Groupe de travail Réseau
Request for Comments : 2439
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

C. Villamizar, ANS
R. Chandra, Cisco
R. Govindan, ISI
novembre 1998

Atténuation de fluctuations de chemin dans BGP

Résumé
On décrit une utilisation du protocole d'acheminement BGP qui est capable de réduire le trafic d'acheminement passé aux homologues d'acheminement et donc la charge qui pèse sur ces homologues sans affecter de façon contraire le temps de convergence de chemin pour les chemins relativement stables. Cette technique a été mise en œuvre dans des produits commerciaux qui prennent en charge BGP. La technique est aussi applicable à IDRP.
Les objectifs globaux sont :
de fournir un mécanisme capable de réduire la charge de traitement des routeurs causée par l'instabilité
d'empêcher ce faisant les oscillations durables de chemin
de le faire sans sacrifier le temps de convergence de chemin pour ceux qui se comportent généralement bien.
Cela doit se réaliser en gardant en vue les autres objectifs de BGP :
grouper les changements en un petit nombre de mises à jour
préserver la cohérence de l'acheminement
minimiser l'espace supplémentaire et la surcharge de calcul.
Un taux excessif de mise à jour des annonces d'accessibilité d'un sous-ensemble des préfixes de l'Internet a eu largement cours dans l'Internet. Cette observation a été faite au début des années 1990 par de nombreuses personnes impliquées dans le fonctionnement de l'Internet et elle reste valable. Ces mises à jour excessives ne sont pas nécessairement périodiques, de sorte que l'oscillation de chemin serait un terme trompeur. Le terme informel utilisé pour décrire cet effet est "fluctuation de chemin". Les techniques décrites ici sont maintenant largement déployées et on s'y réfère couramment sous le nom de "atténuation de fluctuation de chemin".

Table des matières

1. Vue d'ensemble....2
2. Méthodes de limitation des annonces de chemin....2
2.1 Recommandations existantes de temporisateur fixe....2
2.2 Propriétés souhaitables des algorithmes d'atténuation....3
2.3 Choix de conception....3
3. Limiter les annonces de chemin avec des temporisateurs fixes....4
4. Suppression sensible à la stabilité des annonces de chemin....4
4.1 Ensembles de paramètres de configuration simples ou multiples....5
4.2 Paramètres de configuration....5
4.3 Lignes directrices pour le réglage des paramètres....6
4.4 Structures de données au moment du démarrage....9
4.5 Traitement des paramètres de configuration....12
4.6 Construction de la matrice des indices de réutilisation....12
4.7 Exemple de configuration....13
4.8 Traitement de l'activité du protocole d'acheminement....14
5. Expérience de mise en œuvre....18
Remerciements....19
Références....20
Adresse des auteurs....20
Déclaration complète de droits de reproduction....20

1. Vue d'ensemble

Pour conserver l'adaptabilité de l'acheminement dans l'Internet, il est nécessaire de réduire le nombre de changements des états d'acheminement propagés par BGP afin de limiter les exigences de traitement. Les principaux contributeurs à la charge de traitement résultant des mises à jour BGP sont les processus de prise de décision de BGP et l'ajout et le retrait des entrées de transmission.

Considérons l'exemple suivant. Une mise en œuvre largement déployée de BGP peut tendre à échouer à cause d'un fort volume de mises à jour d'acheminement. Par exemple, elle peut être incapable de conserver ses sessions BGP ou IGP si elles sont suffisamment chargées. L'échec d'un routeur peut de plus contribuer à la charge sur les autres routeurs. Cette charge supplémentaire peut causer des défaillances dans d'autres instances de la même mise en œuvre ou dans d'autres mises en œuvre avec une faiblesse similaire. Dans le pire des cas, il peut en résulter une oscillation stable. De tels pires cas ont déjà été observés en pratique.

Une mise en œuvre de BGP doit être prête à un gros volume de trafic d'acheminement. Une mise en œuvre de BGP ne peut pas compter qu'un envoyeur se protège suffisamment contre les instabilités de chemin. Les lignes directrices qui sont données ici sont conçues pour empêcher les oscillations continuelles, mais n'éliminent pas le besoin d'une mise en œuvre robuste et efficace. Les mécanismes décrits ici permettent que l'instabilité de l'acheminement soit contenue au routeur frontière de l'AS qui touche l'instabilité.

Même lorsque les mises en œuvre de BGP sont très robustes, les performances du processus d'acheminement sont limitées. Limiter la propagation d'un changement non nécessaire devient alors le problème de conserver un temps de convergence de changement de chemin raisonnable lorsque augmente une topologie d'acheminement.

2. Méthodes de limitation des annonces de chemin

Deux méthodes de contrôle de la fréquence des annonces de chemin sont décrites ici. La première implique des temporisateurs fixes. La technique du temporisateur fixe n'a pas de redondance d'espace par chemin mais présente l'inconvénient de ralentir la convergence du chemin pour le cas normal où un chemin n'a pas un historique d'instabilité. La seconde méthode surmonte cette limitation aux dépens de la conservation d'une certaine redondance spatiale supplémentaire. La redondance supplémentaire inclut une petite quantité d'état par chemin et une très petite redondance de traitement.

Il est possible et souhaitable de combiner les deux techniques. En pratique, des temporisateurs fixes ont été réglés à de très courts intervalles et se sont révélés utiles pour grouper des chemins en un petit nombre de mises à jour lorsque les chemins arrivent dans des mises à jour séparées. Le protocole BGP se réfère à cela sous le nom de paquetage d'informations d'accessibilité de couche réseau (NLRI, *Network Layer Reachability Information*) [RFC1771].

Rares sont les temporisateurs fixes réglés aux dizaines de minutes ou heures qui seraient nécessaires pour atténuer réellement les fluctuations de chemin. Le faire produirait l'effet indésirable de limiter sévèrement la convergence de chemin.

2.1 Recommandations existantes de temporisateur fixe

BGP-3 ne fait pas de recommandation spécifique dans ce domaine [RFC1268]. Le court paragraphe intitulé "Fréquence de choix de chemin" recommande simplement que quelque chose soit fait et fait une vague déclaration concernant certaines propriétés qui sont désirables ou indésirables.

BGP4 conserve le paragraphe "Fréquence des annonces de chemin" et ajoute un paragraphe "Fréquence de génération de chemin". BGP-4 décrit une méthode de limitation des annonces de chemin impliquant un temporisateur fixe (configurable) MinRouteAdvertisementInterval et un temporisateur fixe MinASOriginationInterval [RFC1771]. Les valeurs de temporisateur recommandées sont de 30 secondes pour MinRouteAdvertisementInterval et de 15 secondes pour MinASOriginationInterval.

2.2 Propriétés souhaitables des algorithmes d'atténuation

Avant de décrire les algorithmes d'atténuation, les objectifs doivent être clairement définis. Certaines propriétés clés sont examinées pour préciser les raisons de la conception.

L'objectif global est de réduire la charge de mise à jour de chemins sans limiter le temps de convergence pour les chemins

qui se comportent bien. Pour accomplir cela, des critères doivent être définis pour les chemins qui se comportent bien et pour ceux qui se comportent mal. Un algorithme doit être défini qui permette d'identifier les chemins qui se comportent mal. Idéalement, cette mesure serait une prédiction de la future stabilité d'un chemin.

Tout retard dans la propagation des chemins qui se comportent bien devrait être minimal. Un certain retard est tolérable pour prendre en charge un meilleur groupage des mises à jour. Le retard des chemins au mauvais comportement devrait, si possible, être proportionnel à une mesure de l'instabilité future attendue du chemin. Le retard de propagation d'un chemin instable devrait être cause de suppression du chemin instable jusqu'à ce qu'il y ait un certain degré de confiance que le chemin s'est stabilisé.

Si un grand nombre de changements de chemin sont reçus dans des mises à jour séparées sur une très courte période et si ces mises à jour ont un potentiel de combinaison en une seule mise à jour, elles devraient alors être groupées aussi efficacement que possible avant de les propager plus avant. Un petit retard dans la propagation de chemins au bon comportement est tolérable et est nécessaire pour permettre un meilleur groupage des mises à jour.

Lorsque des chemins sont instables, l'utilisation et l'annonce des chemins devraient être supprimées plutôt que de supprimer leur retrait. Lorsque un chemin pour une destination est stable, et qu'un autre chemin pour la même destination est un peu instable, si possible, le chemin instable devrait être supprimé plus agressivement que si il n'y avait pas de chemin de remplacement.

La cohérence de l'acheminement au sein d'un système autonome (AS, *Autonomous System*) est très importante. Seul un retard très minimal du BGP interne (IBGP) devrait être fait. La cohérence de l'acheminement à travers les frontières d'AS est aussi très importante. Il n'est pas du tout souhaitable d'annoncer un chemin qui est différent de celui qui est utilisé, sauf pour un temps très court. Il est plus souhaitable de supprimer l'acceptation d'un chemin (et donc, l'utilisation de ce chemin dans l'IGP) plutôt que de supprimer seulement la redistribution.

Il est clair qu'il n'est pas possible de prédire avec précision la stabilité future d'un chemin. L'histoire récente de la stabilité est généralement considérée comme une bonne base pour estimer la probabilité de la stabilité future. Les critères utilisés pour distinguer les chemins qui se comportent bien de ceux qui se comportent mal se fondent donc sur l'histoire récente de la stabilité du chemin. Il n'y a pas d'expression quantitative simple de la stabilité récente, de sorte qu'un classement doit être défini. Certaines caractéristiques souhaitables de ce classement pourraient être que plus loin dans le passé cette instabilité s'est produite, moins son effet sur le classement se fera sentir et la mesure de l'instabilité serait cumulative plutôt que reflétant seulement les événements les plus récents.

Les algorithmes devraient se comporter de telle façon que pour les chemins qui ont une histoire de stabilité mais subissent quelques transitions, ces transitions devraient être faites rapidement. Si les transitions continuent, l'annonce du chemin devrait être supprimée. Il devrait y avoir une mémoire de l'instabilité antérieure. Le degré de prise en compte de l'instabilité antérieure devrait être graduellement réduit tant que le chemin reste annoncé et stable.

2.3 Choix de conception

Après que les chemins ont été acceptés, leur réannonce sera rapidement supprimée pour améliorer le groupage des mises à jour. Il peut y avoir une lente suppression de l'acceptation d'un chemin externe. Le temps pendant lequel un chemin va être supprimé se fonde sur un classement qui est supposé corrélé à la probabilité d'une future instabilité du chemin. Les chemins avec des valeurs de classement élevées seront supprimés. Un algorithme de diminution exponentielle a été choisi comme base de la réduction du classement au fil du temps. Ces choix devraient être vus comme des suggestions pour la mise en œuvre.

