Stabilité dans BGP

Ken Schumacher[1]

LIRMM - Université de Montpellier II - 2004

Encadrants : Jean-Claude König[2] et Ehoud Ahronovitz[3]
Responsable Eurécom : Guillaume Urvoy-Keller

1. ken.schumacher@enst.fr
2. konig@lirmm.fr
3. aro@lirmm.fr

Résumé

Le protocole BGP est le protocole le plus utilisé actuellement dans Internet pour échanger des informations de routages entre systèmes autonomes, mais celui-ci souffre d'instabilités.

Nous étudierons tout d'abord le protocole BGP et proposerons un modèle. Ensuite nous présenterons et essayerons de résoudre un premier type d'instabilité qui intervient à l'intérieur même des systèmes autonomes avec l'utilisation de l'attribut MED. Finalement nous étudierons des instabilités entre systèmes autonomes dues à l'utilisation du filtrage d'annonce et proposerons un modèle.

The BGP protocol is currently the most widely used protocol in the context of Internet when exchanging routing information between autonomous systems, but it tends to be unstable.

We will first study the BGP protocol, and will propose a model. We will then go on to present and attempt to solve a first type of instability which occurs inside autonomous systems when using the MED attribute. We will then study instabilities occurring between autonomous systems, which are due to the use of announcement filtering, for which we will also propose a model.

Table des matières

Table des figures

Chapitre 1

Introduction

Le protocole BGP (Border Gateway Protocol) est le protocole actuellement utilisé dans l'Internet pour échanger des informations de routage entre systèmes autonomes. Mais il souffre d'incohérences et d'instabilités. Nous essayerons ici de les caractériser et de proposer des améliorations en vu de rendre BGP plus stable.

Au fur et à mesure de l'accroissement d'Internet, il a fallu adapter des algorithmes et protocoles de routage en tenant compte de l'évolution du réseau. Le protocole GGP (Gateway to Gateway Protocol) utilisait de grandes tables de routage et énormément de bande passante pour véhiculer les informations d'accessibilité, une panne dans le réseau entraînait un recalcul de toutes les distances et de toutes les routes. D'autre part, la vision centralisée et unifiée ne permettait plus de gérer convenablement le réseau, une modification d'un protocole ou la recherche d'une panne devenait un exercice périlleux. Il fallait découper et hiérarchiser le réseau.

Le protocole EGP (Exteral Gateway Protocol) [ECRI82, Mil84] a été conçu dans le but de résoudre ces problèmes. La notion de système autonome (AS) a été alors introduite, c'est "un ensemble de routeurs et de réseaux sous une administration unique". Sous cette définition se cache un concept bien vague, en fait un AS regroupe un ensemble de routeurs interconnectés (deux routeurs d'un même AS peuvent communiquer sans traverser un autre AS). Une autre vision est que chaque AS est géré de manière indépendante et un protocole commun permet d'interconnecter, d'échanger des informations de routage entre AS. EGP est donc le protocole commun qui va permettre d'échanger ces informations. Son principe est très simple : chaque routeur EGP va envoyer périodiquement à ses voisins sa table de routage qui contient les destinations possibles ainsi que la "distance" qui le sépare de chacune d'elle. Cette distance n'est pas forcément calculée de la même manière d'un AS à un autre comme elle l'est dans RIP ou OSPF, car chaque AS est libre de choisir son protocole de routage interne et donc sa notion de distance ; cela permet plus de liberté mais a pour inconvénient de créer des incohérences. Effectivement si un AS reçoit une route, c'est la meilleure suivant les critères de l'AS qui a envoyé la route mais pas forcément

la meilleure suivant les critères de cet AS.

L'évolution rapide d'Internet a obligé à dépasser les limites d'EGP. Tout d'abord il est trop contraignant, la topologie du réseau doit être en arbre pour éviter la formation de boucles. Ensuite il n'a pas été conçu pour supporter une telle augmentation du nombre de réseaux. Bien qu'une panne ait moins d'incidences que dans GGP, il est toutefois nécessaire d'envoyer la totalité des tables ce qui représente une lourde charge pour le réseau. Finalement il n'est pas compatible avec les besoins actuels de notre société où la bande passante est devenue un bien commercial. En d'autres termes, il est difficile dans EGP d'indiquer des préférences suivant des choix économiques ou de confiance. Il était donc nécessaire de créer un nouveau protocole de communication entre systèmes autonomes tenant compte de ces choix économiques et de confiance, le protocole BGP (Border Gateway Protocol).

Tout comme EGP, BGP permet d'échanger les informations de routages entre AS. Mais ce n'est plus un protocole à vecteur de distances mais à vecteur de chemins, cela pour détecter plus facilement les boucles. D'autre part, la notion de politique a pris place pour que le "meilleur chemin" ne soit pas forcément le plus rapide ou celui qui traverse le moins de routeurs.

Dans le chapitre 2, nous présentons le protocole BGP et proposons un modèle pour notre étude. Dans le chapitre 3, nous étudions un cas d'instabilité interne aux systèmes autonomes (des oscillations entre plusieurs routes) et proposons des solutions simples. Dans le chapitre 4, nous voyons un autre type d'instabilité qui peut apparaître entre les AS. Finalement, dans le chapitre 5, nous proposons un modèle pour étudier ces instabilités.

Chapitre 2

Le protocole BGP

2.1 Principes généraux

L'objectif principal de BGP est de pouvoir échanger des informations de routage entre les différents organismes indépendants appelés systèmes autonomes, c'est-à-dire un ensemble de routeurs sous une même entité administrative, nous avons introduit cette notion dans le chapitre précédent avec EGP.

Un annonceur BGP est un périphérique réseau qui utilise le protocole BGP, ce n'est pas forcément un routeur, ça peut être par exemple un serveur de route. Il est composé de tables où sont stockées les informations de routage, d'un mécanisme pour sélectionner la "meilleure route" à destination d'un préfixe, c'est à dire d'un réseau, d'un mécanisme pour filtrer les routes apprises et annoncées (aussi appelé politique d'annonce) ainsi que d'une méthode pour échanger entre voisins les informations connues.

La granularité du routage est le système autonome, c'est à dire qu'un AS ne connaît pas tous les routeurs traversés pour atteindre une destination, mais seulement le numéro des AS traversés.

Le "Border Gateway Protocol" (BGP) a été introduit pour la première fois en 1989 dans le RFC 1105 [KL89] puis a subi plusieurs évolutions ([KL90], [KL91]). Nous sommes actuellement à la version 4 du protocole, défini dans [YR94] et corrigé dans le RFC 1771 [YR95].

2.2 Modèle très simple de BGP

Nous allons commencer par donner un modèle très simple de BGP, nous approfondirons ensuite le fonctionnement de BGP entre système autonomes, puis nous verrons le fonctionnement de BGP à l'intérieur même d'un AS.

2.2.1 Interactions inter-AS

Modélisation des AS

Chaque système autonome peut contenir beaucoup d'annonceurs BGP, de routeurs non BGP..., mais d'un point de vue extérieur, nous pouvons les considérer comme un et un seul routeur BGP possédant un grand nombre de ports car tous les annonceurs BGP appartenant à un même AS doivent apparaître comme identiques vue de l'extérieur. Il est donc possible de modéliser un réseau comme étant un graphe où chaque système autonome est représenté par un noeud et les arêtes modélisent les liens (physiques) de communication entre les AS. Comme il peut y avoir plusieurs arêtes entre deux noeuds, il s'agir plutôt d'un multigraphe. Par exemple la figure 2.2 est la représentation du réseau général 2.1.

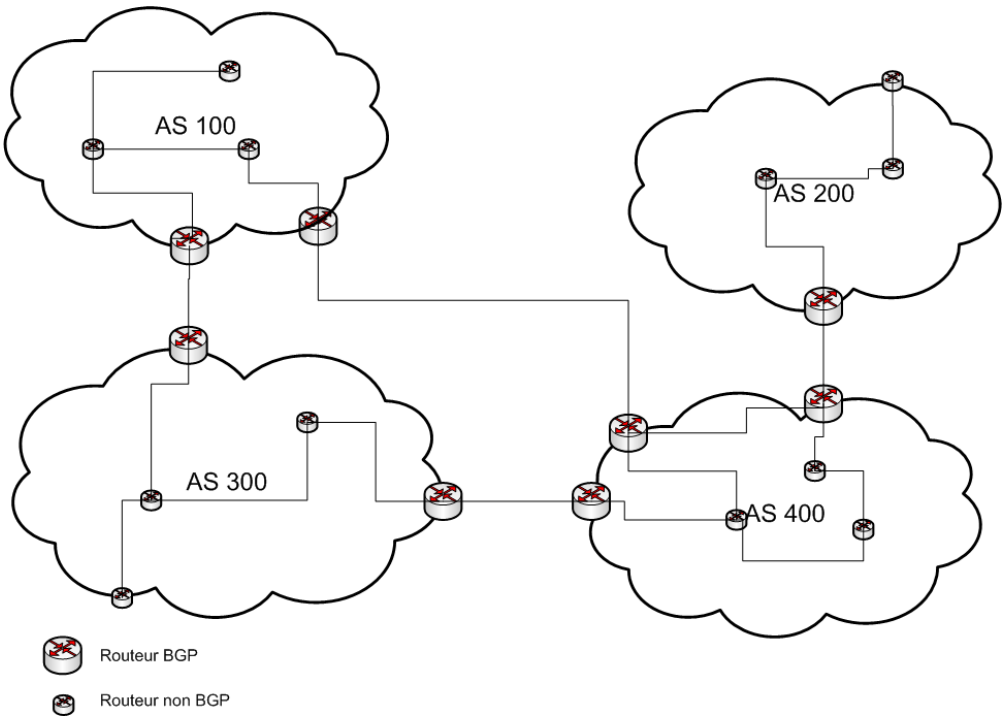

FIG. 2.1 – *Exemple d'un réseau quelconque*

Identification des AS

Chaque système autonome est identifié par un numéro unique et chaque AS connaît le numéro de tous ses voisins (information transmise à l'ouverture d'une session BGP entre deux annonceurs BGP), ce numéro est donné par des autorités, les mêmes que celles qui sont en charge de la distribution des adresses IP (RIPE, APNIC, ARIN...).

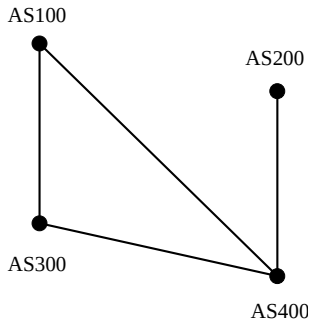

FIG. 2.2 – *Modèle BGP inter-AS*

Fonctionnement des annonces de routes

Un réseau qui désire se faire connaître, informe ses voisins BGP de son existence ; ceux-là peuvent ignorer l'information, ou la prendre en compte et la réannoncer à tous ou une partie des voisins (sauf au routeur qui a envoyé l'annonce) ; **si un AS annonce une route, il doit par la suite accepter le trafic vers cette destination**. Si un AS ne veut pas servir de réseau de transit, il ne doit pas réannoncer la route.

Attributs d'une route

Une annonce de route est composée d'un NLRI (Network Layer Reachability Information) c'est-à-dire la destination à laquelle correspond l'annonce de la route. Elle peut aussi contenir un ensemble de routes "à oublier" dans le cas par exemple où un lien tombe en panne. D'autre part, BGP peut agréger des routes. C'est une technique permettant de réduire le nombre de lignes dans les tables de routage en compactant les préfixes des réseaux. Par exemple, si un AS est en charge de tous les réseaux ayant les préfixes 198.234.0 jusqu'à 198.234.255, il peut les agréger et annoncer seulement le préfix 198.234. Une route est accompagnée d'informations ou attributs. Nous y trouvons :

ORIGIN indique l'origine de la route (interne à l'AS d'origine, apprise par le protocole EGP, ou inconnue) ce paramètre n'est pratiquement plus utilisé.

AS_PATH indique la route suivant une liste ordonnée d'AS. S'il y a eu agrégation de route, tous les AS traversés par toutes les routes contenues dans l'agrégation sont indiqués mais pas forcément dans l'ordre.

NEXT_HOP indique l'IP du prochain routeur BGP (utile dans le cas ou plusieurs annonceurs partagent le même lien de communication, par exemple un BUS) ce paramètre n'est pas utile dans notre modèle.

LOCAL_PREF (optionnel) est un attribut interne à l'AS, il n'est jamais communiqué aux autres AS. Il pondère dans l'AS la priorité donnée aux routes en d'autres termes il accorde un degré de préférence à chaque route.

ATOMIC_AGGREGATE (optionnel) indique s'il y a eu agrégation de routes.

AGGREGATOR (optionnel) indique l'AS ainsi que l'IP du routeur qui a effectué l'agrégation.

MULTI_EXIT_DISC (MED) (optionnel) permet que lorsque deux AS sont interconnectés à l'aide de plusieurs liens, d'en discriminer en associant à chaque lien un degré de préférence. Le premier AS propose une valeur de MED, le voisin calcul le sien et prend le plus petit des deux. Ce paramètre n'est pas utile lorsque deux AS ne sont pas multi connectés. Sa portée est limitée à l'AS ainsi que l'AS voisin avec qui un MED est négocié ; il n'est en aucun cas réannoncé aux autres AS.

La liste des attributs n'est pas figée, elle est ouverte à toute proposition, le RFC 2042 [Man97] propose certaines évolutions.

Unicité des routes

Si un réseau reçoit deux routes vers une même destination il n'en sélectionne qu'une, suivant des métriques que nous verrons plus tard mais qui n'est pas forcément le plus court chemin. D'autre part BGP a été conçu pour éviter les boucles, concrètement un AS qui reçoit une route vérifie qu'il n'est pas déjà dans le chemin ; s'il l'est il détruit l'information.

Modèles de communication entre AS

En résumé pour notre modèle nous devons savoir que chaque noeud contient une liste de chemins. Chacun étant identifié par la liste des AS à traverser pour atteindre la destination, d'un degré de préférence interne (LOCAL_PREF) et d'une métrique permettant de choisir un lien lorsque deux AS sont multi connectés (MED). Ensuite lorsque qu'une destination d désire se faire connaître, l'AS en charge de cette destination envoie l'information à ses AS voisins, puis l'information se transmet de proche en proche avec au final un arbre recouvrant dont la racine est la destination d, les noeuds sont les AS qui peuvent communiquer avec celle-ci et les arcs représentent les routes empruntées par les paquets à destination de d. Cette structure reste arborescente car pour q'il y ait une route xy (x et y étant des noeuds), il faut qu'il y ait la route yd ; si yd venait à disapraitre la route xy serait éliminée. D'autre part BGP élimine automatiquement les cycles.

