekhter, "EGP et l'acheminement fondé sur la politique dans le nouveau cœur de réseau NSFNET", février 1989.

[RFC1093] H. Braun, "Architecture d'acheminement NSFNET", février 1989.

[RFC1105] K. Lougheed et Y Rekhter, "Protocole de routeur frontière BGP)", juin 1989. *(Historique)*

[RFC1163] K. Lougheed, Y. Rekhter, "Protocole de routeur frontière", juin 1990. *(Historique)*

[RFC1267] K. Lougheed et Y Rekhter, "Protocole de routeur frontière 3 (BGP 3)", octobre 1991. *(Historique)*

[RFC1771] Y. Rekhter, T. Li , "Protocole de routeur frontière v. 4 (BGP-4)", mars 1995. *(Obsolète, voir* RFC4271) *(D.S.)*

[RFC1772] Y. Rekhter, P. Gross, "Application du protocole de routeur frontière dans l'Internet", mars 1995. *(D.S.)*

[RFC1518] Y. Rekhter et T. Li, "Architecture pour l'allocation d'adresses IP avec CIDR", septembre 1993. *(Historique)*

[RFC1519] V. Fuller, T. Li, J. Yu et K. Varadhan, "Acheminement inter domaine sans classe (CIDR : stratégie d'allocation et d'agrégation d'adresses", septembre 1993. *(D.S., obsolète, voir la RFC4632)*

[RFC1930] J. Hawkinson, T. Bates, "Lignes directrices pour la création, sélection, et l'enregistrement d'un système autonome (AS)", mars 1996. (BCP0006)

[RFC1997] R. Chandra, P. Traina, T. Li, "Attribut Community de BGP", août 1996. *(P.S.)*

[RFC2439] C. Villamizar, R. Chandra, R. Govindan, "Élimination des oscillations de chemin dans BGP", novembre 1998. *(P.S.)*

[RFC2474] K. Nichols, S. Blake, F. Baker et D. Black, "Définition du champ Services différenciés (DS) dans les en-têtes IPv4 et IPv6", décembre 1998. (*MàJ par* RFC3168, RFC3260) *(P.S.)*

[RFC2796] T. Bates, R. Chandra, E. Chen, "Réflexion de chemin BGP - une alternative à IBGP à maillage complet", avril 2000. (*Obsolète, voir* RFC4456) *(P.S.)*

[RFC2858] T. Bates et autres, "Extensions multiprotocoles pour BGP-4", juin 2000. (*Obsolète, voir* RFC4760) *(P.S.)*

[RFC2918] E. Chen, "Capacité de rafraîchissement de chemin pour BGP-4", septembre 2000. *(P.S.)*

[RFC3065] P. Traina, D. McPherson, J. Scudder, "Confédérations de systèmes autonomes pour BGP", février 2001. (*Obsolète, voir* RFC5065) *(P.S.)*

[RFC3392] R. Chandra et J. Scudder, "Annonces de capacités avec BGP-4", novembre 2002.

[RFC3562] M. Leech, "Considérations sur la gestion de clés pour l'option de signature MD5 dans TCP", juillet 2003. *(Information)*

[RFC4020] K. Kompella et A. Zinin, "Allocation précoce par l'IANA de codets pour des RFC en cours de normalisation", BCP 100, février 2005.

[RFC4272] S. Murphy, "Analyse des faiblesses de la sécurité de BGP", janvier 2006. *(Information)*

Appendice A Comparaison avec la RFC 1771

Il y a de nombreux changements rédactionnels par rapport à la [RFC1771] (trop pour en faire la liste).