Une fonction de décroissance exponentielle a la propriété que l'instabilité antérieure peut être remémorée pendant assez longtemps. Le taux auquel la valeur d'instabilité diminue se ralentit à mesure que le temps passe. La diminution exponentielle a la propriété suivante :

f(f(classement, t1), t2) = f(classement, t1+t2)

Cette propriété permet que la diminution sur une longue période soit calculée dans une seule opération sans considération de la valeur actuelle (classement). Comme optimisation des performances, la diminution peut être appliquée dans des incréments de temps fixes. Connaissant la demie vie d'une diminution désirée, la diminution pour un seul incrément de temps peut être calculée à l'avance. La diminution pour plusieurs incréments de temps est exprimée par .

f(classement, n*t0) = f(classement, t0)**n = K**n

Les valeurs de K ** n peuvent être pré calculées pour un nombre raisonnable de "n" et mémorisées dans une matrice. La valeur de "K" est toujours inférieure à un. La taille de la matrice peut être bornée car la valeur approche rapidement de zéro. Cela rend la diminution facile à calculer en utilisant une vérification de limite de la matrice, une recherche dans la matrice et une seule multiplication sans considération de la quantité de temps écoulée.

3. Limiter les annonces de chemin avec des temporisateurs fixes

Cette méthode de limitation des annonces de chemin implique l'utilisation de temporisateurs fixes appliqués au processus d'envoi des chemins. Son principal objet est d'améliorer le groupage des chemins dans les messages de mise à jour de BGP. Le retard de l'annonce d'un chemin stable devrait être borné et minimal. Le retard à l'annonce d'un chemin inaccessible n'a pas besoin d'être zéro, mais devrait aussi être borné et devrait probablement avoir une borne distincte réglée à une valeur inférieure ou égale à la borne pour une annonce d'accessibilité.

Le protocole BGP définit l'utilisation d'une base de données d'informations d'acheminement (RIB, *Routing Information Base*). Les chemins qui ont besoin d'être réannoncés peuvent être marqués dans la RIB ou dans un ensemble externe de structures qui font référence à la RIB.

Périodiquement, un sous ensemble des chemins marqués peut être purgé. Ceci est très direct et réalise les objectifs. Le calcul pour une mise en œuvre très simple peut être d'ordre N au carré. Pour éviter d'effectuer l'élévation de N au carré, une certaine forme de structure de données est nécessaire pour grouper les chemins avec des attributs communs.

Une mise en œuvre devrait grouper efficacement les mises à jour, fournir un délai minimum de ré annonce, fournir une borne au délai maximum de ré annonce qui va être subi seulement comme résultat de l'algorithme utilisé pour fournir un délai minimum, et doit être efficace en calcul en présence d'un très grand nombre de candidats à la ré annonce.

4. Suppression sensible à la stabilité des annonces de chemin

Cette méthode de limitation des annonces de chemin utilise une mesure de la stabilité de chemin appliquée sur la base du chemin. Cette technique est appliquée lors de la réception de mises à jour provenant seulement d'homologues externes (EBGP). Appliquer cette technique aux chemins appris de IBGP ou pour les annonces aux homologues IBGP ou EBGP après le choix de chemin peut résulter en boucles d'acheminement.

Un classement fondé sur une mesure de l'instabilité est tenu sur la base du chemin. Ce classement est utilisé dans la décision de supprimer l'utilisation d'un chemin. Les chemins qui ont un rang élevé dans le classement sont supprimés. Chaque fois qu'un chemin est retiré, le classement est incrémenté. Alors que le chemin ne change pas, la valeur du classement subit une diminution exponentielle avec des taux de diminution séparés selon que le chemin est stable et accessible ou a été moins stable et inaccessible. Le taux de diminution peut être plus lent lorsque le chemin est inaccessible, ou le classement de stabilité pourrait rester fixe (ne pas diminuer du tout) alors que le chemin reste inaccessible. Diminuer les chemins inaccessibles au même taux, à un taux plus lent, ou pas du tout, est un choix de la mise en œuvre. Diminuer à un taux plus lent est recommandé.

Une mise en œuvre très efficace est suggérée dans les paragraphes suivants. La mise en œuvre exige seulement le calcul pour les chemins contenus dans une mise à jour, lorsque une mise à jour est reçue ou retirée (par opposition à l'approche simpliste de diminution périodique de chaque chemin). La mise en œuvre suggérée implique seulement un petit nombre d'opérations simples, et peut être mise en œuvre en utilisant des entiers normalisés.

Le comportement des chemins instables est très prévisible. Les chemins très fluctuants vont souvent être annoncés et retirés à des intervalles de temps réguliers correspondant aux temporisateurs d'un protocole particulier (IGP ou le protocole extérieur utilisé là où le problème existe). Des circuits marginaux ou un encombrement léger peuvent résulter en un schéma à long terme de brefs retraits occasionnels de chemin ou en brève connexité occasionnelle.

4.1 Ensembles de paramètres de configuration simples ou multiples

Le comportement de l'algorithme est modifié par un certain nombre de paramètres configurables. Il est possible de configurer des ensembles de paramètres distincts conçus pour traiter des fluctuations de chemins sévères à court terme et des fluctuations chroniques plus douces (un schéma d'abandons occasionnels sur une longue période). Les premières vont exiger une diminution rapide et un seuil bas (permettant qu'un petit nombre de fluctuations consécutives cause la suppression d'un chemin, mais lui permettant d'être réutilisé après une période relativement courte de stabilité). Les dernières vont exiger une diminutions très lente et un seuil plus élevé et pourraient être appropriées pour des chemins pour lesquels il y a un chemin de remplacement de bande passante similaire.

Il peut aussi être désirable de configurer, pour les chemins qui ont des solutions de remplacement en gros équivalentes, des seuils différents de ceux des chemins où les solutions de remplacements ont une bande passante plus faible ou tendent à être encombrés. Cela peut se résoudre en associant un ensemble différent de paramètres à différentes gammes de valeurs de préférence. Le choix des paramètres pourrait être fondé sur LOCAL_PREF de BGP.

Le choix des paramètres pourrait aussi se fonder sur la connaissance de chemins de remplacement. Un chemin serait pris en compte si, pour tout ensemble de paramètres applicables, un chemin de remplacement avec la valeur de préférence spécifiée existe et si le classement associé à l'ensemble de paramètres n'indique pas le besoin de supprimer le chemin. Une suppression moins agressive serait appliquée au cas où il n'existe pas du tout de chemin de remplacement. Dans le plus simple des cas, une suppression plus agressive serait appliquée si il existait un chemin de remplacement. Seule la plus haute valeur de préférence (la préférée) a besoin d'être spécifiée, car les gammes peuvent se chevaucher.

Il peut aussi être désirable de configurer un ensemble différent de seuils pour les chemins qui s'appuient sur des services commutés et peuvent parfois déconnecter pour réduire les charges de connexion. On peut s'attendre à ce que de tels chemins changent d'état peut-être un peu plus souvent, mais ils devraient être supprimés si des changements continuels d'état indiquent une instabilité.

Bien que ce ne soit pas essentiel, il peut être désirable d'être capable de configurer plusieurs ensembles de paramètres de configuration par chemin. Il peut aussi être désirable d'être capable de configurer des ensembles de paramètres qui ne correspondent qu'à un ensemble de chemins (identifiés par chemin d'AS, routeur homologue, destinations spécifiques ou autres moyens). L'expérience peut dicter quelle souplesse est nécessaire et comment régler au mieux les paramètres. Il appartient à la mise en œuvre de choisir d'allouer des ensembles de paramètres d'atténuation différents pour les différents chemins, et d'allouer plusieurs classements par chemin.

Le choix des paramètres peut aussi se fonder sur la longueur du préfixe. La raison en est que les préfixes plus longs tendent à atteindre moins de systèmes d'extrémité et sont moins importants et ces préfixes moins importants peuvent être atténués plus agressivement. Cette technique est très largement utilisée. Les petits sites ou ceux qui ont une allocation d'adresses dense ou qui sont multi rattachements sont souvent accessibles par de longs préfixes qui ne sont pas facilement agrégés. Ces sites tendent à contester le choix de la longueur de préfixe pour la sélection de paramètres. Les avocats de cette technique soulignent qu'elle encourage une meilleure agrégation.

4.2 Paramètres de configuration

Au moment de la configuration, un certain nombre de paramètres peuvent être spécifiés par l'usager. Les paramètres de configuration sont exprimés en unités significatives pour l'usager. Cela diffère des paramètres utilisés au démarrage qui sont en unités convenables pour le calcul. Les paramètres du démarrage sont déduits des paramètres de configuration. Les paramètres de configuration suggérés sont cités ci-dessous.

seuil de coupure *(cutoff threshold)* (cut)
Cette valeur est exprimée par un nombre de retraits de chemins. C'est la valeur au dessus de laquelle une annonce de chemin sera supprimée.

seuil de réutilisation *(reuse threshold)* (reuse)
Cette valeur est exprimée par un nombre de retraits de chemins. C'est la valeur en dessous de laquelle un chemin supprimé sera maintenant réutilisé.

durée maximum de suppression *(maximum hold down time)* (T-hold)
Cette valeur est la durée maximum pendant laquelle un chemin peut être supprimé quelle que soit l'instabilité qu'elle a subie avant cette période de stabilité.

diminution de demie vie lorsque accessible *(decay half life while reachable)* (decay-ok)

Cette valeur est la durée en minutes ou secondes durant laquelle le classement de stabilité accumulé sera réduit de moitié si le chemin est considéré comme accessible (qu'il soit supprimé ou non).

diminution de demie vie lorsque inaccessible *(decay half life while unreachable)* (decay-ng)
Cette valeur est la durée en minutes ou secondes durant laquelle le classement de stabilité accumulé sera réduit de moitié si le chemin est considéré comme inaccessible. Si elle n'est pas spécifiée ou si elle est réglée à zéro, aucune diminution ne va se produire tant qu'un chemin reste inaccessible.

limite de mémoire de diminution *(decay memory limit)* (Tmax-ok ou Tmax-ng)
C'est la durée maximum pendant laquelle tout souvenir de l'instabilité antérieure sera conservé étant donné que l'état du chemin reste inchangé, qu'il soit accessible ou inaccessible. Ce paramètre est généralement utilisé pour déterminer les tailles de matrice.