2.2.2 Modèle du routeur unique

Nous avons vu précédemment qu'un AS peut être vu comme un seul routeur BGP, nous allons voir maintenant ce qui se passe à l'intérieur de ce routeur.

Tables BGP

Un annonceur BGP est constitué de trois tables Adj-RIBs-In, Loc-RIB et Adj-RIBs-Out qui servent respectivement à conserver les annonces entrantes, les meilleurs chemins et les informations à annoncer. Pour simplifier, dans notre modèle nous n'en utiliserons qu'une seule contenant la liste des routes (équivalante à Adj-RIBs-In) et nous ajoutons un indicateur pour simuler la deuxième. Adj-RIBs-Out n'est pas utilisé car dans un premier temps nous réannonçons toutes les routes sélectionnées. Chaque ligne correspond à une route et contient les informations suivantes : la destination (NLRI), un indicateur permettant de savoir si la route a été sélectionnée, le prochain routeur, la liste des AS à traverser (AS_PATH), le MED, et le degré de préférence interne à l'AS d'une route (LOCAL_PREF).

Réception et annonce d'une route

Lorsqu'un routeur BGP reçoit l'annonce d'une nouvelle route, il applique la politique de filtrage en entrée. Celle-ci peut éliminer le chemin ou modifier la valeur du LOCAL_PREF. D'autre part si l'AS auquel appartient le routeur est déjà dans la liste des AS traversés, c'est qu'il y a eu création d'une boucle, il élimine donc immédiatement cette annonce. S'il a décidé de conserver cette route, il l'ajoute à la table de routage. Attention il n'y a qu'une ligne par couple {annonceur BGP voisin, destination}, en d'autres termes, si un voisin a déjà annoncé une route, et qu'il en réannonce une pour la même destination, mais avec d'autres paramètres, il faut écraser l'ancienne annonce et relancer le processus de décision.

Processus de sélection

Nous allons introduire deux nouvelles notions : E-BGP et I-BGP. Ce sont deux parties du protocole BGP. E-BGP est chargé des communications entre AS et I-BGP des communications interne à l'AS. Nous verrons plus tard les différences entre ces deux protocoles.

Le processus de décision consiste à sélectionner la "meilleure route" parmi les chemins possibles suivant le schéma ci-dessous ; il s'arrête dès qu'il n'y a plus qu'une seule route dans la liste des possibilités :

1. Choisir la route (ou les routes en cas d'égalité) ayant le plus grand LOCAL_PREF
2. Choisir les routes avec le moins d'AS dans l'AS_PATH (les routes qui traversent le moins d'AS)
3. Pour chaque voisin, sélectionner les routes qui ont le plus petit MED
4. S'il reste au moins une route E-BGP, c'est à dire annoncée par un AS voisin et non un routeur du même AS, éliminer toutes les routes qui passent au travers de l'AS (route annoncée en I-BGP)

5. Choisir les routes avec un coût IGP minimal (cela ne nous concerne pas pour le moment puisqu'un AS est représenté par un noeud, il n'y a donc pas de coût IGP)
6. Utiliser une information déterministe, par exemple, prendre la route qui a l'adresse IP la plus grande dans l'attribut NEXT_HOP.

Si la route était déjà connue et qu'elle reste la meilleure, le routeur BGP ne fait rien. Si la nouvelle route est meilleure ou que la destination n'était pas connue, le routeur ajoute son numéro d'AS à l'AS_PATH, efface la valeur du MED et remplace le NEXT_HOP par son adresse IP. Ensuite il entame le processus de réannoncement qui consiste à appliquer d'autres politiques de filtrage pour sélectionner les voisins qu'il informera de la présence de cette route. **Une route annoncée à un AS voisin entraîne l'obligation d'accepter le trafic provenant de ce voisin et à destination de l'originaire de la route**. Parallèlement si l'ancienne route n'est plus valable et que le routeur BGP ne désire pas annoncer la nouvelle route à un voisin, il lui envoie un message indiquant de supprimer l'ancienne route.

2.2.3 Exemple de fonctionnement de BGP

L'exemple 2.3 montre le fonctionnement de BGP lorsqu'un réseau (193.29.108.0/24 appartenant à l'AS100) désire se faire connaître. Nous n'avons pas mis la colonne LOCAL_PREF car dans cet exemple nous supposons qu'aucune politique n'a été définie.

Notations :
- Un plus dans la colonne "Best_path" indique que c'est un nouveau meilleur chemin donc qu'il faut le traiter.
- Un moins dans la colonne "Best_path" indique l'ancien meilleur chemin.
- Le MED est l'attribut qui permet de discriminer un lien entre deux AS ; seuls R_a et R_d sont concernés puisque ce sont les seuls à avoir plusieurs liens entre eux. Le MED est modélisé sur le schéma par les valeurs entre parenthèses.
- NEXT_HOP est le routeur de d'origine de l'annonce.
- NLRI est le préfixe du réseau à qui correspond l'annonce (dans notre cas c'est 193.29.108.0/24).
- AS_PATH est la liste des AS traversés.
- Nous avons séparé R_a en R_{a1} et R_{a2} car il y a deux liens et donc deux interfaces, comme nous gardons les annonces apprises par chaque interface, il était nécessaire de les différencier.

Etape a : situation initiale

Au début de notre exemple toutes les tables sont vides et nous voulons propager l'information que le réseau 193.29.108.0/24 existe et appartient à R_a

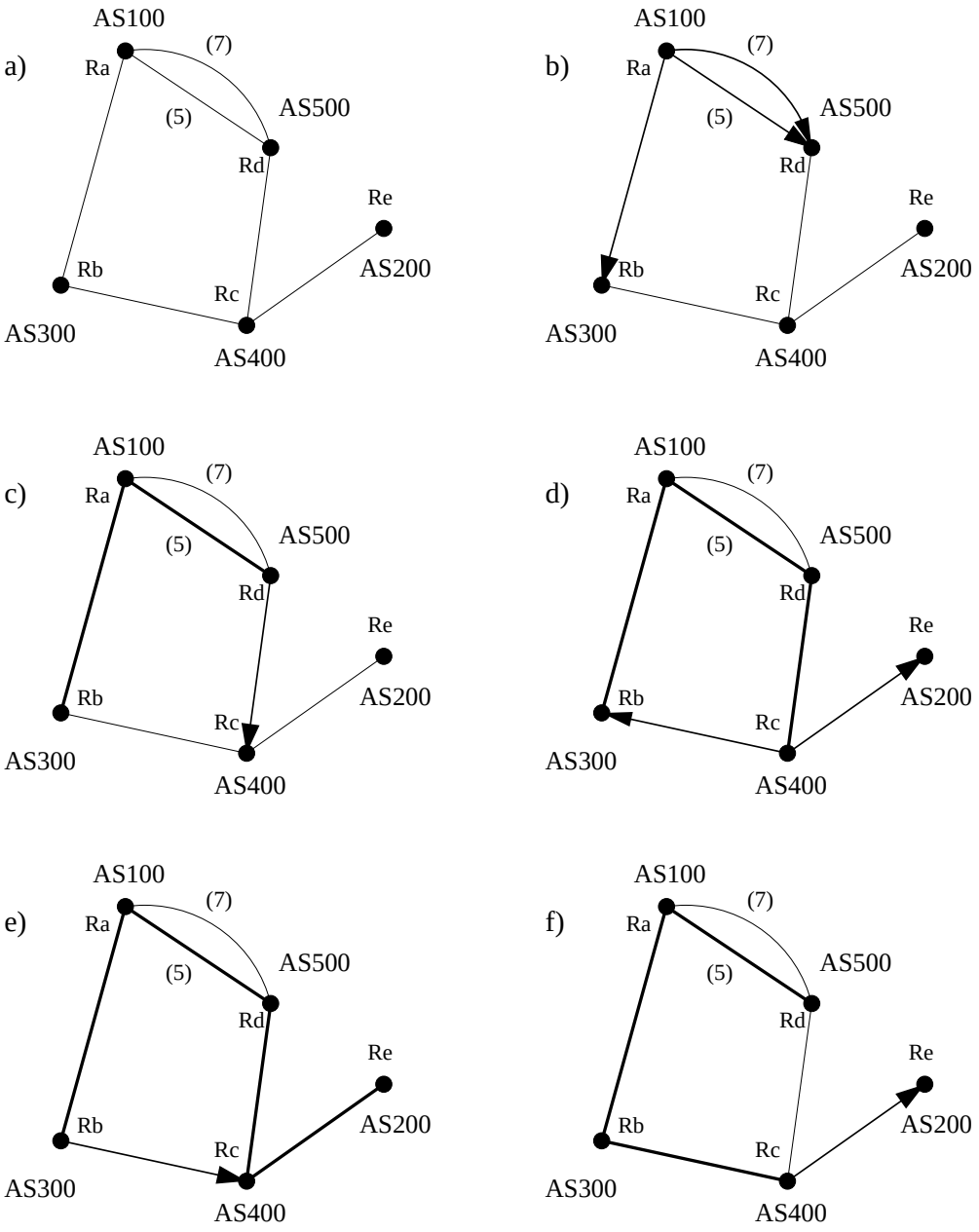

FIG. 2.3 – *Exemple de la propagation d'un chemin*

Etape b : R_a annonce à R_b et R_d le nouveau réseau

Le routeur R_a voulant annoncer le réseau 193.29.108.0/24 envoie la nouvelle route (193.29.108.0/24 ; R_a ; 100) à ses voisins R_b et R_d. R_b est dans l'état suivant :

Routeur	NLRI	Best_path	NEXT_HOP	AS_PATH	MED
R_b	193.29.108.0/24	+	R_a	100	()

R_d reçoit l'annonce en double puisqu'il est connecté avec deux liens à R_a mais il n'a pas la même valeur de MED sur les deux chemins. R_d se trouve dans l'état suivant :

Routeur	NLRI	Best_path	NEXT_HOP	AS_PATH	MED
R_d	193.29.108.0/24	+	R_{a1}	100	(5)
R_d	193.29.108.0/24		R_{a2}	100	(7)

Etape c : R_d fait suivre l'annonce à R_c

R_d a sélectionné l'annonce de R_{a1} car elle a un plus petit MED, il en averti R_c en envoyant une annonce pour la route (193.29.128.0/24 ; R_d ; 100,500). R_c la reçoit, il la considère comme meilleur chemin (puisqu'il n'en a pas d'autre). A cette étape là, le routeur R_c est dans l'état :

Routeur	NLRI	Best_path	NEXT_HOP	AS_PATH	MED
R_c	193.29.108.0/24	+	R_d	100,500	()

Etape d : R_c envoie l'annonce à R_b et R_e

R_c envoie une annonce de nouveau chemin à R_b et R_e contenant le chemin (193.29.128.0/24 ; R_c ; 100,500,400). R_e est dans l'état :

Routeur	NLRI	Best_path	NEXT_HOP	AS_PATH	MED
R_e	193.29.108.0/24	+	R_c	100,500,400	()

Et R_b, qui a déjà effectué son processus de décision de l'étape b mais n'a pas encore informé ses voisins, garde R_a comme meilleure route puisqu'elle traverse moins d'AS :

Routeur	NLRI	Best_path	NEXT_HOP	AS_PATH	MED
R_b	193.29.108.0/24	-	R_a	100	()
R_b	193.29.108.0/24		R_c	100,500,400	()

R_b va annoncer sa meilleure route non pas parce qu'il a reçu une nouvelle annonce (car son choix n'est pas modifié) mais parce qu'il n'avait pas eu le temps de le faire jusqu'a présent.

Etape e : R_b envoie sa première sélection à R_c

Donc R_c reçoit l'annonce et se retrouve dans l'état suivant :

Routeur	NLRI	Best_path	NEXT_HOP	AS_PATH	MED
R_c	193.29.108.0/24	-	R_d	100,500	()
R_c	193.29.108.0/24	+	R_b	100,300	()

R_c compare la route par R_b et celle par R_d. Ayant effectué toutes les étapes de sélection et n'ayant toujours que deux chemins, il choisit de manière déterministe R_b (car R_b a une IP plus grande, par exemple). Donc 100,300 devient sa meilleure route. Il en informe R_d et R_e.

Etape f : R_c annonce le chemin par R_b à R_d et R_e

L'annonce de R_c a pour effet de remplacer sans condition la seule route connue chez R_e car on ne garde que la dernière annonce qui arrive d'un routeur (plus précisément d'une interface d'un routeur). R_d quant à lui se retrouve dans l'état suivant :

Routeur	NLRI	Best_path	NEXT_HOP	AS_PATH	MED
R_d	193.29.108.0/24	-	R_{a1}	100	(5)
R_d	193.29.108.0/24		R_{a2}	100	(7)
R_d	193.29.108.0/24		R_c	100,300,400	()

Le nouveau chemin n'est pas meilleur donc il reste dans l'état précédent sans envoyer de message.

Etat final

L'annonce de la nouvelle destination possible est terminée, nous avons construit un arbre sur l'AS100 et chaque noeud (AS) est capable de communiquer avec le réseau 193.29.108.0/24.

2.2.4 Interaction Intra-AS

Nous avons vu précédemment comment fonctionnait un AS vu de l'extérieure, mais en réalité un AS est composé d'une multitude de routeurs et d'annonceurs BGP, ainsi que de routeurs non BGP. Dans la version de base du protocole, chaque annonceur BGP est directement relié avec tous les autres routeurs de l'AS, en d'autres termes, il y a une session BGP entre chaque paire d'annonceurs. Le graphe des annonceurs est une clique. Cela ne veut pas pour autant dire qu'un lien physique entre eux soit nécessaire, il suffit que le réseau soit connexe ce qui est la définition de base d'un AS. La figure 2.4 représente un AS quelconque avec son modèle BGP.

Différences entre E-BGP et I-BGP

Le fonctionnement de BGP à l'intérieur d'un AS (I-BGP) reste très proche de son fonctionnement entre AS (E-BGP). Les différences sont :

– Une annonce reçue en I-BGP n'est pas réanoncée en I-BGP puisque le graphe des routeurs est plein, si un routeur a fait une annonce, tous les routeurs l'ont reçue.