Voici la liste des changements techniques :
Changements pour refléter l'usage de dispositifs tels que TCP MD5 [RFC2385], les réflecteurs de route BGP [RFC2796], les confédérations BGP [RFC3065], et le rafraîchissement de route BGP [RFC2918].
Précisions sur l'utilisation de l'identifiant BGP dans l'attribut AGGREGATOR.
Procédures pour imposer une limite supérieure au nombre de préfixes qu'un locuteur BGP va accepter d'un homologue.
Capacité d'un locuteur BGP d'inclure plus d'une instance de son propre AS dans l'attribut AS_PATH pour les besoins de l'ingénierie de trafic inter AS.
Précisions des divers types de NEXT_HOP.
Clarification de l'usage de l'attribut ATOMIC_AGGREGATE.
Relations entre le prochain bond immédiat, et le prochain bond comme spécifié dans l'attribut de chemin NEXT_HOP.
Clarification des procédures de départage.
Clarification de la fréquence des annonces de route.
Le type 1 de paramètre facultatif (Informations d'authentification) a été déconseillé.
Le sous code 7 d'erreur de message UPDATE (boucle d'acheminement dans l'AS) a été déconseillé.
Le sous code d'erreur de message OPEN (Échec d'authentification) a été déconseillé.
L'utilisation du champ Marqueur pour l'authentification a été déconseillée.
Les mises en œuvre DOIVENT prendre en charge TCP MD5 [RFC2385] pour l'authentification.
Précisions sur l'automate à états finis de BGP.

Appendice B Comparaison avec la RFC 1267

Tous les changements mentionnés dans l'Appendice A, plus les suivants :
BGP-4 est capable de fonctionner dans un environnement où un ensemble de destinations accessibles peut être exprimé via un seul préfixe IP. Le concept de classes de réseau, ou de sous réseautage, est étranger à BGP-4. Pour s'accommoder de ces capacités, BGP-4 change la sémantique et le codage associés à l'attribut AS_PATH. Un nouveau texte a été ajouté pour définir la sémantique associée aux préfixes IP. Ces capacités permettent à BGP-4 de prendre en charge le schéma de sous réseautage proposé [RFC1518], [RFC1519].
Pour simplifier la configuration, cette version introduit un nouvel attribut, LOCAL_PREF, qui facilite les procédures de choix de route.
L'attribut INTER_AS_METRIC a été renommé MULTI_EXIT_DISC.
Un nouvel attribut, ATOMIC_AGGREGATE, a été introduit pour assurer que certains agrégats ne sont pas désagrégés. Un autre nouvel attribut, AGGREGATOR, peut être ajouté pour agréger les routes pour annoncer quel AS et quel locuteur BGP au sein de cet AS ont causé l'agrégation.
Pour assurer que les temporisateurs de garde sont symétriques, ils sont maintenant négociés connexion par connexion. Les temporisateurs de garde de zéro sont maintenant acceptés.

Appendice C Comparaison avec la RFC 1163

Tous les changements mentionnés dans les Appendices A et B, plus les suivants :

Pour détecter et récupérer de la collision de connexions BGP, un nouveau champ (identifiant BGP) a été ajouté au message OPEN. Un nouveau texte (paragraphe 6.8) a été ajouté pour spécifier la procédure pour détecter et récupérer de collision.

Le nouveau document ne restreint plus le routeur qui est passé dans l'attribut de chemin NEXT_HOP à faire partie du même système autonome que le locuteur BGP.

Le nouveau document optimise et simplifie l'échange d'informations sur les précédemment accessibles.

Appendice D Comparaison avec la RFC 1105

Tous les changements mentionnés dans les Appendices A, B, et C, plus les suivants :
Des changements mineurs à l'automate à états finis de la [RFC1105] ont été nécessaires pour s'accommoder de l'interface d'utilisateur TCP fournie par BSD version 4.3.
La notion de relations Haut/Bas/Horizontal présentées dans la RFC 1105 a été retirée du protocole.
Les changements du format de message par rapport à la RFC 1105 sont comme suit :
1. Le champ Hold Time a été retiré de l'en-tête BGP et ajouté au message OPEN.
2. Le champ Version a été retiré de l'en-tête BGP et ajouté au message OPEN.