Il peut y avoir plusieurs ensembles des paramètres ci-dessus comme décrit au paragraphe 4.1. Les paramètres de configuration cités ci-dessus seront appliqués à l'ensemble du système. Cela inclut la granularité du temps de tous les calculs, et les paramètres utilisés pour contrôler la réévaluation des chemins qui ont été supprimés antérieurement.

granularité du temps (delta-t)
C'est la granularité du temps en secondes utilisée pour effectuer tous les calculs de diminution.

granularité du temps de liste de réutilisation (delta-reuse)
C'est l'intervalle de temps entre les évaluations des listes de réutilisation. Chaque liste de réutilisation correspond à un incrément de temps supplémentaire.

souvenir de la liste de réutilisation *(reuse list memory)* (reuse-list-max)
C'est la valeur de l'heure correspondant à la dernière liste de réutilisation. Cela peut être la valeur maximum de T-hold pour tous les ensembles de paramètres ou cela peut être configuré.

nombre de listes de réutilisation (reuse-list-size)
C'est le nombre de listes de réutilisation. Il peut être déterminé à partir de reuse-list-max ou réglé explicitement.

Une optimisation recommandée est décrite au paragraphe 4.8.6 qui implique une matrice appelée "matrice d'indice de réutilisation". Une matrice d'indice de réutilisation est nécessaire pour chaque taux de diminution utilisé. La matrice d'indice de réutilisation est utilisée pour estimer dans quelle liste de réutilisation placer un chemin lorsque il est supprimé. Un placement approprié évite qu'il soit besoin d'évaluer périodiquement la diminution pour déterminer si un chemin peut être réutilisé ou quand la mémorisation peut être récupérée. L'utilisation de la matrice d'indice de réutilisation évite d'avoir besoin de calculer un logarithme pour déterminer le placement. Un paramètre supplémentaire pour l'ensemble du système peut être introduit.

taille de matrice d'indice de réutilisation (reuse-index-array-size)
C'est la taille des matrices d'indice de réutilisation. Cette taille détermine la précision avec laquelle les chemins supprimés peuvent être placés au sein de l'ensemble des listes de réutilisation lorsque ils sont supprimés pendant longtemps.

4.3 Lignes directrices pour le réglage des paramètres

La diminution de demie vie devrait être réglée à une durée considérablement plus longue que la période que la fluctuation de chemin est destinée à prendre en compte. Par exemple, si la diminution est réglée à dix minutes et si un chemin est retiré et ré annoncé exactement toutes les dix minutes, le chemin va continuer à fluctuer si l'arrêt a été réglé à une valeur de 2 ou plus.

Le classement de stabilité est lui même un total de temps cumulé diminué. On doit s'en souvenir lors du réglage du temps de diminution, de valeurs d'arrêt, et des valeurs de réutilisation. Le classement est augmenté chaque fois que des chemins passent d'accessible à inaccessible. Le classement est diminué à un taux proportionnel à sa valeur actuelle. Augmenter le taux de fluctuation de chemin augmente donc plus souvent le classement et atteint un certain seuil dans un temps plus court. Lorsque est préparée la réponse à un taux constant de fluctuation de chemin, cela ressemble à des dents de scie avec un bord montant abrupt et un bord descendant à pic. Comme la valeur absolue de la diminution est proportionnelle au classement, à un taux de fluctuation constante continu la ligne de base des dents de scie va tendre à arrêter de croître et va converger si elle n'est pas coupée par une valeur plafond.

Si elle est coupée par une valeur plafond, la ligne de base des dents de scie va simplement atteindre plus vite le plafond à un taux de fluctuation de chemin plus élevé. Par exemple, si il fluctue à quatre fois le taux de diminution, la progression

suivante se produit. Lorsque le chemin devient inaccessible pour la première fois, la valeur devient 1. Lorsque se produit la fluctuation suivante, on ajoute un à la valeur précédente, qui a été diminuée de la racine quatrième de 2 (la quantité de diminution qui se produirait en 1/4 de la demie vie si la diminution est exponentielle). La séquence est 1, 1,84, 2,55, 3,14, 3,64, 4,06, 4,42, 4,71, 4,96, 5,17, ..., convergeant à environ 6,285. Si un chemin fluctue à quatre fois le taux de diminution, il va atteindre 3 en 4 cycles, 4 en 6 cycles, 5 en 10 cycles, et va converger à environ 6,3. À deux fois le temps de diminution, il va atteindre 3 en 7 cycles, et converger à une valeur de moins de 3,5.

À la Figure 1 le classement de stabilité pour le chemin fluctue à un taux constant. L'axe du temps est étiqueté en multiples de la demie vie de diminution. Les plots représentent la fluctuation de chemin avec une période de 1/2, 1/3, 1/4, et 1/8 fois la demie vie de diminution. Un plafond de 4,5 été établi, qui peut être vu comme affectant trois des plots, limitant effectivement le temps que cela prend pour réannoncer le chemin sans considération de l'histoire antérieure. Avec des seuils de coupure et de réutilisation de 1,5 et 0,75, les chemins seront supprimés après avoir été déclarés inaccessibles 2 à 3 fois et seront réutilisés après approximativement deux périodes de demie vie de diminution de périodes de stabilité.

Cette fonction peut être exprimée de façon formelle. L'accessibilité d'un chemin peut être représentée par une variable "R" avec des valeurs possibles de 0 et 1 représentant l'accessible et l'inaccessible. Un temps R discret peut seulement avoir une valeur. Le classement est augmenté de 1 à chaque transition de R = 1 à R = 0 et borné par une valeur plafond. La diminution du classement peut alors être exprimée sur un ensemble de temps discrets comme suit.

classement(t) = K * classement(t - delta-t)
K = K1 pour R = 0 K = K2 pour R = 1

Les quatre plots sont présentés verticalement. Du fait des limitations d'espace, seul un ensemble limité de points est représenté le long de l'axe des temps. La valeur du classement est donnée. À côté de chaque valeur, il y a un graphique à très faible résolution constitué de points ASCII. C'est juste destiné à donner une sensation grossière de la croissance et décroissance des valeurs. Les graphiques ne sont pas affichés sur un ensemble d'axes se chevauchant parce que les ondes en dents de scie se recoupent assez fréquemment. Avec la très faible résolution de ces plots, la montée et la descente de la ligne de base est évidente, mais la nature des dents de scie n'est observable que dans la valeur imprimée.

À partir de la valeur maximum de temps de garde (T-hold), on peut déterminer un ratio de la valeur de réutilisation par rapport à un plafond. On peut choisir une valeur d'entier pour le plafond telle que le dépassement ne soit pas un problème et que toutes les autres valeurs puissent être adaptées en conséquence. Si les deux coupures sont spécifiées ou si plusieurs jeux de paramètres sont utilisés, le plafond le plus élevé sera utilisé.

temps	**classement en fonction du temps** (en minutes)			
0,00	0,000 .	0,000 .	0,000 .	0,000 .
0,08	0,000 .	0,000 .	0,000 .	0,000 .
0,16	0,000 .	0,000 .	0,000 .	0,973 .
0,24	0,000 .	0,000 .	0,000 .	0,920 .
0,32	0,000 .	0,000 .	0,946 .	1,817 .
0,40	0,000 .	0,953 .	0,895 .	2,698 .
0,48	0,000 .	0,901 .	0,847 .	2,552 .
0,56	0,953 .	0,853 .	1,754 .	3,367 .
0,64	0,901 .	0,807 .	1,659 .	4,172 .
0,72	0,853 .	1,722 .	1,570 .	3,947 .
0,80	0,807 .	1,629 .	2,444 .	4,317 .
0,88	0,763 .	1,542 .	2,312 .	4,469 .
0,96	0,722 .	1,458 .	2,188 .	4,228 .
1,04	1,649 .	2,346 .	3,036 .	4,347 .
1,12	1,560 .	2,219 .	2,872 .	4,112 .
1,20	1,476 .	2,099 .	2,717 .	4,257 .
1,28	1,396 .	1,986 .	3,543 .	4,377 .
1,36	1,321 .	2,858 .	3,352 .	4,141 .
1,44	1,250 .	2,704 .	3,171 .	4,287 .
1,52	2,162 .	2,558 .	3,979 .	4,407 .
1,60	2,045 .	2,420 .	3,765 .	4,170 .
1,68	1,935 .	3,276 .	3,562 .	4,317 .
1,76	1,830 .	3,099 .	4,356 .	4,438 .
1,84	1,732 .	2,932 .	4,121 .	4,199 .
1,92	1,638 .	2,774 .	3,899 .	3,972 .
2,00	1,550 .	2,624 .	3,688 .	3,758 .
2,08	1,466 .	2,483 .	3,489 .	3,555 .
2,16	1,387 .	2,349 .	3,301 .	3,363 .

2,24	1,312 .	2,222 .	3,123 .	3,182 .
2,32	1,242 .	2,102 .	2,955 .	3,010 .
2,40	1,175 .	1,989 .	2,795 .	2,848 .
2,48	1,111 .	1,882 .	2,644 .	2,694 .
2,56	1,051 .	1,780 .	2,502 .	2,549 .
2,64	0,995 .	1,684 .	2,367 .	2,411 .
2,72	0,941 .	1,593 .	2,239 .	2,281 .
2,80	0,890 .	1,507 .	2,118 .	2,158 .
2,88	0,842 .	1,426 .	2,004 .	2,042 .
2,96	0,797 .	1,349 .	1,896 .	1,932 .
3,04	0,754 .	1,276 .	1,794 .	1,828 .
3,12	0,713 .	1,207 .	1,697 .	1,729 .
3,20	0,675 .	1,142 .	1,605 .	1,636 .
3,28	0,638 .	1,081 .	1,519 .	1,547 .
3,36	0,604 .	1,022 .	1,437 .	1,464 .
3,44	0,571 .	0,967 .	1,359 .	1,385 .