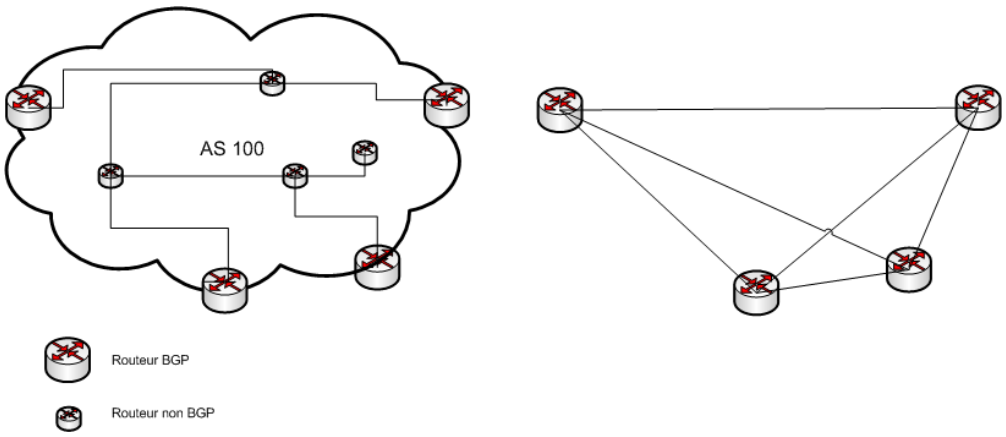

FIG. 2.4 – *Modèle BGP intra-AS*

- L'attribut LOCAL_PREF n'est pas annoncé en E-BGP, c'est à dire qu'il ne transmet pas l'information à un autre AS. Il n'est pas réanoncé s'il est reçu en E-BGP (ce qui ne devrait pas arriver) et n'est pas modifié s'il est reçu en I-BGP.
- Les annonces reçues en I-BGP ne modifient ni l'AS_PATH ni le NEXT_HOP. Ces attributs ne sont modifiés que lorsque l'annonce est reçue en E-BGP.
- Le MED est réannoncé en I-BGP lorsqu'il a été reçu en I-BGP ; il ne doit pas sortir de l'AS.

Il est à noter que seul le premier routeur de l'AS qui reçoit une annonce est autorisé à modifier le NEXT_HOP, le LOCAL_PREF, le MED et l'AS_PATH. D'autre part, il n'y a pas de filtrage pour une annonce en I-BGP, c'est à dire qu'un routeur ne peut pas décider d'éliminer une annonce ou de cacher une annonce reçue d'un autre routeur du même AS, et ce pour garder une cohérence au sein de l'AS vis-à-vis des autres AS.

Connexion deux à deux des routeurs BGP d'un même AS

Le choix du graphe plein des routeurs interne à un AS a été fait pour que chaque routeur ait la totalité de l'information et donc que l'AS offre une vue consistante aux autres AS. Nous verrons plus loin que l'augmentation du nombre de routeurs a entraîné la nécessité d'utiliser d'autres techniques pour ne pas surcharger les routeurs.

Informations disponibles pour sélectionner les routes

BGP utilise des métriques contenues dans les annonces de routes, mais se sert aussi d'informations obtenues par d'autres moyen tel que par l'IGP, le protocole de routage interne à l'AS qui lui s'occupe de router les paquets au niveau

des liens physiques. Nous introduisons donc immédiatement l'IGP_COST qui sera utile lors de la sélection de routes qui n'est autre que la somme des coûts IGP (des distances IGP) pour atteindre le routeur BGP de bordure qui a fait cette annonce.

2.3 BGP

Nous avons proposé dans la section précédente un modèle simple de BGP où nous avons étudié l'annonceur BGP en tant que AS ou élément d'un AS et la mise à jour des informations de routage. Mais ce modèle cache certains éléments de BGP, d'ou la nécessité de présenter plus en profondeur ses mécanismes.

2.3.1 Composition des annonceurs BGP

Comme nous l'avons vu, un routeur BGP est composé de tables pour enregistrer les routes apprises, d'une méthode pour filtrer les annonces et d'une méthode pour sélectionner les meilleurs chemins. Nous allons étudier plus en détail la composition de ces tables et donner une vue plus complète des politiques.

Les tables de BGP

Le routeur BGP est composé en fait de trois tables appelées des RIB (Routing Information Base) :

Adj-RIBs-In contient les informations de routages apprises par les annonceurs BGP voisins

Loc-RIB contient les informations de routage que l'annonceur BGP a sélectionnées suivant des politiques locales parmi les informations contenues dans Adj-RIBs-In

Adj-RIBs-Out contient les informations de routage qui peuvent être communiquées à un autre annonceur BGP particulier.

Il est important de rappeler qu'il ne peut pas y avoir deux routes apprises par le même routeur à partir de la même interface dans ces tables, la deuxième annonce écrase automatiquement la première.

La table Adj-RIBs-In contient donc les routes annoncées par les voisins, certaines de ces routes seront considérées comme meilleures pour une destination donnée, ces routes seront enregistrées dans Loc-RIB. C'est la table qui sert au routage des annonces BGP d'un point de vue logique c'est à dire entre routeurs BGP qui ont une session BGP ouverte. Elle ne sert pas au routage des paquets au niveau physique, c'est pourquoi les informations de routage BGP doivent être transmises au protocole de routage locale (celle d'IGP). Inversement c'est la table de routage local qui va donner des informations précieuses à

BGP telles que les coûts IGP. La dernière table, Adj-RIBs-Out, est un sous ensemble de Loc-RIB qui contient les routes à réanoncer, celles-ci étant forcément des meilleurs chemins.

Politique de filtrage et sélection des meilleurs routes

Dans BGP, il y a trois points de filtrage. Le premier permet de sélectionner les routes reçues. Il n'est pas question ici de définir la meilleure route, mais seulement de modifier la valeur du LOCAL_PREF ou de supprimer purement et simplement une annonce ; les routes ainsi filtrées seront enregistrées dans Adj-RIBs-In. Par exemple un AS qui n'a absolument pas confiance en un autre AS peut interdire les annonces des routes et donc ainsi empêcher que des paquets circulent par cet AS. Ce filtrage n'offre pas une totale liberté, par exemple nous ne pouvons pas interdire aux paquets d'un AS x de transiter par un AS y et autoriser ceux des autres AS à passer par y.

La deuxième étape est un filtrage des routes contenues dans Adj-RIBs-In pour sélectionner les meilleures et les enregistrer dans Loc-Rib. Nous avons vu ce processus précédemment dans "Processus de sélection", section 2.2.2.

La dernière étape consiste à filtrer les routes qui seront réanoncées, puis enregistrer ces routes dans Loc-RIB-out. Attention, rappelons qu'une route annoncée vers un AS implique l'obligation d'accepter le trafic de ce réseau. Par exemple, si un AS ne veut pas servir d'AS de transit, mais veut pouvoir router ses propres paquets, il peut accepter les annonces de ses voisins et en interdire la rediffusion.

Le schéma 2.5, inspiré de [Sac01] résume l'utilisation des tables et du filtrage. Les flèches à double sens soulignent l'interaction entre IGP et BGP.

Il est important de se souvenir que le protocole BGP ne définit pas les différents filtrages, chacun peut utiliser des informations extérieures différentes, ou tout simplement chacun peut avoir des politiques extrêmement différentes.

2.3.2 Communication entre annonceurs BGP

Tout annonceur BGP qui désire communiquer avec un autre, doit ouvrir une session BGP avec celui-ci. BGP s'appuie sur TCP, ce pour ne pas avoir à s'occuper dans le protocole des problèmes relevant de la transmission. Ensuite les annonceurs BGP communiquent à l'aide de quatre types de messages :

OPEN est le premier message envoyé, il permet d'informer son voisin de la version BGP utilisée, de son numéro d'AS, d'un numéro permettant d'identifier ce processus BGP et négocier le temps entre deux KEEPALIVE. En retour le voisin BGP envoie un KEEPALIVE si il accepte la connexion et un NOTIFICATION si il la refuse.

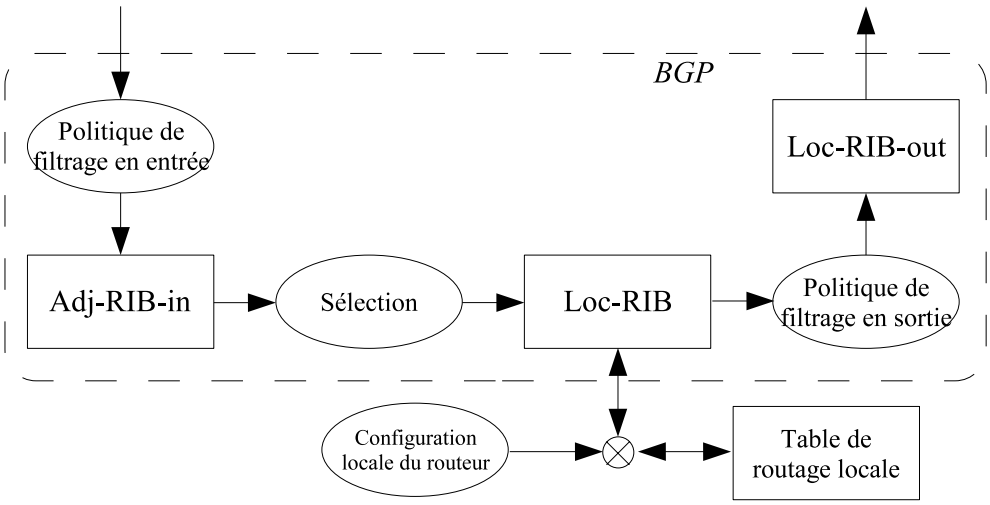

FIG. 2.5 – *Tables et filtrages*

KEEPALIVE est un message envoyé régulièrement (généralement toutes les 30 secondes) il permet d'indiquer au voisin qu'il est toujours vivant. Le compteur associé au KEEPALIVE est réinitialisé à la réception d'un KEEPALIVE ou d'un UPDATE. Si le temps du compteur s'est écoulé, l'annonceur BGP envoie un message NOTIFICATION et ferme la session.

NOTIFICATION est envoyé en cas d'incident dans BGP, comme la fin du minuteur entre deux KEEPALIVE, un message erroné... Ce message a pour action de fermer la session BGP et TCP entre l'émetteur et le récepteur du message et d'indiquer un code d'erreur. Lors de la réception d'un tel message, toutes les routes apprises par cet annonceur sont supprimées.

UPDATE est le message principal du protocole BGP. Il permet d'échanger les informations de routage entre voisins. Un message peut contenir les routes à éliminer et une nouvelle route avec ses attributs. Ce message est envoyé uniquement si l'annonceur BGP trouve un nouveau meilleur chemin.

2.4 Evolutions du protocole

Le protocole BGP a déjà atteint ses limites, il a été nécessaire de le faire évoluer. Par exemple : entièrement connecter deux à deux tous les routeurs d'un AS a pour effet d'ouvrir une multitude de sessions BGP et de faire accroître significativement les tables de routages. Il a donc été proposé deux évolutions : les confédérations et les réflecteurs de route.

Les confédérations

Les confédérations[PT01] fonctionnent comme les AS, c'est à dire qu'à l'intérieure d'un AS, nous partageons les routeurs en sub-AS ; à l'intérieur d'un sub-AS, les routeurs sont entièrement connectés deux à deux et les sub-AS sont connectés (pas forcément en clique) pour garder la connectivité de l'AS. La différence entre la notion de sub-AS et d'AS est que entre les sub-AS tous les attributs de route comme le MED, le LOCAL_PREF,... sont transmis. D'autre part c'est le premier routeur du premier sub-AS qui reçoit une annonce, qui peut modifier les paramètres de la route. D'un point de vue extérieur, un AS composé de sub-AS apparaît toujours comme un unique AS.

Les réflecteurs de routes

Les réflecteurs de route [TB00] permettent de diviser un AS en clusters (l'équivalent des sub-AS dans les confédérations) chacun contenant au moins un réflecteur de route (RR). Les clients (les autres routeurs du cluster) sont connectés uniquement au réflecteur de route. Les RR sont connectés entre eux. Cela permet de réduire significativement le nombre de sessions BGP[Sac01] : si il y a N routeurs dans un AS, nous aurons $N(N-1)/2$ sessions BGP sans RR et entre $N - R + R(R-1)/2$ et $NR - R(R+1)/2$ sessions BGP si nous avons R réflecteurs de route. Il est intéressant de noter que le minimum est atteint lorsqu'il n'y a que deux réflecteurs de routes.

2.5 Conclusion

Nous avons donné dans cette partie une vue d'ensemble de BGP. C'est un protocole simple qui permet de donner une certaine indépendance aux différents systèmes autonomes, et permet de contrôler le trafic. Mais les politiques n'offrent pas encore assez de souplesse. Un exemple est donné par C. Huitema[Hui00] dans la figure 2.6 ci-dessous. Le problème est qu'il est possible d'utiliser, pour transmettre des paquets, les trois AS de transit (1,2 et 3) de l'AS 4 vers les clients de l'AS 0 à l'aide des politiques, mais tous les paquets des clients de l'AS 0 sont obligés d'utiliser un même AS de transit (l'AS 2 dans notre exemple) pour transmettre des paquets vers l'AS 4. Un AS est appelé AS de transit lorsqu'il ne fait que transmettre les paquets d'un AS à un autre.

D'autre part il serait intéressant d'étudier les confédérations pour savoir quel est le meilleur découpage des routeurs d'un AS en fonction de la réduction de sessions BGP et de la solidité du réseau que nous voulons obtenir.

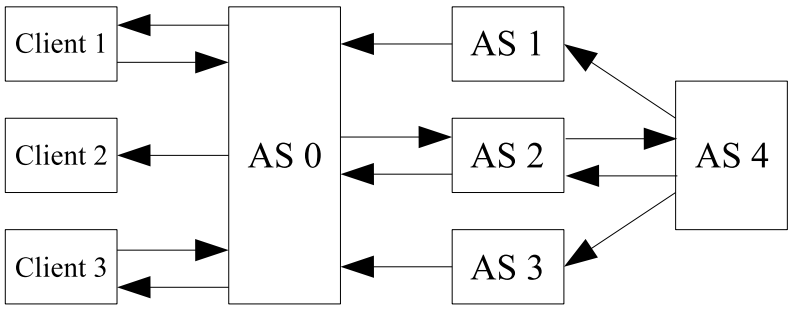

FIG. 2.6 – *Limitation du filtrage*

Chapitre 3

Oscillations interne à l'AS

BGP souffre d'une certaine instabilité entre plusieurs routes : un des premiers problèmes mis en valeur est une oscillation permanente à l'intérieur même d'un AS face à l'utilisation de réflecteurs de route ou de confédérations et de l'utilisation du MED. Avant d'étudier ce problème nous allons donner un exemple d'oscillation.

3.1 Exemple d'oscillation

Cet exemple est directement issu du RFC 3345[DM02]. Nous allons donner ici l'état des routeurs pas à pas pour joindre la destination 10.0.0.0/8 présente dans l'AS 100. Le réseau étudié est schématisé dans la figure 3.1. Il utilise la technique des réflexions de routes pour réduire la taille des tables et du nombre de messages. Les valeurs entre parenthèses représente le MED.