3. Le champ Type de liaison a été supprimé du message OPEN.
4. Le message OPEN CONFIRM a été éliminé et remplacé par une confirmation implicite, fournie par le message KEEPALIVE.
5. Le format du message UPDATE a été significativement changé. De nouveaux champs ont été ajoutés au message UPDATE pour prendre en charge plusieurs attributs de chemin.
6. Le champ Marqueur a été étendu et son rôle élargi pour prendre en charge l'authentification.

Noter qu'assez souvent, BGP, comme spécifié dans la RFC 1105, est appelé BGP-1 ; BGP, comme spécifié dans la RFC1163, est appelé BGP-2 ; BGP, comme spécifié dans la RFC 1267 est appelé BGP-3, et BGP, spécifié dans le présent document est appelé BGP-4.

Appendice E Options TCP qui peuvent être utilisées avec BGP

Si une interface d'utilisateur de système local TCP supporte la fonction TCP PUSH, chaque message BGP DEVRAIT alors être transmis avec le fanion PUSH établi. Établir le fanion PUSH force la transmission rapide des messages BGP au receveur.

Si une interface d'utilisateur de système local TCP prend en charge le réglage du champ DSCP [RFC2474] pour les connexions TCP, la connexion TCP utilisée par BGP DEVRAIT alors être ouverte avec les bits 0 à 2 du champ DSCP réglés à 110 (en binaire).

Une mise en œuvre DOIT prendre en charge l'option TCP MD5 [RFC2385].

Appendice F Recommandations de mise en œuvre

Cette section présente des recommandations de mise en œuvre.

F.1 Plusieurs réseaux par message

Le protocole BGP permet que plusieurs préfixes d'adresse avec les mêmes attributs de chemin soient spécifiés dans un message. L'utilisation de cette capacité est fortement recommandée. Avec un préfixe d'adresse par message, il y a une augmentation substantielle des frais généraux chez le receveur. Non seulement les frais généraux du système augmentent à cause de la réception de plusieurs messages, mais les frais généraux de l'examen du tableau d'acheminement pour les mises à jour chez les homologues BGP et les autres protocoles d'acheminement (et l'envoi des messages associés) sont aussi supportés plusieurs fois.

Une méthode de construction des messages qui contiennent plusieurs préfixes d'adresse par ensemble d'attributs de chemin à partir d'un tableau d'acheminement qui n'est pas organisé par ensemble d'attributs de chemin est de construire de nombreux messages lorsque le tableau d'acheminement est examiné. Lorsque chaque préfixe d'adresse est traité, un message pour l'ensemble associé d'attributs de chemin est alloué, 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 ajouté. Si le message n'a pas la place pour contenir le nouveau préfixe d'adresse qui est transmis, un nouveau message est alloué, et le nouveau préfixe d'adresse est inséré dans le nouveau message. Quand le tableau d'acheminement a été entièrement examiné, tous les messages alloués sont envoyés et leurs ressources sont libérées. Un maximum de compression est réalisé quand toutes les destinations couvertes par les préfixes d'adresse partagent un ensemble commun d'attributs de chemin, rendant possible l'envoi de beaucoup de préfixes d'adresse dans un message de 4096 octets.

Lors de l'appariement avec une mise en œuvre BGP qui ne compresse pas plusieurs préfixes d'adresse dans un message, il peut être nécessaire de prendre des mesures pour réduire les frais généraux résultant du flux de données reçues lors de l'acquisition d'un homologue ou quand un changement significatif de la topologie du réseau se produit. Une méthode pour le faire est de limiter le taux de mises à jour. Cela va éliminer la redondance d'examen du tableau d'acheminement jusqu'à fournir des mises à jour "éclair" pour les homologues BGP et les autres protocoles d'acheminement. L'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 "flash" qui ne soit pas 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.