Figure 1 : Représentation de l'instabilité pour fluctuation à taux constant

temps	**classement en fonction du temps** (en minutes)		
0,00	0,000 .	0,000 .	0,000 .
0,20	0,000 .	0,000 .	0,000 .
0,40	0,000 .	0,000 .	0,000 .
0,60	0,000 .	0,000 .	0,000 .
0,80	0,000 .	0,000 .	0,000 .
1,00	0,999 .	0,999 .	0,999 .
1,20	0,971 .	0,971 .	0,929 .
1,40	0,945 .	0,945 .	0,809 .
1,60	0,919 .	0,865 .	0,704 .
1,80	0,894 .	0,753 .	0,613 .
2,00	1,812 .	1,657 .	1,535 .
2,20	1,762 .	1,612 .	1,428 .
2,40	1,714 .	1,568 .	1,244 .
2,60	1,667 .	1,443 .	1,083 .
2,80	1,622 .	1,256 .	0,942 .
3,00	1,468 .	1,094 .	0,820 .
3,20	2,400 .	2,036 .	1,694 .
3,40	2,335 .	1,981 .	1,475 .
3,60	2,271 .	1,823 .	1,284 .
3,80	2,209 .	1,587 .	1,118 .
4,00	1,999 .	1,381 .	0,973 .
4,20	2,625 .	2,084 .	1,727 .
4,40	2,285 .	1,815 .	1,503 .
4,60	1,990 .	1,580 .	1,309 .
4,80	1,732 .	1,375 .	1,139 .
5,00	1,508 .	1,197 .	0,992 .
5,20	1,313 .	1,042 .	0,864 .
5,40	1,143 .	0,907 .	0,752 .
5,60	0,995 .	0,790 .	0,654 .
5,80	0,866 .	0,688 .	0,570 .
6,00	0,754 .	0,599 .	0,496 .
6,20	0,656 .	0,521 .	0,432 .
6,40	0,571 .	0,454 .	0,376 .
6,60	0,497 .	0,395 .	0,327 .
6,80	0,433 .	0,344 .	0,285 .
7,00	0,377 .	0,299 .	0,248 .
7,20	0,328 .	0,261 .	0,216 .
7,40	0,286 .	0,227 .	0,188 .
7,60	0,249 .	0,197 .	0,164 .
7,80	0,216 .	0,172 .	0,142 .
8,00	0,188 .	0,150 .	0,124 .

Figure 2 : constantes de diminution séparées lors d'inaccessibilité

La Figure 2 montre les effets de la configuration de taux de diminution séparés à utiliser lorsque le chemin est accessible ou inaccessible. Le taux de diminution est 5 fois plus faible lorsque le chemin est inaccessible. Dans les trois cas indiqués, la période de la fluctuation de chemin est égale à la demie vie de diminution mais le chemin est accessible 1/8 du temps dans un, accessible 1/2 du temps dans un, et accessible 7/8 du temps dans l'autre. Dans le dernier cas, le chemin n'est pas supprimé jusqu'au tiers inaccessible (lorsque il est au dessus du seuil supérieur après être redevenu accessible).

Le point principal de la Figure 2 est de montrer l'effet de changer le facteur d'utilisation du signal carré dans la variable "R" pour une fréquence fixée du signal carré. Si les constantes de diminution sont choisies de façon telle que la diminution soit plus lente lorsque R=0 (le chemin est inaccessible) alors le classement remonte plus lentement (plus précisément, la ligne de base de l'onde en dents de scie monte plus lentement) si le chemin est accessible un plus grand pourcentage du temps. L'effet lorsque le chemin redevient accessible de façon persistante peut être très négligeable si les dents de scie sont bordées par une valeur plafond, mais est plus significatif si un taux de fluctuation de chemin lent ou un court intervalle de fluctuation de chemin est tel que les dents de scie n'atteignent pas la valeur plafond. Dans la Figure 2, l'intervalle dans lequel les chemins sont instables est assez court pour que la valeur plafond ne soit pas atteinte, donc les chemins qui sont accessibles pendant un pourcentage supérieur du cycle de fluctuation du chemin sont réutilisés (placés dans la RIB et annoncés aux homologues) plus tôt que les autres après que le chemin est redevenu stable ("R" devient 1, indiquant que l'état annoncé devient accessible et le reste).

Dans les deux figures 1 et 2, les chemins seraient supprimés. Les chemins qui fluctuent à la demie vie de diminution ou moins seraient retirés deux ou trois fois et restent ensuite retirés jusqu'à ce qu'ils soient restés annoncés et stables pendant de l'ordre de 1,5 à 2,5 fois la demie vie de diminution (étant donné le plafond de l'exemple).

L'objet de l'atténuation de la fluctuation de chemin dans BGP est de réduire la charge de traitement chez le routeur immédiat et la charge de traitement sur les routeurs en aval (routeurs BGP homologues et homologues qui vont voir les annonces de chemin faites par le routeur immédiat). Calculer un classement à chaque intervalle de temps discret en utilisant classement (t) = K * classement(t - delta-t) serait très inefficace et contre productif. Ce problème est traité en différant le calcul autant que possible et en faisant un seul simple calcul pour compenser la diminution durant le temps écoulé depuis la dernière mise à jour du classement. L'utilisation de matrices de diminution donne le seul calcul simple. L'utilisation de listes de réutilisation (décrites plus loin) donne le moyen de différer les calculs. Un chemin devient utilisable si il n'y avait pas eu d'autre changement pendant un certain temps et si le chemin est inaccessible. La mémorisation de la structure de données est récupérée si l'état du chemin n'a pas changé pendant un certain temps et si il a été inaccessible. La matrice de réutilisation donne un moyen pour estimer pendant combien de temps un calcul peut être différé si il n'y a pas d'autre changement.

Un plus grosse granularité temporelle fera baisser la mémorisation du tableau. La granularité temporelle devrait être inférieure à un temps minimal raisonnable entre les fluctuations de chemin dans le pire des cas. Il peut être raisonnable de fixer ce paramètre au moment du calcul ou d'établir une valeur par défaut et il est vivement recommandé que l'utilisateur n'y touche pas. Avec une diminution exponentielle, la taille de la matrice peut être grandement réduite en établissant une période de complète stabilité après laquelle le total diminué sera considéré comme zéro plutôt que de garder une petite quantité. Autrement, de très longues diminutions peuvent être mises en œuvre en multipliant plus d'une fois si les bornes de la matrice sont dépassées.

Les listes de réutilisation contiennent les chemins supprimés groupés selon le temps qu'il faudra pour que les chemins soient éligibles à la réutilisation. Chaque liste va être périodiquement avancée d'une position et une liste sera retirée dans les conditions décrites au paragraphe 4.8.7. Tous les chemins supprimés dans la liste retirée seront réévalués et soit utilisés, soit placés dans une autre liste, selon la quantité de temps supplémentaire qui doit s'écouler avant que le chemin puisse être réutilisé. La dernière liste va toujours contenir tous les chemins qui ne seront pas annoncés pendant plus de temps qu'il est approprié pour les têtes de liste restantes. Lorsque la dernière liste avance en tête, certains des chemins ne seront pas prêts à être utilisés et devront être remis dans la file d'attente. L'intervalle de temps pour reconsidérer les chemins supprimés et le nombre de têtes de liste devrait être configurable. Les valeurs par défaut raisonnables pourraient être de 30 secondes et de 64 têtes de liste. Un chemin supprimé pendant longtemps aurait besoin d'être réévalué toutes les 32 minutes.

4.4 Structures de données au moment du démarrage

Une petite quantité fixe de mémorisation par système sera nécessaire. Lorsque plusieurs ensembles de paramètres de configuration sont utilisés, la mémorisation devra se faire par ensemble de paramètres. Une petite quantité de mémorisation par chemin est nécessaire. Un ensemble de têtes de liste est nécessaire. Ces têtes de liste sont utilisées pour arranger les chemins supprimés selon le temps qui reste jusqu'à ce qu'ils puissent être réutilisés.

Une liste de réutilisation distincte peut être tenue pour les chemins inaccessibles pour les besoins de récupération ultérieure

de la mémorisation si ils restent trop longtemps inaccessibles. Cela peut être décrit plus précisément comme une liste de recyclage. L'avantage que cela procurerait est de libérer aussitôt que possible des structures de données devenues disponibles. Autrement, les structures de données peuvent simplement être placées dans une file d'attente et la capacité de mémorisation récupérée lorsque le chemin arrive en tête de la file d'attente et qu'on a besoin d'espace de mémorisation. Cette dernière solution est moins optimale mais simple.

Si plusieurs ensembles de paramètres de configuration par chemin sont permis, il est besoin d'un moyen pour associer plus d'un classement et ensemble de paramètres à chaque chemin. Construire une liste à liens de ces objets semble une mise en œuvre raisonnable, entre autres. De même, on a besoin d'un moyen d'associer un chemin à une liste de réutilisation. Une petite redondance sera exigée pour les pointeurs nécessaires à la mise en œuvre de la structure de données qui sera choisie pour les listes de réutilisation. La mise en œuvre suggérée utilise des listes doublement liées et exige donc deux pointeurs par classement.

Chaque jeu de paramètres de configuration peut faire référence aux matrices de diminution et aux matrices de réutilisation. Ces matrices devraient être partagées par plusieurs jeux de paramètres car leurs exigences de mémorisation ne sont pas négligeables. Il y aura seulement un jeu de têtes de liste de réutilisation pour le routeur entier.

4.4.1 Structures de données pour ensembles de paramètres de configuration

Sur la base des paramètres de configuration décrits au paragraphe précédent, les valeurs suivantes peuvent être calculées comme des entiers normalisés directement des paramètres de configuration correspondants.

- facteur d'échelle de matrice de diminution (decay-array-scale-factor)
- valeur de coupure (cut)
- valeur de réutilisation (reuse)
- plafond de classement (ceiling)

Chaque jeu de paramètres de configuration va faire référence à une ou deux matrices de diminution et à une ou deux matrices de réutilisation. Une seule matrice sera nécessaire si le taux de diminution est le même lorsque le chemin est inaccessible que lorsque il est accessible, ou si le classement de stabilité ne diminue pas lorsque un chemin est inaccessible.

4.4.2 Structures de données par matrice de diminution et matrice d'indice de réutilisation

Les éléments suivants sont aussi calculés à partir des paramètres de configuration mais pas aussi directement. Le calcul est décrit au paragraphe 4.5.

- taux de diminution par tic (decay-delta-t)
- taille de matrice de diminution (decay-array-size)
- matrice de diminution (decay[])
- taille de matrice d'indice de réutilisation (reuse-index-array-size)
- matrice d'indice de réutilisation (reuse-index-array[])

Pour chaque taux de diminution spécifié, une matrice sera utilisée pour mémoriser la valeur d'un paramètre calculé élevé à la puissance de l'indice de chaque élément de matrice. Ceci est destiné à accélérer les calculs. Le taux de diminution par tic est une valeur intermédiaire exprimée par un nombre réel et utilisée pour calculer les valeurs mémorisées dans les matrices de diminution. La taille de la matrice est calculée à partir du paramètre de configuration de limite de mémoire exprimé par une taille de matrice ou comme un temps de garde maximum.