Les tableaux qui suivent représentent l'état de R_a et R_d après chaque réception d'une mise à jour. Chaque ligne des tables correspond à une route présente dans les annonceurs BGP.

- "Routeur" indique à quel routeur appartiennent les informations.
- Un plus dans le champ "Best_path" indique que BGP vient de sélectionner cette route (cette distinction n'est pas faite dans le RFC).
- Un moins dans le champ "Best_path" indique que la route à été sélectionné précédemment.
- "Annoncé par" est un champ que nous avons ajouté pour une meilleure compréhension, il correspond au routeur voisin qui a fait l'annonce de la route.
- "AS_PATH" correspond aux différents AS que traverse la route.
- Le "MED" correspond au MULTI_EXIT_DISCRIMINATOR
- Le "Coût IGP" correspond au poids IGP total de la route à l'interieur de l'AS.

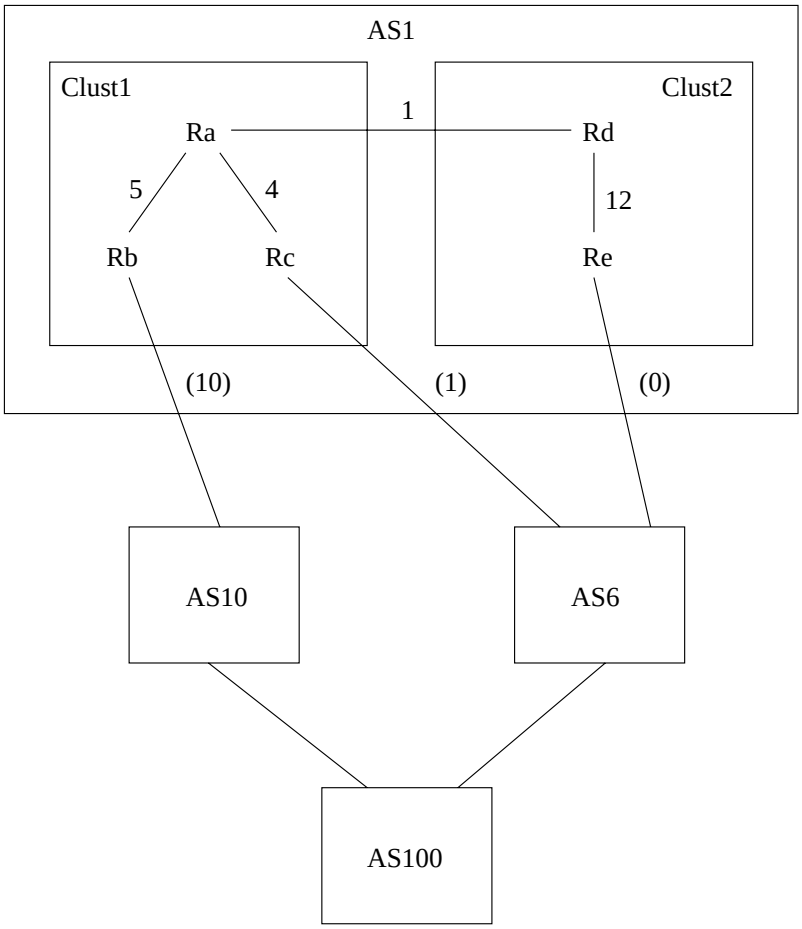

FIG. 3.1 – *Réseau utilisé pour l'étude du MED*

Etat 1 : état initial

Le tableau ci-dessous représente les routes connues par R_a et R_d à l'état initial.

Router	Best_path	Annoncé par	AS_PATH	MED	Coût IGP
R_a	-	R_b	10 100	10	5
R_d	-	R_e	6 100	0	12

Etat 2 : R_c informe R_a de son meilleur chemin

Router	Best_path	Annoncé par	AS_PATH	MED	Coût IGP
R_a	+	R_c	6 100	1	4
	-	R_b	10 100	10	5
R_d	-	R_e	6 100	0	12

R_a découvre que le chemin appris par R_c est le meilleur car la métrique IGP est plus faible donc meilleure, il en informe immédiatement R_d par un message UPDATE.

Etat 3 : R_d reçoit l'annonce de R_a pour le chemin (6 100)

Router	Best_path	Annoncé par	AS_PATH	MED	Coût IGP
R_a	-	R_c	6 100	1	4
		R_b	10 100	10	5
R_d	-	R_e	6 100	0	12
		R_a	6 100	1	5

R_d a deux chemins passant par le même AS suivant, il applique donc le MED pour sélectionner le meilleur ; il choisit donc la route apprise par R_e car elle a le plus petit MED ; il envoie donc un UPDATE pour informer R_a de son choix.

Etat 4 : R_d informe R_a de son chemins

Router	Best_path	Annoncé par	AS_PATH	MED	Coût IGP
R_a	-	R_c	6 100	1	4
	+	R_b	10 100	10	5
		R_d	6 100	0	13
R_d	-	R_e	6 100	0	12
		R_a	6 100	1	5

R_a reçoit l'UPDATE de R_d. Il a deux chemins qui ont le même NEXT_HOP, il sélectionne celui qui a le plus petit MED, c'est à dire la route apprise par R_d, puis compare cette route à celle apprise par R_b ; il sélectionne cette deuxième car elle a un coût IGP moindre. Il en informe R_d de sa décision.

Etat 5 : R_a envoie à R_d un UPDATE pour la route par R_b

Router	Best_path	Annoncé par	AS_PATH	MED	Coût IGP
R_a		R_c	6 100	1	4
	-	R_b	10 100	10	5
		R_d	6 100	0	13
R_d	-	R_e	6 100	0	12
	+	R_a	10 100	10	5

L'entrée dans R_d correspondant aux annonces faites par R_a est remplacée par la nouvelle annonce. R_d sélectionne l'annonce de R_a à cause du plus faible coût IGP. Il est donc d'accord avec l'annonce faite par R_a et demande à R_a d'oublier sa propre première annonce, la route (6 100) passant par R_e ; il envoie un UPDATE/withdraw.

Etat 6 : R_d demande à R_a d'oublier sa précédente annonce

Router	Best_path	Annoncé par	AS_PATH	MED	Coût IGP
R_a	+	R_c	6 100	1	4
	-	R_b	10 100	10	5
R_d	+	R_e	6 100	0	12
	-	R_a	10 100	10	5

R_a supprime l'entrée correspondant à R_d et sélectionne la route passant par R_c. Les deux routeurs se retrouvent dans l'état 3, il y a donc une boucle qui se traduit par une oscillation infinie entre les état 3, 4 et 5.

3.2 Explication du problème

Cette oscillation vient du fait que d'une part toutes les informations nécessaires au choix ne sont pas toujours disponibles, d'autre part que le classement des routes ne peut être ordonné lexicalement, c'est à dire qu'étant donné les routes x et y, on ne peut garantir que $x < y$ ou que $y < x$. En d'autres termes, si le routeur R_a n'a connaissance que des routes X et Y (cf Fig. 3.1), X est meilleur. Si R_a n'a connaissance que des routes Y et Z, Y est meilleur. Par contre, si R_a a connaissance de X et Z (qu'il connaisse ou non Y, Z est meilleur. Nous avons donc $X > Y > Z > X$ (en notant $>$ pour "meilleur que").

Route	Annoncé par	AS_PATH	MED	Coût IGP
X	R_c	6 100	1	4
Y	R_b	10 100	10	5
Z	R_d	6 100	0	13

TAB. 3.1 – *Exemple de tri non déterministe dans R_a*

Pour résoudre ce problème on peut essayer de :

- changer la signification ou l'utilisation du MED,
- modifier le protocole pour détecter les oscillations,
- modifier le protocole pour que les informations nécessaires soient transmises à tous les routeurs concernés,
- modifier la topologie du réseau logique BGP.

3.3 Premières solutions

Nous allons étudier les solutions proposées dans le RFC 3345 [DM02]. Ces solutions sont les plus simples ; elles modifient l'action du MED. Certaines solutions n'ont aucun sens, le RFC ne les propose pas pour les adopter mais pour expliquer qu'elles existent. Elles consistent à pouvoir trier les routes selon un ordre lexicographique afin de choisir la meilleure. Les différentes propositions

sont :

- Ne pas accepter de MED d'un voisin, ce n'est pas une bonne solution car l'utilisation du MED permet entre autres de dénigrer les liaisons à bas débit. La figure 3.2 offre un exemple de l'utilité du MED. Mais par extension de cette solution, nous pourrions proposer d'intégrer le MED directement au calcul du LOCAL_PREF.
- Utiliser des attributs plus forts dans la décision du meilleur chemin, pour ne pas avoir à passer dans l'étape du choix suivant le MED. Cette solution est proche de la première.
- Toujours comparer le MED, même s'il provient de plusieurs AS. Cette solution retire toute la signification au MED, car chaque AS peut calculer le MED de différentes manières donc comparer des MED de différents AS n'a aucun sens.
- Revenir à une solution avec un graphe des annonceurs BGP plein. Cette solution n'est pas possible du fait du grand nombre d'annonceurs BGP au sein d'un même AS.
- Mettre un coût IGP plus grand entre les clusters (ou inter Sub-AS dans le cas des confédérations des AS) qu'à l'intérieur. Cette solution à un sens car nous pourrions considérer qu'un sous-AS regroupe des routeurs BGP physiquement proches et que les sous-AS sont géographiquement éloignés (par exemple un sous-AS dans chaque pays), mais cette solution n'est valable que dans certains cas bien précis, elle ne résoud pas entièrement le problème.
- Dans le cas des réflecteurs de routes, relier tous les clients. Dans le cas des confédérations relier les routeurs de bord (les routeurs qui sont reliés directement à un autre AS ou sub-AS) à chaque niveau. Cette solution est sans doute la moins contraignante mais elle n'est pas forcément la meilleure, car cela ajoute un grand nombre de liens et donc accroît rapidement la taille des tables et le nombre de messages.

Une solution similaire a été proposée dans le cadre des réflecteurs de routes[1] où chaque RR envoie la totalité des routes et pas uniquement les meilleures aux autres RR. Cette solution retire l'intérêt des réflecteurs de routes, c'est à dire réduire les tables de BGP.

3.4 Utilité et limitation du MED

Avant d'étudier plus en profondeur le problème du MED, il est important de comprendre pourquoi ce paramètre de route est important. La figure 3.2 présente deux schémas. Le premier est un exemple d'utilisation du MED. Le

1. Solution proposée par A. Basu et al. dans "Route oscillation in I-BGP with Route Reflection", ACM SIGCOMM 2002, cité dans [TK03]

routeur BGP R_d veut annoncer la présence d'un réseau dont il est en charge à l'AS 6. Le coût de la liaison entre R_d et R_e étant très important, il préfère que les communications se fassent via le routeur R_c, pour cela il a deux solutions :

- soit il interdit l'annonce de cette route par R_e (politique d'annonce interne à l'AS) mais dans ce cas, il n'y a pas de lien de redondance,
- soit l'annonce se fait par R_c et R_e mais la liaison entre R_e et R_b a un MED beaucoup plus important que sur la liaison R_a R_c, ce qui aura pour effet de forcer le trafic à circuler via R_c.

Le MED a quelques limitations : dans le deuxième schéma, le MED n'intervient plus puisque les AS ne sont pas multiconnectés, or il serait agréable de pouvoir limiter le trafic via R_e. Cela pourrait être possible en ajoutant une métrique en paramètre d'un chemin, qui serait incrémentée à chaque traversée d'AS, mais cela poserait des problèmes de confiance, en d'autres termes un AS est dépendant du fait que tous les AS aient un calcul similaire des incrémentations.

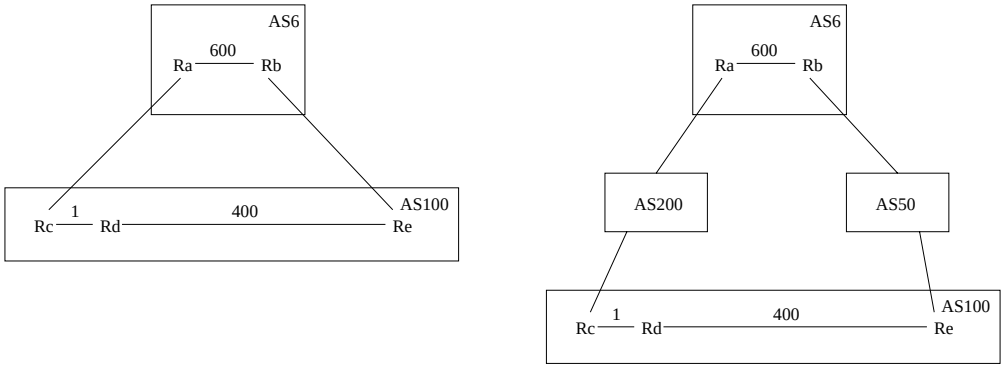

FIG. 3.2 – *Utilité et limitation du MED*

3.5 Solutions plus approfondies

Nous allons étudier et proposer dans cette section des solutions plus complexes qui permettent de conserver intégralement la signification du MED. Nous verrons trois types de solutions :

- permettre à certaines annonces d'avoir une portée plus grande
- détecter les oscillations dynamiquement
- éviter les oscillations avant même qu'elles ne se déclarent.

3.5.1 Modification du protocole pour que les annonces aient une portée plus grande

Nous proposons la solution suivante : ajouter un paramètre aux routes annoncées qui proviennent d'un routeur appartenant à un AS multiconnecté avec l'AS qui reçoit l'annonce. En d'autres termes si nous avons deux AS connectés avec plusieurs liens, les annonces passant par un de ces liens verront ce paramètre activé. Ce pourrait être juste un bit indiquant si oui ou non la valeur du MED est nécessaire aux choix du meilleur chemin. Un routeur interne à l'AS qui reçoit une annonce avec cet indicateur activé réémet l'annonce même si ce n'est pas son meilleur chemin en indiquant si l'annonce a été transmise car c'est un nouveau meilleur chemin ou parce qu'il a le bit activé. Cette technique aura pour effet de propager l'information nécessaire au bon déroulement du processus de sélection à tous les routeurs de l'AS.

- Si P est le nombre de préfixes annoncé par les AS avec qui nous avons plus d'un lien physique,
- L le nombre de liens redondants entrants dans l'AS,
- et N le nombre de liens BGP internes à l'AS,

alors le nombre de messages supplémentaires est inférieur à $N * P * L$ et le nombre de lignes supplémentaires dans les tables est au total de $L * N$.