F.2. Réduction de l'oscillation de route

Pour éviter d'excessives oscillations de route, un locuteur BGP qui a besoin de retirer une destination et envoyer une mise à jour sur une route plus spécifique ou moins spécifique devrait les combiner dans le même message UPDATE.

F.3 Ordre des attributs de chemin

Les mises en œuvre qui combinent les messages de mise à jour (comme décrit au paragraphe 6.1) peuvent préférer voir tous les attributs de chemin présentés dans un ordre connu. Cela permet d'identifier rapidement la règle des attributs provenant des différents messages de mise à jour qui sont sémantiquement identiques. Pour faciliter cela, ordonner les attributs de chemin selon le code de type est une optimisation utile. Elle est entièrement facultative.

F.4 Tri de AS_SET

Une autre optimisation utile qui peut être faite pour simplifier cette situation est de trier les numéros d'AS qui se trouvent dans un AS_SET. Cette optimisation est entièrement facultative.

F.5 Contrôle de la négociation de version

Comme BGP-4 est capable de porter des routes agrégées qui ne peuvent pas être représentées de façon appropriée dans BGP-3, une mise en œuvre qui prend en charge BGP-4 et une autre version de BGP devrait fournir la capacité de ne parler BGP-4 que homologue par homologue.

F.6 Agrégation AS_PATH complexe

Une mise en œuvre qui choisit de fournir un algorithme d'agrégation de chemins retenant des quantités significatives de systèmes autonomes peut souhaiter utiliser la procédure suivante :

Pour les besoins de l'agrégation des attributs AS_PATH de deux routes, on modélise chaque AS comme un doublet <type, valeur>, où "type" identifie le type du segment de chemin auquel appartient l'AS (par exemple, AS_SEQUENCE, AS_SET), et "valeur" est le numéro d'AS. Deux AS sont dits être le même si leurs doublets correspondants <type, valeur> sont les mêmes.

L'algorithme pour agréger deux attributs AS_PATH fonctionne comme suit :

(a) Identifier les mêmes AS (comme défini ci-dessus) au sein de chaque attribut AS_PATH qui sont dans le même ordre relatif au sein des deux attributs AS_PATH. Deux AS, X et Y, sont dits être dans le même ordre si :
- soit X précède Y dans les deux attributs AS_PATH,
- soit Y précède X dans les deux attributs AS_PATH.

(b) L'attribut AS_PATH agrégé consiste en AS identifiés dans (a) exactement le même ordre que celui dans lequel ils apparaissent dans les attributs AS_PATH à agréger. Si deux AS consécutifs identifiés en (a) ne se suivent pas immédiatement dans les deux attributs AS_PATH à agréger, alors les AS intermédiaires (AS entre des deux AS consécutifs qui sont les mêmes) dans les deux attributs sont combinés en un segment de chemin AS_SET qui consiste en les AS intermédiaires provenant des deux attributs AS_PATH. Ce segment est alors placé entre les deux AS consécutifs identifiés dans (a) de l'attribut agrégé. Si deux AS consécutifs identifiés dans (a) se suivent immédiatement dans un attribut, mais pas dans l'autre, les AS intermédiaires du dernier sont alors combinés dans un segment de chemin AS_SET. Ce segment est ensuite placé entre les deux AS consécutifs identifiés dans (a) de l'attribut agrégé.

(c) Pour chaque paire de doublets adjacents dans le AS_PATH agrégé, si les deux doublets ont le même type, les fusionner si cela ne cause pas la génération d'un segment de plus de 255 octets.

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

Adresse des éditeurs

Yakov Rekhter
Juniper Networks
mél : yakov@juniper.net

Tony Li
mél: tony.li@tony.li

Susan Hares
NextHop Technologies, Inc.
825 Victors Way
Ann Arbor, MI 48108
téléphone : (734)222-1610
mél : skh@nexthop.com

Déclaration complète de droits de reproduction

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