La taille de la matrice de diminution doit être suffisante pour s'accommoder de la mémoire de diminution spécifiée sachant la granularité temporelle, ou suffisante pour contenir le nombre d'éléments de matrice jusqu'à ce que les arrondis d'entier produisent un résultat zéro si cette valeur est plus petite, ou une taille raisonnable imposée par la mise en œuvre pour empêcher les configurations qui utilisent une mémoire excessive. Les mises en œuvre peuvent choisir de rendre la taille de matrice plus courte et de multiplier plus d'une fois lors d'une réduction sur un long intervalle de temps pour réduire la mémorisation.

Les matrices d'indice de réutilisation ont un objet similaire à celui des matrices de diminution. Dans BGP, un chemin est dit être "utilisé" si il est considéré comme le meilleur chemin. Dans ce contexte, si le chemin est "utilisé", il est placé dans la RIB et est éligible pour annonce aux homologues BGP. Si un chemin est retiré (une annonce BGP est faite par un homologue qui indique qu'il n'est plus accessible) il n'est alors plus éligible pour "être utilisé". Lorsque un chemin devient accessible, il se peut qu'il ne puisse pas être "utilisé" immédiatement si le classement indique que s'est récemment produite une instabilité. Après que le chemin est resté stable et que le classement est descendu en dessous du seuil de "réutilisation", le chemin est dit éligible à la "réutilisation" (traité comme vraiment accessible, placé dans la RIB et annoncé aux homologues). La quantité de temps avant qu'un chemin puisse être réutilisé peut être déterminé en utilisant une recherche dans la matrice. La matrice peut être construite selon un taux de diminution. La matrice est indexée en utilisant un entier

normé proportionnel au ratio entre une valeur courante de classement de stabilité et la valeur nécessaire pour que le chemin soit réutilisé.

4.4.3 État par chemin

Les informations doivent être entretenues par un tuple représentant un chemin. Au minimum, les NLRI (préfixe et longueur BGP) doivent être contenues dans le tuple. Différents attributs BGP peuvent être inclus ou exclus selon la situation spécifique. Le chemin d'AS devrait aussi être contenu dans le tuple par défaut. Le tuple peut aussi facultativement contenir d'autres attributs BGP tels que MULTI_EXIT_DISCRIMINATOR (MED).

Le tuple représentant un chemin pour les besoins de l'atténuation de fluctuation de chemin est :

entrée de tuple de NLRI	**par défaut**	**options**
préfixe	exigé	
longueur	exigé	
chemin d'AS	inclus	option d'exclure
dernier AS établi dans le chemin	exclu	option d'inclure
prochain bond	exclu	option d'inclure
MED	exclu	option d'inclure seulement dans les comparaisons

Le chemin d'AS est généralement inclus afin d'identifier l'instabilité vers l'aval qui n'est pas, ou pas suffisamment, atténuée et alterne entre un chemin stable et instable. Dans de rares circonstances, il peut être désirable d'exclure le chemin d'AS pour tous les préfixes ou un de leurs sous-ensembles. Si un chemin d'AS se termine dans un ensemble d'AS, en pratique le chemin est toujours pour un agrégat. Les changements à l'ensemble d'AS de queue devraient être ignorés. Idéalement, la comparaison de chemin d'AS devrait assurer qu'au moins un AS est resté constant dans l'ancien et le nouvel ensemble d'AS, mais ignorer complètement le contenu d'un ensemble d'AS de queue est aussi acceptable.

Inclure les changements de prochain bond et de MED peut aider à supprimer l'utilisation d'un AS qui est instable en interne ou à éviter un prochain bond qui est plus proche d'un chemin IGP instable dans l'AS adjacent. Si un grand nombre de valeurs de MED sont utilisées, l'augmentation de la quantité d'état peut devenir un problème. Pour cette raison MED est désactivé par défaut et n'est activé qu'au titre de la comparaison de tuple, en utilisant une seule entrée d'état, sans considération de la valeur de MED. L'inclusion de MED supprime l'utilisation de l'AS adjacent même si le changement n'a pas besoin d'être propagé plus loin. Utiliser MED n'est une pratique sûre que si il est connu qu'un chemin existe à travers un autre AS ou lorsque il y a suffisamment de sites qui échangent du trafic avec l'AS adjacent pour que les chemins entendus seulement sur un sous ensemble des sites d'échange de trafic soient supprimés.

4.4.4 Structures de données par chemin

Les informations suivantes doivent être tenues par chemin. Un chemin est ici considéré comme un tuple contenant usuellement NLRI, prochain bond, et chemin d'AS comme défini au paragraphe 4.4.3.

Classement de stabilité (figure-of-merit)
Chaque chemin doit avoir un classement de stabilité par ensemble de paramètre applicable.

Dernière mise à jour (time-update)
L'heure exacte de la dernière mise à jour doit être conservée pour permettre une diminution exponentielle du classement cumulé à différer jusqu'à ce que le chemin puisse raisonnablement être considéré éligible à un changement d'état (étant passé de inaccessible à accessible ou en avançant sur les listes de réutilisation).

Pointeur de bloc de configuration
Toute mise en œuvre qui prend en charge des ensembles multiples de paramètres doit fournir un moyen d'identifier rapidement quel ensemble de paramètres correspond au chemin actuellement considéré. Pour les mises en œuvre qui ne prennent en charge que certains ensembles de paramètres où tous les chemins doivent être traités de la même façon, ce pointeur n'est pas nécessaire.

Pointeur de traversée de liste de réutilisation
Si des listes à double liaison sont utilisées pour mettre en œuvre les listes de réutilisation, deux pointeurs seront alors nécessaires, un sur le précédent et un sur le suivant. Généralement, il y a une liste à double liaison, inutilisée lorsque un chemin est supprimé de l'utilisation, qui peut être utilisée pour la traversée de liste de réutilisation, ce qui élimine le besoin de mémorisation d'un pointeur supplémentaire.

4.5 Traitement des paramètres de configuration

À partir des paramètres de configuration, il est possible de pré calculer un certain nombre de valeurs qui seront utilisées de façon répétitive et de les conserver pour accélérer les calculs ultérieurs qui seront exigés fréquemment.
L'ordre de grandeur dépend généralement de la plus forte valeur que peut atteindre la classement, qu'on appelle ici le plafond. La valeur du nombre réel du plafond va normalement être déterminée par l'équation qui suit. Le plafond peut aussi être configuré à une valeur spécifique, qui à son tour dicte T-hold.

plafond = réutilisation * (exp(T-hold/demie-vie-de-diminution) * log(2))

Dans l'équation ci-dessus, réutilisation est le seuil de réutilisation décrit au paragraphe 4.2.

Les méthodes d'arithmétique d'entier normalisé ne sont pas décrites en détail ici. Les méthodes de détermination des valeurs réelles sont données. La traduction en valeurs d'entier normalisé et les détails de l'arithmétique des entiers normalisés sont laissés au choix des mises en œuvre individuelles.

La valeur plafond peut être réglée au plus grand entier qui peut tenir dans la moitié des bits disponibles pour un entier non signé. Cela va permettre aux entiers normalisés d'être multipliés par la valeur de diminution normalisée et ensuite réduits. Les mises en œuvre peuvent préférer utiliser des nombres réels ou peuvent utiliser toute normalisation d'entier réputée appropriée pour leur architecture.

Valeur et seuils de pénalité (comme entiers normalisés proportionnels)
La pénalité de classement pour un retrait de chemin et les valeurs de coupure doivent être adaptées conformément au facteur d'adaptation.

Taux de diminution par tic (decay[1])
La valeur de diminution par incrément de temps comme définie par la granularité du temps doit être déterminée (au moins initialement comme un nombre à virgule flottante). La diminution par tic est un nombre légèrement inférieur à un. C'est la racine $N^{ème}$ de la moitié de un où N est la demie vie divisée par la granularité du temps.

decay[1] = exp ((1 / (decay-half-life/delta-t)) * log (1/2))

Taille de matrice de diminution (decay-array-size)
La taille de matrice de diminution est la mémoire de diminution divisée par la granularité du temps. Si la troncature d'entier ramène la valeur d'un élément de matrice à zéro, la matrice peut être diminuée. Une mise en œuvre devrait aussi imposer une taille maximum raisonnable de matrice ou permettre plus d'une multiplication.

decay-array-size = (Tmax/delta-t)

Matrice de diminution (decay[])
Chaque $i^{ème}$ élément de la matrice de diminution est le retard par tic élevé à la puissance i. Cela pourrait le mieux être réalisé par des multiplications successives à virgule flottante suivies par une normalisation et un arrondi à l'entier ou une troncature. La matrice elle-même a seulement besoin d'être calculée à son début.

decay[i] = decay[1] ** i

4.6 Construction de la matrice des indices de réutilisation

On peut accéder aux listes de réutilisation assez fréquemment si un certain nombre de chemins fluctuent suffisamment pour être supprimés. On suggère une méthode pour accélérer la détermination de la liste de réutilisation à utiliser pour un certain chemin. Cette méthode est introduite au paragraphe 4.2, sa configuration décrite au paragraphe 4.4.2 et les algorithmes décrits aux paragraphes 4.8.6 et 4.8.7. Le présent paragraphe décrit la construction des matrices d'indice de liste de réutilisation.
Un ratio du classement du chemin considéré sur la valeur de coupure est utilisé comme base pour une recherche dans la matrice. Le ratio est normalisé et tronqué à une valeur d'entier et utilisé pour indexer la matrice. L'entrée de matrice est un entier utilisé pour déterminer quelle liste de réutilisation utiliser.