La contrepartie de cette solution est qu'il faut indiquer "à la main" les routeurs susceptibles d'être concernés par le MED, ce qui entraîne une certaine fragilité (erreur humaine). La solution pourrait consister à ce que lorsqu'un routeur R_a apprend l'existence d'un nouveau voisin E-BGP appartenant à l'AS x, il diffuse l'information. Si un routeur R_b s'aperçoit qu'il est directement relié au même AS x, il enregistre qu'il faut activer l'indicateur signalant l'importance du MED et en informe en retour le routeur R_a initiateur de l'annonce. La diffusion consomme N paquets supplémentaires par nouvelle connexion BGP.

3.5.2 Détecter les oscillations et les éliminer

Pour détecter les oscillations il est nécessaire de conserver un historique, généralement gourmand en mémoire. T. Klockar propose une solution [TK03] basée sur l'utilisation d'un historique et dans le cadre de l'utilisation des réflecteurs de route. Le principe est fondé sur le fait qu'il garde les anciennes annonces dans la table de routage mais en mode passif. Lorsqu'un routeur reçoit une nouvelle route, il peut la comparer aux routes dites actives et aux anciennes dites passives mais son choix ne se fera que parmi les routes actives sinon il y aurait incohérence dans l'AS.

Cette technique lui permet de faire un meilleur choix car il a une meilleure vision des possibilités de chemins. Nous ajouterons que s'il trouve une meilleure route, c'est forcément parmi les chemins actifs (sélectionnés par les voisins) ; si

ce n'est pas le cas alors il y a eu une panne dans le réseau.

Cette stratégie semble intéressante car elle n'est pas trop gourmande en bande passante et en mémoire (seules les routes qui sont susceptibles d'osciller sont enregistrées). Mais elle n'est applicable que dans le cadre des réflecteurs de route. En utilisant l'exemple de réseau basé sur les confédérations de [DM02], on peut montrer que cette technique ne fonctionne pas dans le cas des confédérations.

3.5.3 Eviter les oscillations en modifiant la topologie

Nous proposons de modifier la topologie du réseau pour éviter les oscillations dues au MED. Le problème vient du fait que les routeurs de bord n'ont pas assez d'informations pour prendre une "bonne décision" parmi les chemins disponibles. Pour éviter que ce cas n'arrive nous proposons de connecter deux à deux les routeurs de bord qui sont connectés aux mêmes AS. Il est important de préciser que cette technique n'impose pas de modifications physiques du réseau puisque nous intervenons sur la couche logique BGP.

Les routeurs de bord connectés au même AS forment un graphe plein où chacun possède la même information, donc chacun est en mesure de comparer les routes et tous les routeurs auront le même meilleur chemin. Si une annonce provenant de l'extérieur de cette clique nécessite une comparaison utilisant le MED, c'est que l'annonce provient obligatoirement d'un autre routeur de cette clique. Comme ils ont tous le même meilleur chemin, la nouvelle route est déjà la meilleure. Ce qui entraîne que la portée du MED est limitée à la clique. Donc si le MED est la seule source d'oscillation à l'intérieur d'un AS, alors le réseau est stable.

Nous proposons une deuxième solution inspirée de la première mais qui nécessite de modifier légèrement le protocole puisque nous intégrons la notion de filtrage à l'intérieur de l'AS. Le principe consiste à ce que lorsqu'on ajoute un lien entre deux routeurs pour former une clique, on filtre les annonces pour que seules celles qui proviennent de l'AS dont la clique est en charge circulent. Nous réduisons ainsi le nombre d'entrées dans les tables de BGP et limitons l'utilisation excessive de la bande passante.

3.6 Conclusion

Nous avons présenté ici le premier cas d'oscillations découvert dans BGP. Ce problème peut être résolu par les différentes solutions proposées. Chacune d'elles a ses avantages et inconvénients : soit on consomme de la bande passante et de la mémoire, soit elles sont plus contraignantes vis à vis des administrateurs et plus sensibles aux erreurs humaines. Il est nécessaire de faire un choix suivant les besoins réels.

Le problème du MED semble être la seule source d'oscillation à l'intérieur d'un AS, il serait intéressant de le prouver mais pour cela il faut avant tout trouver un formalisme suffisant. D'autre part, la proposition faite par T. Klockar dans [TK03], basée sur les historiques, semble intéressante. Peut être pouvons nous généraliser cet algorithme à toutes les topologies de réseaux et pas seulement aux réflecteurs de route.

Chapitre 4

Oscillations entre AS

Nous avons vu précédemment des problèmes d'instabilité internes aux systèmes autonomes qui sont dus à l'utilisation du MED, mais BGP pourrait souffrir aussi d'instabilités entre les AS dues aux politiques d'annonce des routes. Ces filtrages ne sont pas définies dans le protocole BGP.

Actuellement le réseau des AS reste très hiérarchique. Entre AS il peu y avoir deux types de relations, l'une est de fournisseur à client, l'autre de pair à pair. Un client paye un fournisseur pour faire transiter le trafic, quant aux pairs ils s'échangent du trafic gratuitement car ils ont un volume de données à se transmettre à peut près équivalent. En résumé, BGP est formé de plusieurs niveaux, à chaque niveau le trafic est échangé librement. Si un AS X doit envoyer du trafic à un AS Y, ce trafic remonte à l'AS Z, le plus proche parent dans la hiérarchie de X et Y, et redescend à Y.

Le problème des oscillations entre AS n'a pas pour le moment été mis en évidence. Cela ne veut pas dire qu'il soit impossible ou inexistant. Mais si la taille de chacun des niveaux augmente significativement, ou que le modèle hiérarchique disparaît, ce problème pourrait apparaître comme une grave instabilité de BGP.

Pour étudier ce problème, nous allons exposer quelques exemples d'instabilités donnés par T. G. Griffin qui a fait un travail important sur ce problème, ensuite nous donnerons la solution actuellement utilisée dans Internet et finalement nous présenterons l'approche de T. G. Griffin pour étudier ce problème. Dans le chapitre suivant nous verrons notre modèle pour l'étude de ces oscillations.

4.1 Modèle SPVP (Stable Path Vector Protocol)

Pour étudier le problème d'instabilité, T. G. Griffin définie le SPVP : chaque noeud représente un AS, et il leur associe la liste des chemins possibles suivant une préférence décroissante (les politiques ne sont pas indiquées, seul le résultat

l'est). Tous les chemins indiqués ne sont pas forcément dans la table de routage ; par exemple si la route 230 est indiquée dans les chemins possibles, mais que 3 n'a pas pu annoncer 30, alors ce chemin n'est pas utilisé bien qu'il soit indiqué. L'AS 0 est toujours le premier annonceur d'une route, les flèches n'indiquent pas la direction des annonces mais la route choisie pour émettre des paquets vers un réseau se situant dans l'AS 0.

Ce modèle est utilisable car l'ensemble des chemins possible peut être concsidéré comme stable. En effet les filtres d'annonce permettant de sélectionner les chemins est une configuration statique modifié par un opérateur. On ne décide pas d'annoncer x plutôt que y parce que y commence à être chargé, sinon le réseau ne pourrait jamais converger vers un état stable puisque la modification d'une route peut avoir un impact important sur l'ensemble des AS.

4.2 Quelques exemples simples d'instabilités inter-AS

Pour commencer nous allons donner quelques exemples d'instabilité présentés par T. G. Griffin dans [TGG02a]. Ces exemples sont basés sur son modèle :

Le premier exemple (fig. 4.1) présente un réseau d'AS et les deux solutions possibles d'assignation des routes. Ce n'est pas à proprement parler un problème d'instabilité, mais T.G. Griffin le considère comme tel car pour lui un routage cohérent est un routage qui a une solution unique.

Le deuxième exemple (fig. 4.2) est plus intéressant : il n'existe pas de solution à ce problème. Cela se traduit dans la réalité par une oscillation constante entre trois états : les routes {210, 10, 30}, les routes {130, 30, 20} et les routes {320, 20, 10}.

D'autres solutions peuvent apparaître comme stables et devenir totalement instables si le réseau est très légèrement modifié. C'est le cas du *Bad Backup* présenté dans la figure 4.3. Il y a une solution unique (deuxième schéma) mais si le lien 40 vient à disparaître (une panne), la configuration revient à celle du *Bad Gadget* présenté dans la figure 4.2.

4.3 Explication du problème

Ce problème d'instabilité entre AS vient du fait que des politiques d'annonce peuvent ne pas être en conflit localement, mais l'être globalement. En d'autres termes un mauvais choix dans les politiques peut être indétectable par les AS, mais créer des conflits entre plusieurs AS.

Ces instabilités sont difficiles à résoudre, tout d'abord parce qu'il n'y a pas ou peu de recommandations faites sur les politiques d'annonce. Ensuite parce

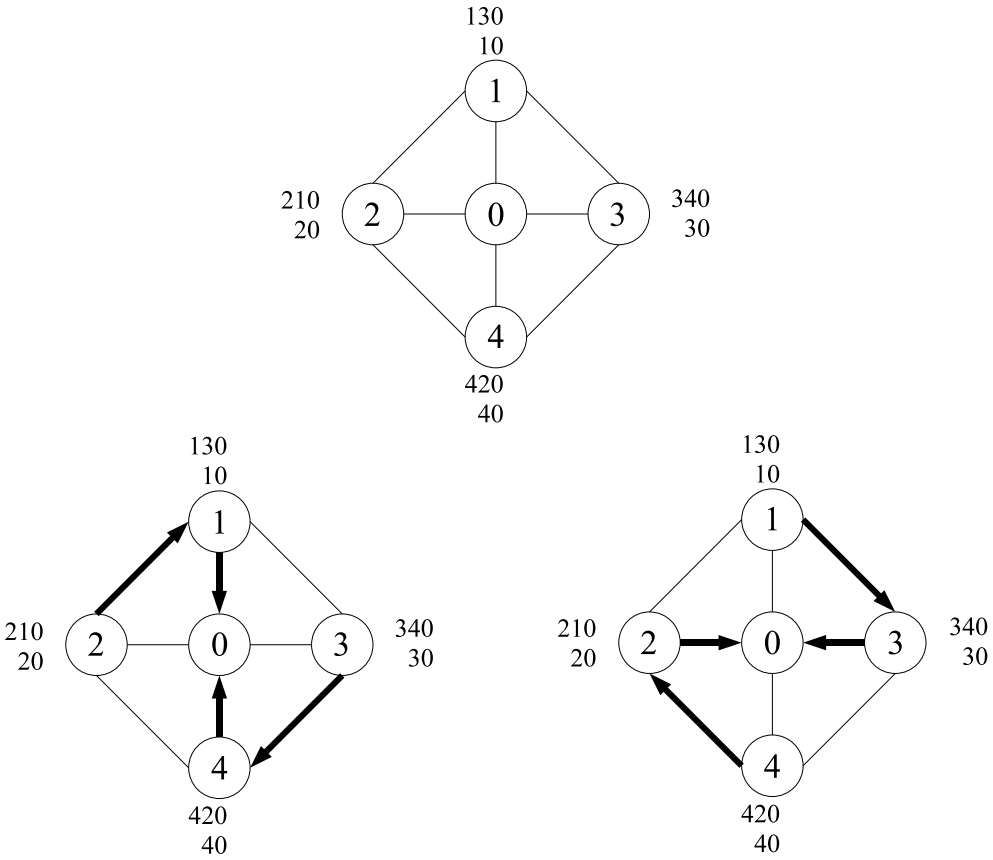

FIG. 4.1 – *Le problème "Disagree"*

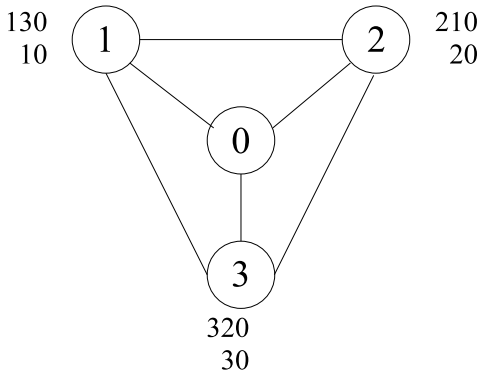

FIG. 4.2 – *Le problème "Bad Gadget"*

qu'il est difficile (du fait de la liberté sur les politiques) d'offrir une formalisation et étudier les interactions entre elles. Finalement parce que la recherche d'une mauvaise configuration dans le réseau est un problème NP-Complet. T. G. Griffin a présenté dans [TGG00] une réduction à 3-Sat de ce problème.

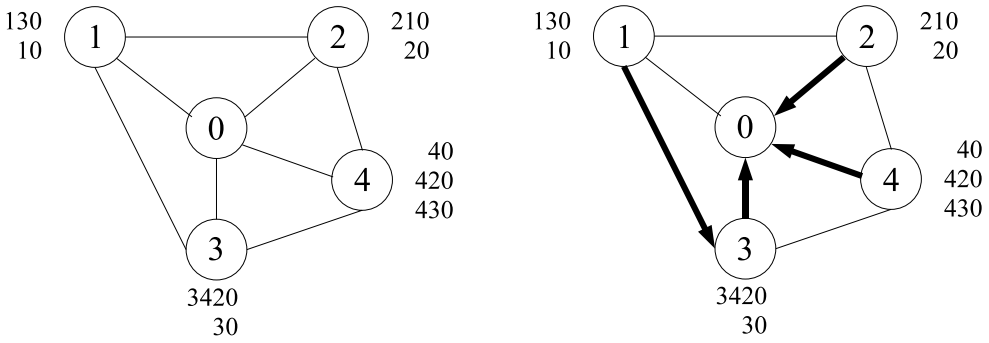

FIG. 4.3 – *Le problème "Bad Backup"*

4.4 Un système de pénalités "Route Flap Damping"

La solution utilisée actuellement dans Internet, est le "Route Flap Damping" présentée dans le RFC 2439[CV98] et résumée dans[Sac01]. Elle consiste à mettre des pénalités :

- Lorsqu'une route est retirée (parce qu'il y a un meilleur chemin) nous ajoutons un nombre de points de pénalité ($P = P + X$).
- Si la route a une pénalité supérieure à une limite ($P > L1$), la route est masquée et ne peut plus être annoncée.
- Si la pénalité retombe en dessous d'une certaine limite ($P < L2$) la route est rétablie.
- Si la route descend en dessous d'une troisième limite ($P < L3$), on oublie la pénalité.
- Si la route n'a pas oscillé pendant un certain temps on enlève des points de pénalité ($P = P/2$).

Cette solution simple n'est pas une bonne solution car elle ne fait que réduire la vitesse des oscillations et non les éliminer. D'autre part elle ralentit considérablement la convergence du réseau. Faute de mieux, c'est toujours celle utilisée dans BGP.
Il est à noter que cette solution n'a pas été proposé pour le problème du MED car elle entraînerait des inconsistances au sein d'un AS, c'est à dire qu'un AS ne serait plus vu de l'extérieur comme un routeur unique. Finalement elle pourrait introduire des boucles. Donc seul les annonces E-BGP peuvent être pénalisées.