Ratio maximum de matrice d'indice de réutilisation (max-ratio)
C'est le ratio maximum entre la valeur actuelle du classement de stabilité et la valeur de réutilisation cible qui peut être indexée par la matrice de réutilisation. Il peut être limité par le plafond imposé par le temps de garde maximum ou par le temps que couvrent les listes de réutilisation.

max-ratio = min(plafond/reuse, exp((1 / (half-life/reuse-array-time)) * log(2)))

Facteur d'adaptation de matrice de réutilisation (scale-factor)
Comme la matrice de réutilisation est une estimation, le facteur d'adaptation de matrice de réutilisation doit être calculé de façon telle que soit utilisée la taille complète de la matrice de réutilisation.

scale-factor = reuse-index-array-size / (max-ratio - 1)

Matrice d'indice de réutilisation (reuse-index-array[])
Chaque entrée d'indice de réutilisation devrait contenir un indice dans la matrice de liste de réutilisation qui pointe sur une des têtes de liste. Cet indice devrait correspondre à la liste de réutilisation qui sera évaluée juste après qu'un chemin sera éligible à la réutilisation, étant donné le ratio de la valeur actuelle du classement de stabilité sur la valeur de réutilisation de la cible correspondant à l'entrée de matrice de réutilisation.

reuse-index-array[j] = entier((decay-half-life / reuse-time-granularity) * log(1/(reuse * (1 + (j / scale-factor)))) / log(1/2))

Pour déterminer dans quelle file d'attente de réutilisation placer un chemin qui va être supprimé, on utilise la procédure suivante. On divise le classement actuel par la valeur de coupure. On soustrait un. On multiplie par le facteur d'adaptation. C'est l'indice dans la matrice d'indice de réutilisation (reuse-index-array[]). La valeur qu'on est allé chercher dans la matrice d'indice de réutilisation (reuse-index-array[]) est un indice dans la matrice des listes de réutilisation (reuse-array[]). Si cet indice est au-delà de la fin de la matrice, utiliser la dernière file d'attente, autrement, chercher dans la matrice et prendre le numéro de la file d'attente de la matrice à cet indice. C'est assez rapide et vaut bien l'établissement et la mémorisation requis.

4.7 Exemple de configuration

On présente ici un exemple simple dans lequel la redondance d'espace est estimée pour un ensemble de paramètres de configuration. On fait les hypothèses suivantes :
1. il y a un seul jeu de paramètres utilisé pour tous les chemins,
2. le temps de diminution pour les chemins inaccessibles est plus lent que pour les chemins accessibles,
3. les matrices doivent être de pleine taille, plutôt que de permettre que plus d'une multiplication par opération de diminution réduise la taille de matrice.

Cet exemple est utilisé dans les paragraphes suivants. L'utilisation de plusieurs jeux de paramètres complique un peu les exemples. Lorsque plusieurs jeux de paramètres sont permis pour un seul chemin, la portion diminution de l'algorithme est répétée pour chaque jeu de paramètres. Si il est permis que des chemins différents aient des jeux de paramètres différents, les chemins doivent avoir des pointeurs sur les jeux de paramètres pour réduire au minimum le temps de localisation, mais les algorithmes sont inchangés par ailleurs.

Un échantillon de jeu de paramètres de configuration et un échantillon de jeu de paramètres de mise en œuvre sont fournis dans les deux listes suivantes.

1. Paramètres de configuration
   cut = 1,25
   reuse = 0,5
   T-hold = 15 min
   decay-ok = 5 min
   decay-ng = 15 min
   Tmax-ok = 15 min
   Tmax-ng = 30 min

2. Paramètres de mise en œuvre
   delta-t = 1 s
   delta-reuse = 15 s
   reuse-list-size = 256
   reuse-index-array-size = 1,024

En utilisant ces paramètres de configuration et de mise en œuvre et les équations du paragraphe 4.5, la redondance d'espace peut être calculée. Il y a une redondance d'espace fixe qui est indépendante du nombre de chemins. Il y a une exigence d'espace associée à un chemin stable. Une plus grande exigence d'espace est associée à un chemin instable. Les exigences

d'espace pour les paramètres ci-dessus sont données dans les listes ci-dessous.

1. Redondance fixe (en utilisant les paramètres de l'exemple précédent)
   900 * entier – matrice de diminution
   1,800 * entier - matrice de diminution
   120 * pointeur – réutilisation des têtes de liste
   2,048 * entier – matrice d'indice de réutilisation

2. Redondance par chemin stable
   pointeur - contenant une entrée nulle

3. Redondance par chemin instable
   pointeur – sur une structure d'atténuation contenant ce qui suit
   entier - classement + bit pour l'état
   entier – dernière mise à jour
   2 * pointeur – pointeurs de listes de réutilisation (précédent, suivant)

Les matrices de diminution sont dimensionnées selon delta-t et Tmax-ok ou Tmax-ng. Le nombre de têtes de liste de réutilisation est fondé sur delta-reuse et le plus grand de Tmax-ok ou Tmax-ng. Deux matrices d'indice de réutilisation ont une taille qui est un paramètre configuré.

La Figure 3 montre le comportement de l'algorithme avec les paramètres ci-dessus. Quatre cas sont donnés dans cet exemple. Dans tous les quatre, il y une période de douze minutes d'oscillations de chemin. Deux périodes d'oscillations sont utilisées, 2 minutes et 4 minutes. Deux cycles de fonctionnement sont utilisés, un dans lequel le chemin est accessible durant 20 % du cycle et l'autre où le chemin est accessible pendant 80 % du cycle. Dans les quatre cas, le chemin est supprimé après être devenu inaccessible pour la deuxième fois. Une fois supprimé, il le reste jusqu'à ce que soit passée une certaine période de stabilité. Les chemins qui oscillent sur une période de 4 minutes ne sont plus supprimés dans les 9 à 11 minutes après être devenus stables. Les chemins avec une période d'oscillation de 2 minutes sont supprimés pour presque la période maximum de 15 minutes après être devenus stables.

4.8 Traitement de l'activité du protocole d'acheminement

Les paragraphes précédents se concentraient sur les paramètres de configuration et leur relation aux paramètres et matrices utilisés au moment du démarrage et sur la fourniture des algorithmes pour initialiser la mémorisation de démarrage. Ce paragraphe décrit les étapes de traitement des événements d'acheminement et des événements de temporisation lors du fonctionnement.

Les événements d'acheminement sont :
1. L'éveil pour la première fois d'un homologue BGP ou d'un nouveau chemin (ou après une longue période de sommeil) (paragraphe 4.8.1)
2. Un chemin devient inaccessible (paragraphe 4.8.2)
3. Un chemin redevient accessible (paragraphe 4.8.3)
4. Un chemin change (paragraphe 4.8.4)
5. Un homologue disparaît (paragraphe 4.8.5)

temps	**classement en fonction du temps (en minutes)**			
0.00	0.000 .	0.000 .	0.000 .	0.000 .
0.62	0.000 .	0.000 .	0.000 .	0.000 .
1.25	0.000 .	0.000 .	0.000 .	0.000 .
1.88	0.000 .	0.000 .	0.000 .	0.000 .
2.50	0.977 .	0.968 .	0.000 .	0.000 .
3.12	0.949 .	0.888 .	0.000 .	0.000 .
3.75	0.910 .	0.814 .	0.000 .	0.000 .
4.37	1.846 .	1.756 .	0.983 .	0.983 .
5.00	1.794 .	1.614 .	0.955 .	0.935 .
5.63	1.735 .	1.480 .	0.928 .	0.858 .
6.25	2.619 .	2.379 .	0.901 .	0.786 .
6.88	2.544 .	2.207 .	0.876 .	0.721 .
7.50	2.472 .	2.024 .	0.825 .	0.661 .
8.13	3.308 .	2.875 .	1.761 .	1.608 .
8.75	3.213 .	2.698 .	1.711 .	1.562 .

9.38	3.122	2.474	1.662	1.436
10.00	3.922	3.273	1.615	1.317
10.63	3.810	3.107	1.569	1.207
11.25	3.702	2.849	1.513	1.107
11.88	3.498	2.613	1.388	1.015
12.50	3.904	3.451	2.312	1.953
13.13	3.580	3.164	2.120	1.791
13.75	3.283	2.902	1.944	1.643
14.38	3.010	2.661	1.783	1.506
15.00	2.761	2.440	1.635	1.381
15.63	2.532	2.238	1.499	1.267
16.25	2.321	2.052	1.375	1.161
16.88	2.129	1.882	1.261	1.065
17.50	1.952	1.725	1.156	0.977
18.12	1.790	1.582	1.060	0.896
18.75	1.641	1.451	0.972	0.821
19.38	1.505	1.331	0.891	0.753
20.00	1.380	1.220	0.817	0.691
20.62	1.266	1.119	0.750	0.633
21.25	1.161	1.026	0.687	0.581
21.87	1.064	0.941	0.630	0.533
22.50	0.976	0.863	0.578	0.488
23.12	0.895	0.791	0.530	0.448
23.75	0.821	0.725	0.486	0.411
24.37	0.753	0.665	0.446	0.377
25.00	0.690	0.610	0.409	0.345

Figure 3 : Cycles très longs d'oscillation de chemin, répétés pendant 12 minutes, suivis d'une période de stabilité

La liste de réutilisation est utilisée pour donner le moyen d'évaluer rapidement un chemin qui a été supprimé, mais a été stable pendant assez longtemps pour être réutilisé, ou a été supprimé pendant assez longtemps pour qu'il puisse être traité comme un nouveau chemin. Les deux opérations suivantes sont décrites.

1. Insertion dans une liste de réutilisation (paragraphe 4.8.6)
2. Traitement de la liste de réutilisation toutes les delta-t secondes (paragraphe 4.8.7)

4.8.1 Traitement d'un nouvel homologue ou de nouveaux chemins

Lors de l'éveil d'un homologue, aucune action n'est requise si les chemins n'avaient pas d'historique d'instabilité antérieure, par exemple si c'est la première fois que l'homologue est activé et qu'il annonce ces chemins. Pour chaque chemin, le pointeur sur la structure d'atténuation serait mis à zéro et le chemin utilisé. La même action sera prise pour un nouveau chemin ou un qui n'a pas été sorti pendant assez longtemps pour que son classement atteigne zéro et que la structure d'atténuation soit supprimée.

4.8.2 Traitement des messages inaccessibles

Lorsque un chemin est retiré ou changé (le paragraphe 4.8.4 décrit comment traiter un changement) on utilise la procédure suivante.

Si il n'y a pas d'historique de stabilité antérieure (le pointeur de la structure d'atténuation est zéro) alors :

1. allouer une structure d'atténuation
2. établir le classement = 1
3. retirer le chemin

Autrement, si il existe une structure d'atténuation, alors :

1. établir t-diff = t-actuel – t-mis à jour
2. si (t-diff fait sortir de la matrice) {
   établir classement = 1
   }autrement {
   établir classement = classement *decay-array-ok [t-diff]+ 1
   si (classement > plafond) {
   établir classement = plafond
   }

}
3. retirer le chemin d'une liste de réutilisation si il est sur une
4. supprimer le chemin sauf si il l'est déjà.

Dans l'un et l'autre cas, alors :
1. établir t-mis à jour = t-actuel
2. insérer dans une liste de réutilisation (voir au paragraphe 4.8.6)

Si il y a un historique de stabilité, la précédente valeur de classement de stabilité est diminuée. Cela se fait en utilisant la matrice de diminution (decay-array). L'indice est déterminé en soustrayant l'heure actuelle de celle de la dernière mise à jour, puis en divisant par la granularité temporelle. Si l'indice est zéro, le classement est inchangé (pas de diminution). Si il est supérieur à la taille de la matrice, il est mis à zéro. Autrement utiliser l'indice pour aller chercher un élément de matrice de diminution et multiplier le classement par l'élément de matrice. Si la méthode d'entier normalisé est utilisée, descendre de la moitié d'un entier. Ajouter la pénalité normalisée pour un ou plusieurs inaccessibles (montrés ci-dessus comme 1). Si le résultat est supérieur au plafond, le remplacer pas la valeur plafond. Mettre maintenant à jour le champ Dernière mise à jour (de préférence en tenant compte du temps qui a été coupé avant de faire le calcul de diminution).

Lorsque un chemin devient inaccessible, les chemins de remplacement doivent être considérés. Ce processus est légèrement compliqué si des paramètres de configuration différents sont utilisés en présence ou absence de chemins de remplacement viables. Si tous ces chemins de remplacement ont été supprimés parce qu'ils avaient été précédemment un chemin de remplacement et si le nouveau changement de chemin change cette condition, les chemins de remplacement supprimés doivent être réévalués. Ils devraient être réévalués dans l'ordre normal de préférence de chemin. Lorsque on rencontre un de ces chemins de remplacement qui avait été supprimé mais est maintenant utilisable car il n'y a pas d'autre chemin de remplacement, il n'y a pas besoin de réévaluer d'autre chemin. Cela ne s'applique que si les chemins ont reçu des seuils de réutilisation différents, un à utiliser lorsque il y a un chemin de remplacement et un seuil plus élevé à utiliser lorsque la suppression du chemin résulterait à rendre la destination complètement inaccessible.

4.8.3 Traitement des annonces de chemin

Lorsque un chemin est ré-annoncé alors qu'il n'y a pas de structure d'atténuation, la procédure est alors la même qu'au paragraphe 4.8.1.

1. ne pas créer une nouvelle structure d'atténuation
2. utiliser le chemin

Si il existe une structure d'atténuation, le classement est diminué et les champs Classement et Dernière mise à jour sont mis à jour. On doit maintenant décider si le chemin peut être utilisé immédiatement ou doit être supprimé pendant un certain temps.

```
1. établir t-diff = t-actuel – t-mis à jour
2. si (t-diff fait sortir de la matrice) {
        régler classement = 0
      }autrement {
        régler classement = classement* decay-array-ng[t-diff]
      }
3. si ( non supprimé et classement < cut ) {
        utiliser le chemin
      } autrement
   si ( supprimé et classement < reuse) {
        régler l'état à ne pas supprimer
        retirer le chemin d'une liste de réutilisation
        utiliser le chemin
      }autrement {
        régler l'état à supprimé
        ne pas utiliser le chemin
        insérer dans une liste de réutilisation (voir le paragraphe 4.8.6)
      }
4. si ( classement > 0 ) {
        établir t-mis à jour = t-actuel
      }autrement {
        récupérer la mémoire pour la structure d'atténuation
        pointeur zéro sur la structure d'atténuation
      }
```

Si le chemin est réputé utilisable, on doit chercher quel est le meilleur chemin actuel. Le chemin nouvellement accessible est alors évalué selon les règles de choix de chemin du protocole BGP.

Si le nouveau chemin est utilisable, le meilleur chemin précédent est examiné. Avant les comparaisons de chemins, le meilleur chemin actuel peut devoir être réévalué si des jeux de paramètres distincts sont utilisés selon la présence ou l'absence d'un chemin de remplacement. Si il n'y avait pas de remplaçant, le meilleur chemin précédent peut être supprimé.

Si le nouveau chemin est à supprimer, il n'est placé sur une liste de réutilisation que si il aurait été préféré au meilleur chemin actuel si le nouveau chemin avait été accepté comme stable. Il n'y a pas de raisons de mettre en file d'attente un chemin sur une liste de réutilisation si après que le chemin devient utilisable il ne sera quand même pas utilisé à cause de l'existence d'un chemin préféré. Un tel chemin ne devrait pas avoir à être réévalué sauf si le chemin préféré devenait inaccessible. Comme spécifié ici, le chemin de moindre préférence serait réévalué et potentiellement utilisé ou potentiellement ajouté à une liste de réutilisation lors du traitement de la suppression d'un meilleur chemin de préférence plus élevée.

4.8.4 Traitement des changements de chemin

Si un chemin est remplacé par un routeur homologue qui fournit un nouveau chemin, le chemin qui est remplacé devrait être traité comme si un inaccessible avait été reçu (voir le paragraphe 4.8.2). Cela va arriver lorsque un homologue quelque part sur l'arrière du chemin d'AS commute continuellement entre deux chemins d'AS et que cet homologue n'atténue pas les oscillations de chemin (ou applique moins d'atténuation). Il n'y a pas de moyen pour déterminer si un chemin d'AS est stable et l'autre oscille, ou si ils oscillent tous les deux. Si le cycle est suffisamment court par rapport aux temps de convergence aucun des deux chemins par cet homologue ne livrera de paquet de façon très fiable. Comme il n'y a aucun moyen d'affecter l'homologue de telle façon qu'il choisisse celui des deux chemins d'AS qui est stable, la seule option viable est de pénaliser les deux chemins en considérant chaque changement comme un inaccessible suivi par une annonce de chemin.

4.8.5 Traitement d'une perte de routeur homologue

Lorsque une session d'acheminement entre homologue est rompue, soit tous les chemins individuels annoncés par cet homologue peuvent être marqués comme instables, soit la session d'échange de trafic elle-même peut être marquée comme instable. Marquer l'homologue va économiser une mémoire considérable. Comme les chemins individuels sont annoncés comme inaccessibles aux routeurs au delà du problème immédiat, l'état par chemin sera subi au delà de l'homologue immédiatement adjacent à la session BGP qui s'est arrêtée. Si l'instabilité continue, le routeur adjacent immédiat a seulement besoin de garder trace de l'historique de stabilité de l'homologue. Les routeurs au delà de ce point ne vont pas recevoir d'autre avertissement ou retrait de chemin et vont supprimer la structure d'atténuation au bout d'un certain temps.

La notification BGP par un attribut transitif facultatif que l'atténuation a déjà été appliquée pourra être examinée à l'avenir pour réduire le nombre de routeurs auxquels incombe la charge de mémorisation de la structure d'atténuation.

4.8.6 Insertion dans la liste de temporisateur de réutilisation

Les listes de réutilisation sont utilisées pour donner un moyen d'évaluation rapide des chemins qui ont été supprimés, mais qui ont été stables assez longtemps pour être réutilisés. La structure de données consiste en une série de têtes de liste. Chaque liste contient un ensemble de chemins qui sont programmés pour réévaluation approximativement au même moment. L'ensemble des têtes de listes de réutilisation est traité comme une matrice circulaire. Voir la Figure 4.

Une mise en œuvre de matrice circulaire des têtes de liste serait une matrice contenant les têtes de liste. Un décalage est utilisé lors de l'accès à la matrice. Le décalage va identifier la première liste. La $N^{ème}$ liste serait à l'indice correspondant à N plus le décalage modulo le nombre de têtes de listes. C'est cette conception qui sera supposée dans les exemples qui suivent.

Une exigence clé est d'être capable d'insérer une entrée dans la file d'attente la plus appropriée avec un minimum de calculs. Le calcul ne donne que la valeur actuelle du classement. Au lieu d'un calcul qui impliquerait un logarithme, la matrice de réutilisation (reuse-array[]) décrite au paragraphe 4.6 est utilisée. La matrice, l'échelle, et les bornes sont pré calculées pour transposer le classement en la plus proche tête de liste sans exiger le calcul d'un logarithme (voir au paragraphe 4.5).

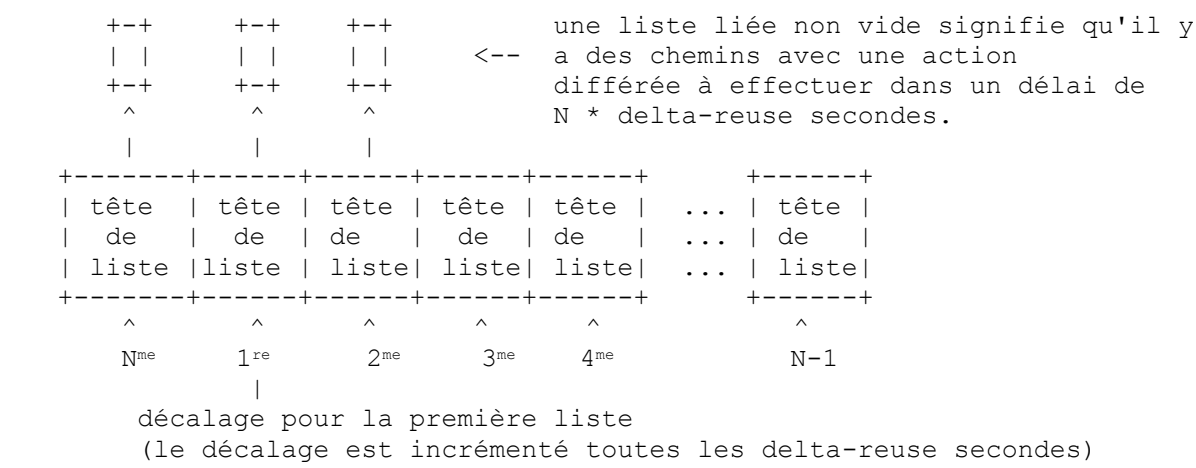

Figure 4 : Structures de données de liste de réutilisation

Noter que dans les paragraphes qui suivent, la notation du préfixe d'opérateur "modulo a b" signifie "b % a" en notation d'opérateur algébrique de langage C. Par exemple, "modulo 16 1023" serait 15.

1. normaliser le classement pour la recherche de matrice d'indice qui produit l'indice
2. comparer l'indice aux limites de la matrice
3. si (dans les bornes de la matrice) {
   régler indice = matrice de réutilisation [index]
   }autrement {
   régler indice = taille de liste de réutilisation -1
   }
4. insérer dans la liste
   reuse-list[modulo reuse-list-size (indice + décalage)]

Choisir la liste de réutilisation correcte implique seulement une multiplication et un glissement pour faire la normalisation, une troncature d'entier, puis une recherche de matrice dans la matrice de réutilisation (reuse-array[]). La valeur restituée par la matrice de réutilisation est utilisée pour choisir une liste de réutilisation. La liste de réutilisation est une liste circulaire. La méthode la plus courante pour mettre en œuvre une liste circulaire est d'utiliser une matrice et d'appliquer un décalage et une opération de modulo pour obtenir l'entrée de matrice correcte. Le décalage est incrémenté pour faire tourner la liste circulaire.