4.5 Le problème de stabilité de chemins

Nous allons présenter dans cette section l'approche de T. G. Griffin [TGG02b] qui a fait un travail important dans l'étude des oscillations entre AS.

4.5.1 Modèle

Une solution du "Stable Paths Problem" (SPP), est l'assignation d'un chemin permis à tous les noeuds tel que :

- L'assignation d'un chemin au noeud u est soit nulle, soit de la forme uwP et il existe un noeud voisin w tel que le chemin assigné à w soit wP.
- A chaque noeud est associé le meilleur chemin parmi ceux qui sont autorisés, dans le respect de la condition précédente.

4.5.2 Graphe orienté de conflit

L'auteur propose deux approches similaires. Dans [TGG02b] il propose une méthode pour calculer les solutions au SPP. S'il y a zéro ou plusieurs solutions, il considère le réseau comme instable. Sa deuxième approche présentée dans [TGG00] est de calculer un graphe de conflit des assignements de chemins.

Construction du graphe de conflit

- Chaque noeud du graphe de conflit représente un chemin possible dans le réseau.
- Soit Q un chemin admis au noeud v et P un chemin admis au noeud u ayant pour premier noeud traversé v ($P = (u,v)P[v,0]$).

Ils existent deux types d'arcs entre les différents chemins :

- des arcs de transmission (noté en pointillé)
- des arcs de conflit (noté en trait plein)

Il y a un arc de transmission de vP vers $(u,v)P$ si u et v sont voisins. En d'autre terme si au noeud u on a la route 130 et au noeud v la route 30, alors on met un arc de transmission de 30 vers 130.
Il y a un arc de conflit de Q vers P si le noeud v peut augmenter le rang de son meilleur chemin en abandonnant P[v,0] au profit de Q ce qui a pour effet de forcer u a abandonner P. En d'autres termes, il y a conflit si un chemin au noeud v peut forcer l'abandon d'un chemin au noeud voisin u.

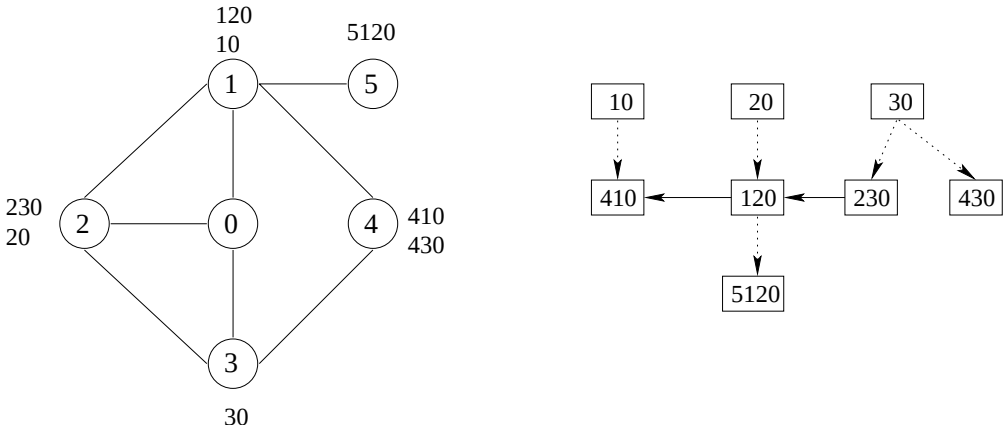

FIG. 4.4 – *Exemple de réseau avec son graphe de conflit orienté*

La propriété principale de ce graphe est que s'il n'y a pas de cycle dans le graphe de conflit, alors le réseau est stable. La figure 4.4 présente un réseau et son graphe de conflit, celui-ci ne contient pas de cycle, le réseau est donc stable.

Cette propriété n'est pas une condition nécessaire, l'inverse n'est pas vrai. La figure 4.6 présente un tel cas : le graphe de conflit contient un cycle alors que le réseau est stable, en sélectionnant les chemins 40, 140, 240 et 340. Cette figure est une extension de la figure 4.5 qui représente le cas d'un réseau qui oscille.

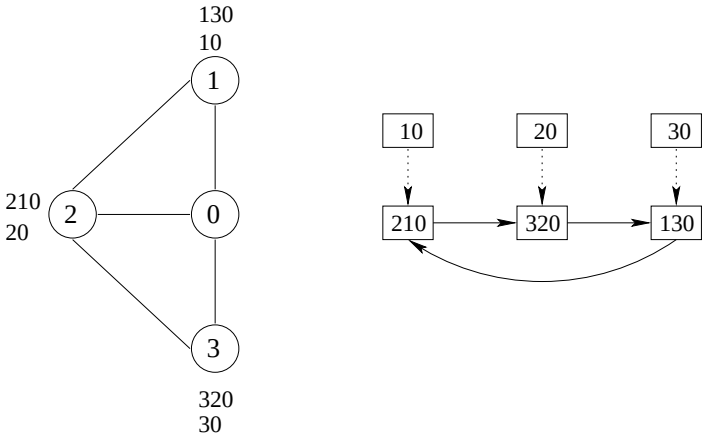

FIG. 4.5 – *Exemple de graphe de conflit avec cycle*

4.5.3 Limitations et conclusion

Nous avons présenté cette solution car elle est la plus aboutie. Mais elle présente néanmoins quelques défauts :

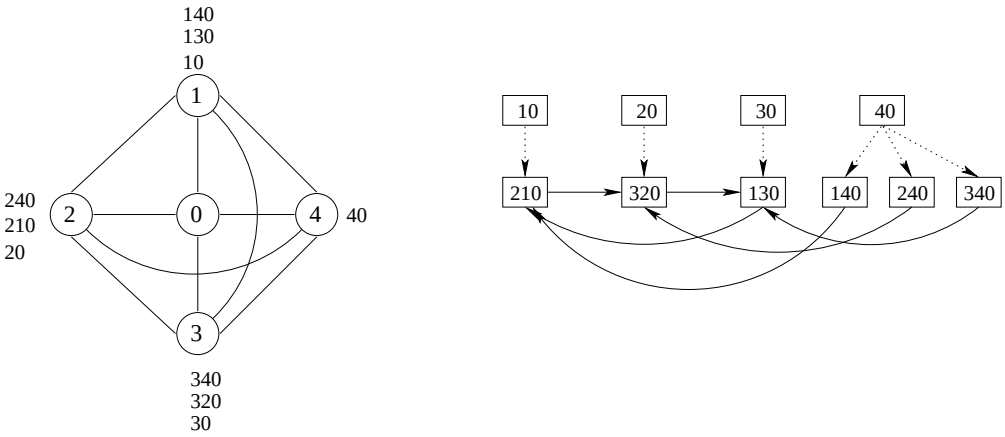

FIG. 4.6 – *Exemple d'un réseau stable et son graphe de conflit avec cycle*

- Cette méthode offre des faux positifs, c'est à dire que certains réseaux sont considérés comme instables et non robustes alors qu'ils le sont.
- le nombre de chemins possibles peut être très important et le graphe de conflit contient autant de noeuds que de chemins donc il peut devenir rapidement incompréhensible et ingérable.
- Cette approche n'offre pas de méthode donnant les solutions possibles des assignations de chemins.

Dans le chapitre suivant nous proposerons un modèle pour palier à ces limitations.

Chapitre 5

Graphe des sous-états

Nous allons dans ce chapitre présenter notre modèle : "le graphe de sous-états" (GSE). Cette approche permet, en se basant sur la modélisation SPVP de T. G. Griffin, d'offrir une nouveau graphe de conflit plus intuitif et surtout qui possède plus de propriétés. Nous verrons deux types de graphes, l'un utilise la notion de sous-états minimaux, l'autre de sous états généralisés.

5.1 Graphe des sous-états minimaux

Ce graphe de conflits dit graphe des sous-états minimaux n'est pas basé sur les conflits entre chemins mais entre des états du réseau. Par état nous entendons un ensemble de chemin cohérent entre eux.

5.1.1 Définitions

Nous allons commencé par une série de définitions utiles à la construction du GSE.

Définition 1 (Sous-état minimal) *Un sous-état minimal du réseau est un ensemble E de chemins inclus dans l'ensemble de chemins possibles du réseau U tel que $\forall x,y \in E$, on ait $x \longrightarrow y$ et $y \longrightarrow x$ et $\nexists z \in U \setminus E$, tel que $z \longleftrightarrow x$. avec $\longrightarrow$ indiquant l'obligation de choisir le chemin.*

La définition 1 revient à dire que deux chemins x et y compatibles avec les politiques d'annonce, sont dans le même sous-état du réseau si le fait de choisir le chemin x entraîne l'obligation de choisir y et vis versa.

Définition 2 (Sous-états disjoints) *Deux ensembles de sous-états sont disjoints si l'ensemble des noeuds traversé par les chemins de chacun des deux ensembles sont disjoints.*

Définition 3 (Sous-états incompatibles) *Deux sous-états E_1 et E_2 sont dit incompatibles ssi $\exists x \in E_1$ et $y \in E_2$ tel que x et y sont deux chemins possibles appartenant au même noeud du réseau.*

La définition 3 revient à dire que deux chemins sont dit incompatibles si ils ne peuvent être choisis simultanément car ils correspondent à deux chemins possibles au même noeud du réseau. Si deux chemins sont incompatibles, la définition 1 implique que les deux sous-états sont incompatibles entre eux.

Définition 4 (Sous-états amis) *Deux sous-états E_1 et E_2 sont dit amis, ssi $\forall x \in E_1$ et $\forall y \in E_2$, x et y peuvent être choisi simultanément.*

La définition 4 revient à dire que deux sous-états E_1 et E_2 sont amis si le fait de choisir les chemins de E_1 ne perturbe pas la sélection des chemins de E_2. Par exemple, E_1 et E_2 sont amis si l'ensemble des chemins de E_1 passe par des noeuds différents de l'ensemble des noeuds utilisés par les chemins de E_2.

Définition 5 (Sous-états en conflits) *Deux sous-états E_1 et E_2 sont dit en conflits ssi $\exists x \in E_1$ tel que x peut interdire le choix d'un chemin de $y \in E_2$.*

Propriété 1 *Deux sous-états minimaux E_1 et E_2 qui ne sont ni amis ni incompatibles sont forcément en conflits.*

Démonstration *Démonstration par l'absurde :*
Soient deux sous-états E_1 et E_2. Si ils ne sont pas amis, alors il existe un chemin x appartenant à E_1 et y appartenant à E_2 tel que x et y aient un noeud en commun.
Si ces deux sous-états ne sont pas en conflit alors qu'il ont un noeud en commun, cela veut dire que x et y sont sélectionnable simultanément. Or un tel cas rentre dans la définition des sous-états amis. $\diamondsuit$

5.1.2 Construction du graphe des sous-états

Nous allons donner dans cette section la méthode de construction du sous-graphe des états. La figure 5.1 représente le réseau sur lequel nous allons construire notre graphe de conflit entre sous-états.

Ecriture des sous-états minimaux

En utilisant la définition 1, nous construisons les quatre sous-états correspondant au réseau représenté à la figure 5.1. Ces sous-états sont donné dans la figure 5.2.

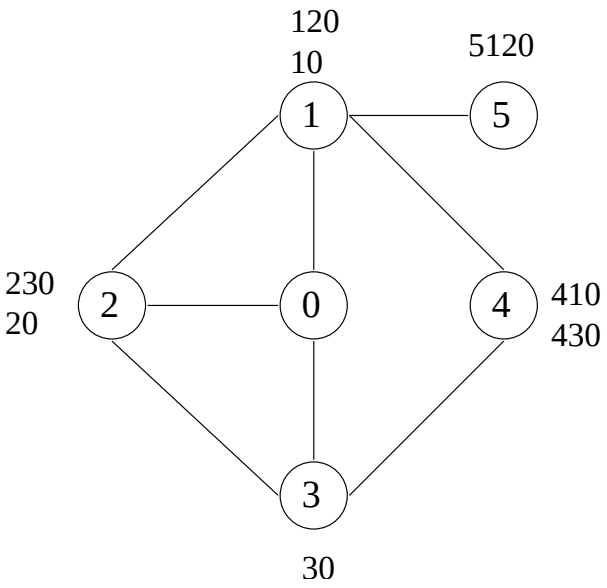

FIG. 5.1 – *Exemple de réseau*

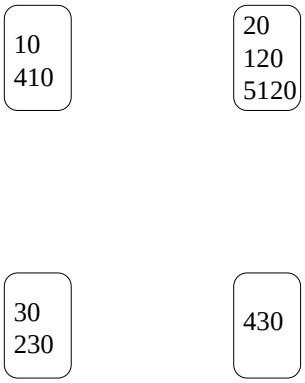

FIG. 5.2 – *Les sous-états minimaux du réseau*

Par exemple les chemins 20, 120 et 5120 ont été regroupés au sein du même sous-état car d'une part le fait d'avoir le chemin 5120 implique que 120 et donc 20 aient été sélectionnés et d'autre part si 20 est sélectionné, 120 l'est aussi puisque c'est le meilleur chemin que puisse avoir le nœud 1. Si on a 120, on a obligatoirement 5120 car le noeud 5 n'a pas d'autres alternatives.

Nous n'avons pas mis le chemin 430 dans le sous-état {30, 230} car il est possible de sélectionner 30 et ne pas choisir 430 car le nœud 4 peut préférer le chemin 410.

Recherche des sous-états incompatibles

L'étape suivante consiste à indiquer les sous-états incompatibles suivant la définition 3. Par exemple les sous-états {10, 410} et {430} sont incompatibles puisque le nœud 4 ne peut choisir simultanément les chemins 410 et 430. Nous indiquons sur le graphe des sous-états cette incompatibilité (figure 5.3) par une flèche en pointié (tiret) du sous-état qui possède le chemin de préférence plus faible vers celui de préférence plus élevé.

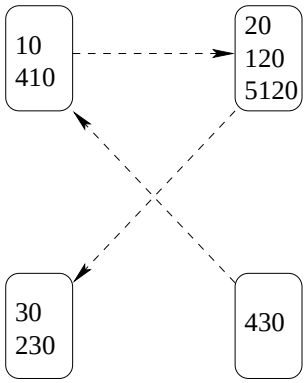

FIG. 5.3 – *Les sous-états incompatibles*

Recherche des sous-états disjoints amis

Tous sous-états disjoints sont par définitions amis. Nous indiquons donc les sous-état disjoints par une ligne en pointillé (point). Dans notre exemple, les sous-états {20, 120, 5120} et {430} utilisent respectivement les nœuds {1, 2, 5} et {3, 4} ; ils sont disjoints donc amis. Il en est de même avec {10, 410} et {30, 230}. La figure 5.4 présente le résultat de cette opération.