4.8.7 Traitement des événements de temporisateur de réutilisation

La granularité du temporisateur de réutilisation devrait être plus grosse que celle du temporisateur de diminution. Il en résulte que lorsque le temporisateur de réutilisation arrive à expiration, les chemins supprimés devraient être diminués de plusieurs incréments de temps de diminution. Certains calculs peuvent être évités en insérant toujours dans la liste de réutilisation l'éligibilité à la réutilisation correspondant à un incrément de temps passé. Dans les cas où les listes de réutilisation ont une "mémoire" plus longue que la "mémoire de diminution" (décrite ci-dessus) tous les chemins dans la première file d'attente seront disponibles pour réutilisation immédiate si accessible ou l'entrée d'historique pourrait être éliminée si ils sont inaccessibles.

Lorsque c'est le moment de faire avance les listes, la première file d'attente sur la liste de réutilisation doit être traitée et la file d'attente circulaire doit pivoter. En utilisant une matrice et un décalage comme une matrice circulaire (comme décrit au paragraphe 4.8.6) l'algorithme ci-dessous est répété toutes les delta-reuse secondes.

1. garder un pointeur sur la tête actuellement à zéro de la file d'attente et mettre à zéro l'entrée de tête de liste
2. régler décalage = modulo reuse-list-size (décalage + 1) faisant par là tourner la file d'attente circulaire des têtes de liste
3. si (le pointeur de tête de liste gardé n'est pas vide)
   pour chaque entrée {
   régler t-diff = t-actuel – t-mis à jour
   régler classement = classement *decay-array-ok [t-diff]
   régler t-mis à jour = t-actuel
   si (classement < reuse)
   réutiliser le chemin
   autrement
   ré-insérer dans une autre liste (voir le paragraphe 4.8.6)
   }

La valeur de la tête de liste mise à zéro serait sauvegardée et l'entrée de matrice elle-même mise à zéro. Les têtes de liste seraient alors avancées par l'incrémentation du décalage. En commençant par la tête sauvegardée de la vieille liste mise à zéro, chaque chemin serait réévalué et utilisé, supprimé entièrement ou remis en file d'attente si il ne se trouve pas prêt pour réutilisation. Si un chemin est utilisé, il doit être traité comme si il était une nouvelle annonce de chemin, comme décrit au paragraphe 4.8.3.

5. Expérience de mise en œuvre

Les premières mises en œuvre "d'atténuation d'oscillation de chemin" ont été le codage du démon de serveur d'acheminement (RSD, *route server daemon*) par Ramesh Govindan (ISI) et la mise en œuvre IOS de Cisco par Ravi Chandra. Les deux mises en œuvre sont devenues disponibles en 1995 et ont connu une utilisation extensive. La mise en œuvre du rsd a été utilisée dans les serveurs d'acheminement sur les points d'accès réseau (NAP, *Network Access Point*) financés par la NSF et à d'autres interconnexions majeures de l'Internet. La version IOS de Cisco a été utilisée par les fournisseurs d'accès Internet du monde entier. La mise en œuvre du rsd a été intégrée dans les livraisons de gated (voir http://www.gated.org) et est disponible dans les routeurs commerciaux qui utilisent gated.

Cela fait maintenant plus de deux années d'expérience de déploiement de l'atténuation d'oscillation de chemin BGP. Certains problèmes sont survenus dans le déploiement. Jusqu'à présent, ils ont pu être résolus par une mise en œuvre soigneuse de l'algorithme et un déploiement attentif. Dans certaines topologies, le déploiement coordonné peut être utile et dans tous les cas, la divulgation de l'utilisation de l'atténuation de l'oscillation de chemin et des paramètres utilisés est très bénéfique au débogage des problèmes de connexité.

Certains des problèmes sont survenus à cause de subtiles erreurs de mise en œuvre. L'atténuation de chemin ne devrait jamais être appliquée sur des chemins appris par IBGP. Le faire ouvre la possibilité de boucles d'acheminement persistantes. Lorsque les chemins IBGP au sein d'un AS sont incohérents, des chemins en boucle peuvent facilement se former. Supprimer des chemins appris par IBGP cause de telles incohérences. Les mises en œuvre devraient interdire la configuration d'atténuation de chemin sur les homologues IBGP. Les pénalités pour instabilité ne devraient être appliquées que lorsque un chemin est retiré ou remplacé et non lorsque un chemin est ajouté. Si les paramètres d'atténuation sont appliqués de façon cohérente, cette contrainte de mise en œuvre va résulter en un chemin secondaire stable qui est préféré à un chemin principal instable à cause de l'atténuation du chemin principal près de la source.

Dans les topologies où il existe plusieurs chemins d'AS pour une certaine destination, les fluctuations du chemin principal peuvent résulter en la suppression du chemin secondaire. Cela peut survenir si aucune atténuation n'a été faite près de la cause de l'oscillation du chemin ou si l'atténuation est appliquée plus agressivement par un AS distant. Ce problème peut être résolu d'une des deux façons suivantes. L'atténuation peut être faite près de la source de l'oscillation et les paramètres d'atténuation peuvent être rendus cohérents. Autrement, un AS distant qui souhaite des paramètres d'atténuation plus agressifs peut désactiver la pénalisation des chemins au changement de chemin d'AS, ne pénalisant les chemins que si ils sont complètement retirés. Pour faire ainsi, la mise en œuvre doit prendre en charge cette option (décrite en 4.4.3).

L'oscillation de chemin devrait être atténuée près de la source. Les destinations à un seul rattachement peuvent être couvertes par des chemins statiques. L'agrégation donne d'autres moyens d'atténuation. Les fournisseurs devraient atténuer leurs propres problèmes en interne ; cependant, l'atténuation sur l'origine de l'état de liaison IGP n'est pas encore mise en œuvre par les fabricants de routeurs. Les fournisseurs qui utilisent plusieurs AS au sein de leur propre topologie devraient atténuer entre leurs propres AS. Les fournisseurs devraient atténuer les AS des fournisseurs adjacents.

L'atténuation donne un moyen de limiter la propagation de changements de chemin excessifs lorsque la connexité est très intermittente. Une fois qu'un problème est corrigé, l'état d'atténuation correspondant aux préfixes connus pour être atténués du fait du problème qu'on vient de résoudre peut être arrangé à la main. Afin de déterminer où l'atténuation peut s'être produite après des problèmes de connexité, les fournisseurs devraient publier leurs paramètres d'atténuation. Les fournisseurs devraient accepter de supprimer manuellement l'atténuation sur des préfixes ou chemins d'AS spécifiques à la demande d'autres fournisseurs lorsque la demande est accompagnée d'assurances crédibles que le problème a été réellement réglé.

En atténuant leurs propres informations d'acheminement, les fournisseurs peuvent réduire leur propre besoin de faire des demandes aux autres fournisseurs de supprimer un état d'atténuation après avoir corrigé un problème. Les fournisseurs devraient être proactifs et surveiller les préfixes et chemins qui sont supprimés en plus de la surveillance des états de liaisons et de l'état de session BGP.

Remerciements

Ce travail et le présent document n'auraient pas pu être menés à bien sans les conseils, commentaires et encouragements de Yakov Rekhter (Cisco). Dennis Ferguson (MCI) a fourni une description des algorithmes de la mise en œuvre de BGP et de nombreux commentaires et conseils précieux. David Bolen (ANS) et Jordan Becker (ANS) ont fourni de précieux commentaires, particulièrement en ce qui concerne les premières simulations. Plus de quatre années se sont écoulées entre le projet initial présenté au groupe de travail BGP (en octobre 1993) et la présente réalisation. Au moment de cette rédaction, il y a une expérience significative avec deux mises en œuvre, chacune ayant été développée depuis 1995. L'une d'elles était conduite par Ramesh Govindan (ISI) pour le projet NSF Routing. La seconde été menée par Ravi Chandra (Cisco). Sean Doran (Sprintlink) et Serpil Bayraktar (ANS) ont été parmi les premiers à faire des essais indépendants de la mise en œuvre Cisco pre-beta. Des retours de mise en œuvre sous forme de commentaires précieux ont été apportés par de nombreuses personnes du groupe de travail IDR de l'IETF et du groupe de travail RIPE Routing ainsi que de NANOG et IEPG.

Merci aussi à Rob Coltun (Fore Systems), Sanjay Wadhwa (Fore), John Scudder (IENG), Eric Bennet (IENG) et Jayesh Bhatt (Bay Networks) pour avoir relevé des erreurs mathématiques non découvertes durant le codage des plus récentes mises en œuvre. Ces erreurs sont apparues dans les détails des sections de suggestion de mise en œuvre écrites après l'achèvement des deux premières mises en œuvre. Merci aussi à Vern Paxson pour sa très attentive relecture dont ont résulté de nombreuses précisions dans le document.

Références

[ISO10747] ISO/IEC 10747 - information technology - telecommunications and information exchange between systems - protocol for exchange of inter-domain routeing information among intermediate systems to support forwarding of iso 8473 pdus. Technical report, International Organization for Standardization, août 1994. ftp://merit.edu/pub/iso/idrp.ps.gz

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

[RFC1268] Y. Rekhter et P. Gross, "Application du protocole BGP dans l'Internet", octobre 1991. *(Historique)*

[RFC1520] Y. Rehkter et C. Topolic, "Échange d'informations d'acheminement à travers les frontières du fournisseur dans l'environnement CIDR ", septembre 1993. *(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.)*

[RFC1773] P. Traina, "Expérience avec le protocole BGP-4", mars 1995. *(Information)*

[RFC1774] P. Traina, éd., "Analyse du protocole BGP-4", mars 1995] *(Information)*

Considérations pour la sécurité

Les pratiques présentées dans ce document n'affaiblissent pas la sécurité des protocoles d'acheminement. Le déni de service est possible dans un environnement d'acheminement par ailleurs non sûr mais ces pratiques contribuent seulement à la persistance de telles attaques et n'impactent pas les méthodes de prévention et les méthodes de détermination de la source.

Adresse des auteurs

Curtis Villamizar
ANS
mél : curtis@ans.net

Ravi Chandra
Cisco Systems
mél : rchandra@cisco.com

Ramesh Govindan
ISI
mél : govindan@isi.edu

Déclaration complète de droits de reproduction

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