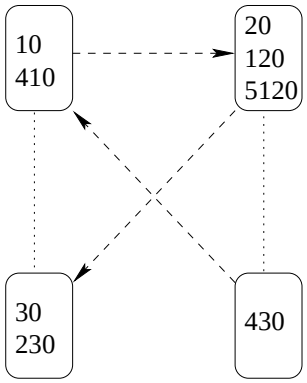

FIG. 5.4 – *Les sous-états disjoints amis*

Recherche des sous-états amis non disjoints

D'autres sous-états sont amis car le choix de l'un ne perturbe pas ceux de l'autre. C'est le cas entre {30, 230} et {430} (figure 5.5), en d'autre terme il est possible de sélectionner simultanément les chemins 30, 230 et 430.

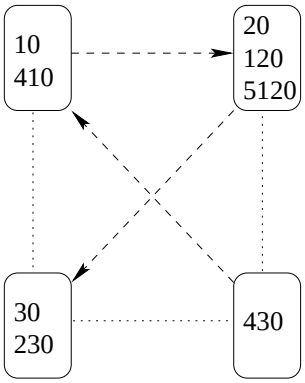

FIG. 5.5 – *Les autres sous-états amis*

Construction du sous graphe de conflit

Suivant la propriété 1, tous sous-états qui ne sont ni amis ni incompatibles sont en conflits. Nous ajoutons donc autant d'arêtes que nécessaire pour obtenir un graphe plein. Ces arêtes correspondent aux conflits. Nous orientons ensuite ces arêtes du sous-état qui peut disparaître vers celui qui en profite.

Certains états incompatibles peuvent être aussi en conflits. Cette dernière propriété étant plus forte nous remplaçons tous les états incompatibles qui sont en conflits par un conflit. Nous l'indiquons par une flèche allant du sous-état qui peut disparaître vers celui qui en profite.

Par exemple si les chemins 10 et 410 ont été choisi, si le nœud 2 sélectionne 20, alors le nœud 1 sélectionne 120 car c'est un meilleur chemin. 120 a donc masqué 410, nous indiquons cette priorité par une flèche du sous-état qui contient 410 vers le celui qui contient 120. La figure 5.6 présente le graphe des sous-états minimaux à la fin de sa construction.

Le graphe modélisé par les flèches pleines est à proprement parlé le graphe de conflit. Ce graphe peut non connexe.

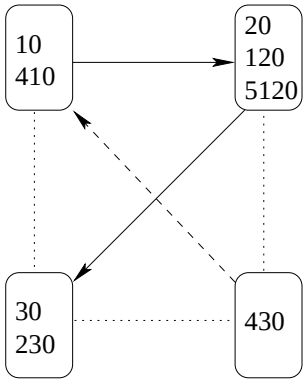

FIG. 5.6 – *Le sous graphe de conflit*

5.1.3 Propriétés

Propriétés immédiates

Propriété 2 *Soit deux sous-états minimaux E_1 et E_2. Si un chemin $x \in E_1$ est en conflit avec un chemin $y \in E_2$, alors tous les chemins de E_1 sont en conflit avec y.*

Démonstration *Si le chemin x est en conflit avec y, cela veut dire que x ne peut etre choisi si le chemin y a été sélectionné.*

Or d'après la définition 1 d'un sous-état minimal, si $x' \in E_1$ on a $x' \longrightarrow x$. Cela entraîne que si x ne peut être choisi x' ne le peut pas non plus, donc x' est aussi en conflit avec y. $\diamondsuit$

Stabilité et solidité

Théorème 1 (CS à la stabilité) *Soit G le graphe des sous-états, $G' \subset G$ le sous graphe des états en conflit. Si G' ne contient pas de cycle, le réseau est stable.*

Démonstration *Idée de démonstration :*
Le graphe de conflit indique l'évolution du choix des chemins. Si le graphe des conflits G' ne contient pas de cycle, alors le sous-état sélectionné tendra vers un des puits du réseau. Ce sous état donne un ensemble de chemin compatible.
$\diamondsuit$

Théorème 2 *Soit G le graphe des états, $G' \subset G$ le sous graphe des états en conflit, si G' ne contient pas de cycle, la suppression de chemins dans G n'introduit pas de cycle dans G'.*

Démonstration *Prenons les cas un par un :*
Soit E_1 et E_2 deux sous-états incompatibles.

- *Si E_1 et E_2 sont amis, cela veut dire que pour tous chemins x appartenant à E_1 et y appartenant à E_2, x et y sont amis, si on supprime un chemin à l'un des deux sous-états, tous les chemins de E_1 et E_2 seront toujours amis. Donc les sous-états E_1 et E_2 seront toujours amis.*
- *Si E_1 et E_2 sont incompatibles, et que l'on supprime un chemin x de E_1 ; si x était en incompatibles avec un chemin y de E_2 et qu'il n'y avait pas d'autres incompatibilités les sous-états deviennent amis. Si x n'était pas en conflit ou si il y avait plusieurs conflits, E_1 et E_2 reste en conflit.*

Il n'est donc pas possible d'ajouter des conflits en supprimant des chemins. Donc si le garphe de conflit initial ne contenait pas de cycle, le fait de supprimer un chemin ne peut créer un cycle.
♢

Il est a noté que si E_1 et E_2 sont deux sous-états en conflit, la suppression d'un chemin peut laisser E_1 et E_2 en l'état ou le transformer en sous-états incompatibles ou amis.

Le théorème 2 introduit la sûreté d'un réseau. Effectivement un réseau stable peut devenir instable en présence de panes. L'absence de cycle dans le graphe entraîne une grande résistance au panne pour ce problèmes des oscillatiions.

Cas des réseaux stables dont le graphe des conflits possède un cycle

La figure 5.7 présente le cas d'un réseau possédant un cycle ainsi que son graphe des sous-états minimaux ou nous avons indiqué que les conflits. Bien que le graphe des sous-états contienne un cycle, ce réseau est stable si il n'y a pas de pannes.

En fait tout réseau possédant un cycle est stable en l'absence de panes si il est possible de sortir du cycle. Dans notre exemple si on prend le sous-état {210}, on préférera aller au sous-état {40, 140, 240, 340} plutôt qu'au sous-état {130} car le noeud 1 préfère le chemin 140 au chemin 130.

5.1.4 Recherche des solutions

L'intérêt principal de cette modélisation est d'offrir directement les solutions possibles de l'assignation des chemins dans le réseau. La solution finale doit contenir au plus un chemin par par noeud.

Recherche des puits dans le sous-graphe de conflit

- Nous découpons le graphe des conflits G (modélisé par les flèches) en sous graphes connexes G_1,G_2,...,G_n sans casser d'arc de conflit. En d'autres

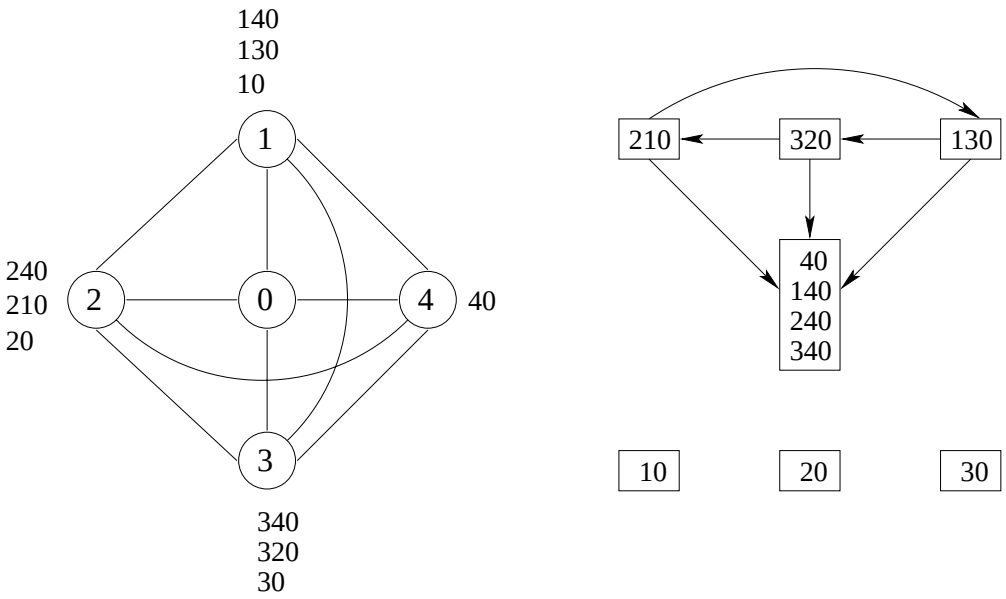

FIG. 5.7 – *Exemple d'un réseau stable et son graphe de conflit avec cycle*

termes si notre graphe de conflit contenait deux sous-graphes non connectés entre eux, nous obtiendrons deux graphe connexe. Un état qui n'a pas de conflit avec d'autres états est un graphe de conflit avec un seul noeud.

– Nous recherchons les puits dans chaque graphe G_i ; un graphe avec un seul noeud est un puit.
– L'ensemble des chemins contenus dans ces puits font parti de la solution.

La figure 5.8 donne le GSE du réseau présenté dans la figure 5.1. Il n'y qu'un seul sous graphe de conflit qui ne contient qu'un seul puit, nous l'avons indiqué en gras. A ce stade l'assignation des chemins est :

Noeud	chemin
1 :	
2 :	230
3 :	30
4 :	

Supression des sous-états superflus

Ensuite nous gardons les états amis des puits et supprimons les autres. Ces états superflus ne contiennent pas de chemin appartenant à la solution, sinon ils auraient été amis de l'un des puits. Dans la figure 5.9 nous supprimons l'état {20, 120, 5120}.

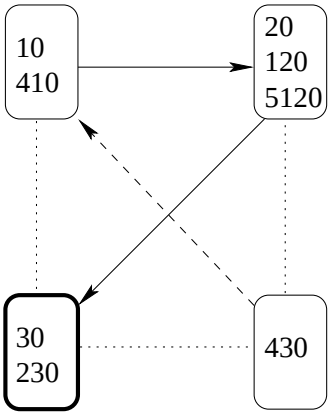

FIG. 5.8 – *Recherche des puits dans le sous-graphe de conflit*

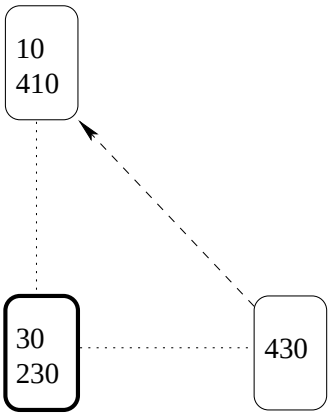

FIG. 5.9 – *Supression des sous-états superflus*

Recherche des sous-états de plus fort poids

Le reste des chemins est à chercher dans les sous-états amis, mais nous pouvons y trouver plusieurs chemins pour un même noeud. La solution consiste a rechercher les puits dans le graphe des incompatibilités. Nous utilisons les chemins pour continuer à completer le tableau des assignations. Dans notre exemple il y a puit dans le graphe des incompatibilités, noté en gras dans la figure 5.10. Notre table d'assignation devient :

Noeud	chemin
1 :	10
2 :	230
3 :	30
4 :	410

Notre table est complete, elle représente une solution à l'assignation des chemins dans notre réseau.

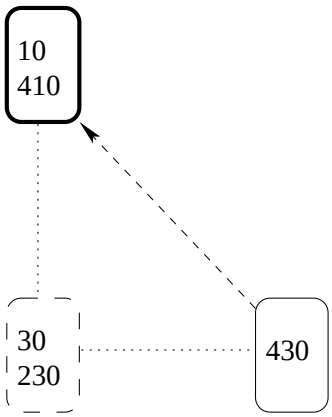

FIG. 5.10 – *Recherche des sous-états de plus fort poids*

Recherche des solutions suivantes

Si nous n'avons pas obtenu tous les chemins nécessaire, il faut supprimer les puits traités dans le graphe des incompatibilités et rechercher les nouveaux puits. Et ainsi de suite jusqu'à ce que la table des assignations de chemins soit complete. Si elle n'est pas complete alors que nous avons parcouru tout le graphe, alors un noeud ne pourra pas joindre le noeud 0 ; ce n'est pas un défaut du modèle mais une incohérence de BGP.

5.2 Graphe des sous-états généralisés

Le graphe des sous-états minimaux présente des propriétés intéressantes, mais ce graphe peut devenir très vite immense. Nous allons donc proposer une solution pour réduire significativement la taille de se graphe : le graphe des sous-états généralisés.

Le principe consiste à regrouper certains sous-états similaires. Par exemple dans la figure 5.7 il serait intéressant de regrouper {10} et {210}, {20} et {320}, {30} et {130}.

5.2.1 Les différentes réductions

La figure 5.11 présente certaines réductions possibles. x, y, z représente des sous-états, les liens sont les amis, incompatibilités et conflits vu précédemment et xy représente un seul sous état contenant les chemins de x et de y. x est le noeud que nous cherchons à intégrer. On suppose aussi que y contient un chemin obligatoire par l'un des chemins de x, par exemple pour {10} et {410}, le chemin 10 est obligatoire pour avoir 410.

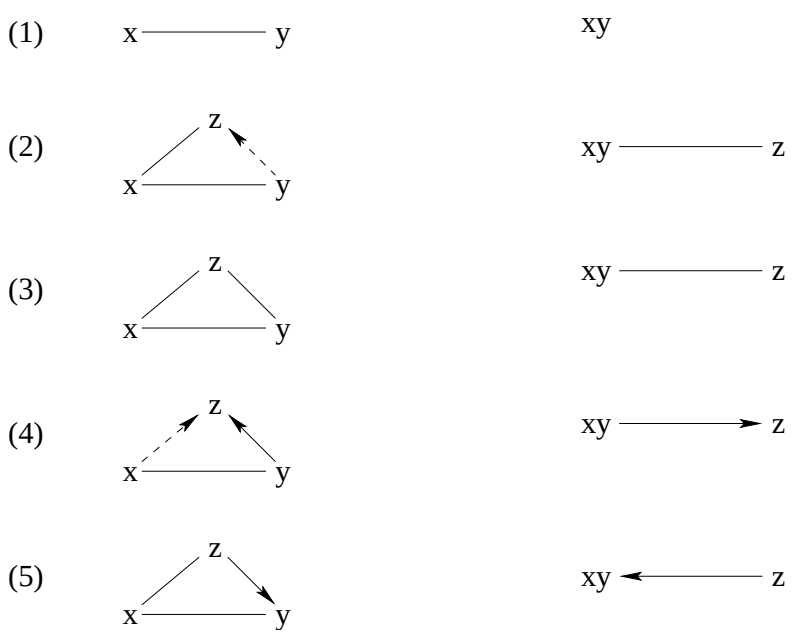

FIG. 5.11 – *Exemples de réductions*

Ci-dessous une explication de chaque cas :

Cas (1) si x est un puit donc une solution, y sera aussi selectionné donc il peuvent appartenir au même sous état.

Cas (2) si z est incompatible avec y , comme y est obligatoire pour avoir x, x est incompatible avec z, donc x peut être intégré à y sans être intégré à z.

Cas (3) est un extension du cas numéro 1.

Cas (4) est un cas similaire du cas 2. Si y et z sont en conflit, x et z lz sont aussi.

Cas (5) c'est la même cas que précédemment.

Exemples de réduction

Nous allons donner un exemple de réduction basé sur le réseau et son graphe des sous-états présentés dans la figure 5.7.

Dans la figure 5.12 nous avons agréger les états {10} et {210} car ils sont amis (cas 1), mais nous n'avons pas ajouté 10 dans les états : {20}, {320}, {30} et {130} car :

- {10} et {30} sont amis donc si {10, 210} est sélectionné, {30} le sera aussi (cas 3).
- {210} et {20} étaient incompatibles donc le chemin 10 est incompatible avec {20} (cas 3).
- {210} et {130} étaient en conflit donc le chemin 10 est en conflit avec {130} (cas 4).

– {210} et {320} étaient en conflit donc le chemin 10 est en conflit avec {320} (cas 5).

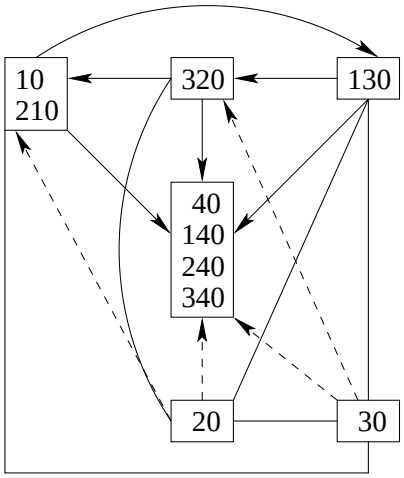

FIG. 5.12 – *Exemple d'une réduction*

5.2.2 Construction

Il est possible de créer directement le graphe des sous-états généralisés sans utiliser le GSE. Le principe de construction est exactement le même que pour le graphe des sous-états minimaux, seul change la construction des sous-états.

Construction des sous-états généralisés

Nous allons construire étape par étape le graphe de conflit généralisés pour le réseau présenté à la figure 5.7. La première étape consiste à construire le noeud du graphe : les sous-états généralisés :

- Il faut tout d'abords regrouper les chemins qui sont liés. En d'autres termes, si le chemin x est inclu dans le chemin y, ils appartiennent au même sous-état. Dans notre exemple 10 et 210 sont liés, on le met dans le même état. L'ensemble des sous-états est donné à la figure 5.13.
- Si un des chemins d'un sous-état peut être en conflit avec un autre du réseau, nous le supprimons et le mettons dans un sous-état à part.
- nous agrégeons les états E_1 et E_2 si le fait de sélectionner l'état E_1 entraine l'obligation de sélectionner l'état E_2 et vice versa. Cette agrégation est similaire à la construction des sous-états minimaux. Dans notre exemple le fait de sélectionner 40, contenu dans le sous-état {40, 140} entraîne l'obligation d'avoir les chemins 240 et 340, nous agrégeons donc ces trois états (figure 5.14).

– Nous verifions que pour chaque état et chaque chemin, on trouve aussi les sous-chemins dans cet état. Par exemple si nous avions le chemin 5230 dans un état, il faut vérifier la présence dans cet état des chemins 230 et 30.

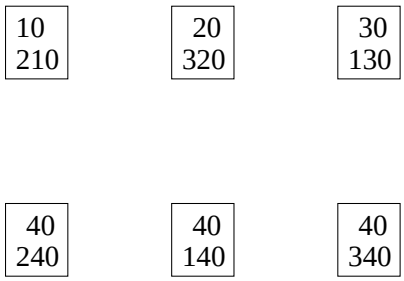

FIG. 5.13 – *Chemins liés*

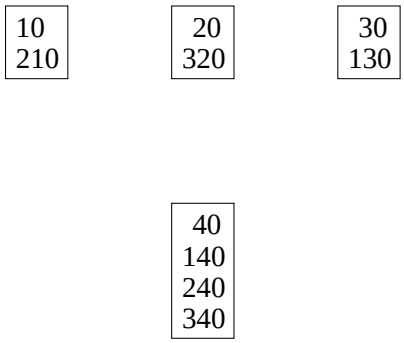

FIG. 5.14 – *Sous-états généralisés*

Sous-états amis, incompatibles et en conflits

Ensuite nous construisons le graphe des incompatibilités, des conflits et des amis de façon identique à la méthode utilisée dans les graphe de sous-états minimaux. La figure 5.15 représente le graphe finale des sous états généralisé.

Réduction

Le graphe que nous avons obtenu est minimal, il n'est pas possible de le réduire, mais ce n'est pas toujours le cas, mais il est toujours possible d'appliquer les règles de réduction sur ce graphe.

5.2.3 Propriétés

Dans ce nouveau graphe nous conservons la propriété des cycles. En effet en agrégeant des sous-états, nous n'avons ni supprimé, ni ajouté de liens de conflits.

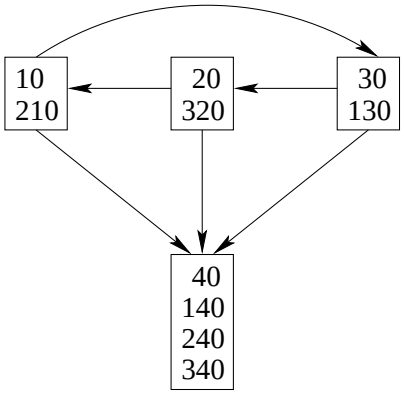

FIG. 5.15 – *Exemple de graphe des sous-états généralisés*

Par contre nous perdons la possibilité de trouver simplement les différentes solutions de l'assignation du réseau car en utilisant l'agrégation de sous-états nous avons ajouté des chemins potentiellement incompatibles.

Par exemple si nous avions x, y et y, z qui étaient nous aurions agrégé x et y, on aurait dont les deux états xy et z. Dans le graphe des sous-états minimaux, la sélection de z n'entraînait pas la sélection de x. Dans ce nouveau graphe, x et y ayant été agrégé, la sélection de z entaîne la sélection de y.

Recherche de solutions

- Pour rechercher les solutions, nous sélectionnons les puits dans le graphe de conflit ainsi que les sous-états amis. Nous obtenons un ensemble de chemin.
- Nous comparons deux à deux les chemins pour un même noeud (ie les chemins qui ont les même premier noeud, par exemple, 530 et 560) et supprimons celui qui a la préférence la moins élevée.

5.3 conclusion

Nous avons proposé deux modèles pour étudier les instabilités entre AS. Ces représentations ont l'avantage, par rapport à celle proposé par T. G. Griffin, d'être plus intuitives et surtout de proposer d'autres propriétés telles que la recherche d'une solution ou l'étude de la stabilité dans le cas de graphe contenant un cycle.

Chapitre 6

Conclusion et perspectives

Après avoir exposé les caractéristiques de BGP, nous avons vu que des oscillations infinies pouvaient apparaître à l'intérieur d'un système autonome et des instabilités entre AS. Ensuite nous avons proposé un modèle pour étudier ce dernier cas d'instabilité.

Les oscillations internes à l'AS sont liées à l'utilisation de l'attribut de route MULTI-EXIT-DISCRIMINATOR et l'utilisation d'un graphe non plein des annonceurs BGP dans un même AS. Elles viennent du fait que d'une part toutes les informations nécessaires à la sélection de la meilleure route ne sont pas toujours disponibles, d'autre part que le classement des routes n'est pas ordonné lexicalement. Nous avons aussi vu que des instabilités (dont des oscillations) pouvaient apparaître entre les AS sans l'utilisation du MED. Ce problème vient du fait que des politiques d'annonce qui localement ne sont pas en conflit, peuvent l'être globalement.

Dans le cadre des oscillations dues à l'utilisation du MED, nous proposons deux solutions :

- La première consiste à ajouter un attribut de route indiquant la portée d'une annonce à l'intérieur d'un AS. En d'autres termes si une route risque d'être sélectionnée suivant le MED, l'annonce de la route doit se propager dans la totalité de l'AS et dans ce cas un indicateur est activé. Chaque routeur qui reçoit une annonce avec cet indicateur activé, renvoie, s'il ne l'a pas déjà fait, l'annonce à tous ses voisins de l'AS. Cette solution a l'avantage de consommer le minimum de bande passante et de mémoire supplémentaire.
- La deuxième solution que nous proposons consiste à modifier la topologie logique du réseau BGP à l'intérieur d'un AS pour que deux routeurs appartenant à un même AS et qui ont un lien E-BGP vers le même AS voisin, soient directement connectés (ont une session BGP ouverte) entre eux. Cela revient à créer un graphe plein entre les routeurs concernés par le MED et donc chacun a toutes les informations nécessaires pour faire un bon choix. Cette technique a l'avantage de ne pas modifier le protocole BGP. Pour limiter la bande passante et la mémoire supplémentaire, nous

proposons une évolution de cette technique qui consiste à introduire le filtrage d'annonce à l'intérieur d'un AS tel que s'il est nécessaire d'ajouter un lien BGP entre deux routeurs, seules les annonces concernées par le MED soient émises sur ce lien.

Ces deux solutions ont l'avantage d'être moins gourmandes en bande passante et mémoire que les algorithmes basés sur des historiques, mais en contrepartie elles sont plus sensibles aux erreurs de configuration (erreur humaine).

Les perspectives immédiates relèvent des oscillations internes aux AS et des instabilités entre AS.

Pour les oscillations internes à l'AS, il faudrait approfondir l'algorithme de T. Klockar basé sur des historiques pour prendre en compte des topologies de réseaux quelconques. En effet cet algorithme est peu gourmand en nombre de messages et mémoire utilisée car il ne conserve que très peu d'informations dans son historique mais il ne s'applique qu'aux réflecteurs de routes ; il ne fonctionne pas sur des topologies comme celles utilisées dans les confédérations.

Pour les instabilités entre AS, le modèle que nous proposons offre une base à leurs études. Les perspectives immédiates interviennent directement sur le modèle :

- Dans le cadre des sous-états généralisés, il faudrait approfondir la notion de réduction admissible pour essayer de trouver un cas général.
- Toujours dans les graphes des sous-états généralisés, il faudrait approfondir la recherche de solutions, par exemple en colorant le graphe.
- Il serait intéressant de trouver un formalisme qui permet de détecter immédiatement si en présence d'un cycle nous avons une chance ou non de sortir de ce cycle.

Les perspectives à plus long terme :

- Le modèle des sous-états permet d'étudier la solidité d'un réseau, c'est à dire qu'est ce qu'il faut enlever pour que le réseau devienne instable.
- Il serait intéressant d'intégrer directement au modèle les politiques d'annonce, ceux pour étudier les conflits entre politiques et proposer des conditions sur les politiques les moins contraignantes possibles pour ne pas avoir de conflit.

Annexe A

Lexique

AS cf. Système autonome.

Coût IGP cf. IGP

EGP est le prédécesseur de BGP. C'est un protocole à vecteur de distances permettant d'échanger des informations de routage entre AS.

E-BGP nom donné au protocole BGP lorsqu'il s'agit d'échanges de messages entre AS.

I-BGP nom donné au protocole BGP lorsqu'il s'agit d'échanges de messages interne à un AS.

IGP nom générique donné aux protocoles internes à un AS permettant de router au niveau des liens physiques. Ces protocoles sont généralement basés sur une notion de distance appelée coût IGP qui peut représenter le temps mis pour aller d'un routeur à un autre ou le nombre de routeurs traversés.

MED (MULTI-EXIT-DISCRIMINATOR) est un attribut de route BGP permettant de discriminer un ou plusieurs liens lorsque deux AS sont multi-connectés

Système autonome ensemble de routeurs sous une même entité administrative.

Bibliographie

[CV98] R. Govindan C. Villamizar, R. Chandra. Bgp route flap damping. *RFC*, (2439), Novembre 1998.

[DM02] D. Walton A. Retana D. McPherson, V. Gill. Border gateway protocol (bgp) persistent route oscillation condition. *RFC*, (3345), Août 2002.

[ECRI82] Bolt Beranek Eric C. Rosen and Newman Inc. Exterior gatway protocol (egp). *RFC*, (827), Octobre 1982.

[Hui00] C. Huitema. *Routing in the Internet*. Prentice Hall, second edition, 2000.

[KL89] Y. Rekhter K. Lougheed. Border gateway protocol (bgp). *RFC*, (1105), Juin 1989.

[KL90] Y. Rekhter K. Lougheed. Border gateway protocol (bgp). *RFC*, (1163), Juin 1990.

[KL91] Y. Rekhter K. Lougheed. Border gateway protocol 3 (bgp-3). *RFC*, (1267), Octobre 1991.

[Man97] B. Manning. Registering new bgp attribute types. *RFC*, (2042), Janvier 1997.

[Mil84] D.L. Mills. Exterior gateway protocol formal specification. *RFC*, (904), Avril 1984.

[PT01] J. Scudder P. Traina, D. McPherson. Autonomous system confederations for bgp. *RFC*, (3065), Février 2001.

[Sac01] Luc Saccavini. Le routage bgp-4 - présentation du protocole. Mars 2001.

[TB00] E.Chen T. Bates, R. Chandra. Bgp route reflection - an alternative to full mesh ibgp. *RFC*, (2796), Avril 2000.

[TGG00] G. Wilfong T. G. Griffin, F. B. Shepherd. The stable paths problem as a model of bgp routing. Septembre 2000.

[TGG02a] G. Wilfong T. G. Griffin. Analysis of the med oscillation problem in bgp. 2002.

[TGG02b] G. Wilfong T. G. Griffin, F. B. Shepherd. The stable paths problem and interdomain routing. *IEE/ACM Transactions on Networking*, 10, 2002.

[TK03] L. Carr-Motychova T. Klockar. Preventing oscillation in i-bgp with route reflectors. Décembre 2003.

[YR94] Eds Y. Rekhter, T. Li. Border gateway protocol 4 (bgp-4). *RFC*, (1654), Juillet 1994.

[YR95] T. Li Y. Rekhter. Border gateway protocol 4 (bgp-4). *RFC*, (1771), Mars 1995.