



HAL
open science

Conception de réseau iBGP

Marc-Olivier Buob, Steve Uhlig, Mickaël Meulle

► **To cite this version:**

Marc-Olivier Buob, Steve Uhlig, Mickaël Meulle. Conception de réseau iBGP. Colloque Francophone sur l'Ingénierie des Protocoles (CFIP), Mar 2008, Les Arcs, France. hal-00250172

HAL Id: hal-00250172

<https://hal.science/hal-00250172v1>

Submitted on 11 Feb 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Conception de réseau iBGP

Marc-Olivier Buob, Steve Uhlig, Mickaël Meulle

France Télécom R&D

38-40, rue du Général Leclerc

92794 Issy les Moulineaux Cedex 9

{marcolivier.buob,michael.meulle,}@orange-ftgroup.com

S.P.W.G.Uhlig@tudelft.nl

RÉSUMÉ. L'Internet est constitué de plus de 25,000 AS (Autonomous System) échangeant des informations de routage grâce à BGP (Border Gateway Protocol). Dans un AS de taille importante, il n'est pas possible d'établir une session BGP entre chaque paire de routeur pour des raisons de scalabilité. On a alors recours à la réflexion de route. Cependant, cette technique induit une opacité en terme de diffusion des routes, et peut provoquer l'apparition de routages sous-optimaux (en terme de coût IGP), des déflexions de routes, voire des boucles de routage. Dans ce travail nous proposons une solution pour construire une topologie de réflexion de route permettant d'avoir un routage identique à celui d'un full mesh iBGP, y compris en cas de panne simple d'équipement, et en installant un minimum de sessions iBGP. Nous avons appliqué cette méthode sur le réseau d'un opérateur tier-1 et calculé une topologie iBGP robuste à tout cas de panne simple. La topologie obtenue reste de taille comparable à celle actuellement déployée.

ABSTRACT. The Internet is made of more than 25,000 Autonomous Systems (AS) exchanging routing information with the Border Gateway Protocol (BGP). In large ASs, establishing a full-mesh of BGP sessions between the routers does not scale. Route-reflection was introduced to provide scalability. The drawback of route-reflection is the opacity in the diffusion of the routes, and may lead to suboptimal routing, deflection, and routing loops. In this work, we propose a solution to build a route-reflection topology that guarantees a routing equivalent to an iBGP full-mesh, even under simple IGP failures, and with a minimal number of iBGP sessions. We applied this method to the network of a tier-1 provider and computed an iBGP topology robust to simple IGP failures. The resulting topology has a similar size to the currently deployed one.

MOTS-CLÉS : BGP, réflexion de route, conception de topologie iBGP, optimisation

KEYWORDS: BGP, route-reflection, iBGP topology design, optimization

1. Introduction

L'Internet rassemble plus de 25,000 domaines interconnectés appelés Autonomous System (AS). Le routage à l'intérieur d'un domaine est assuré par un protocole IGP (Interior Gateway Protocol) [HAL 00], comme IS-IS ou OSPF. Ce protocole de routage permet de trouver les plus courts chemins (au sens IGP) entre chaque paire de routeurs de l'AS. Les destinations extérieures à l'AS sont apprises grâce aux informations de routage BGP [REK 95]. Les routeurs implémentant le protocole BGP sont en mesure de savoir vers quel routeur envoyer le trafic pour joindre une destination extérieure à leur AS. Le trafic est routé vers ce point de sortie en accord avec le routage IGP. Si au cours de ce chemin, le point de sortie est modifié, on parle de déflexion.

Les informations de routage BGP sont échangées entre les AS au travers des sessions eBGP (External BGP). Ces sessions sont installées sur les liens inter-domaines (i.e. les liens interconnectant deux routeurs de bordure (ASBR) appartenant à des AS différents). Les informations de routage circulent ensuite dans chaque AS grâce à des sessions iBGP (Internal BGP). Avec iBGP classique, un routeur BGP ne retransmet jamais un message iBGP vers un autre voisin iBGP. Cette règle permet de réduire le nombre de messages BGP au sein de l'AS. Ainsi, chaque routeur doit établir une session iBGP vers chaque autre routeur de l'AS afin de propager ses routes à l'ensemble des routeurs de l'AS. Une telle topologie, appelée *full mesh iBGP*, requiert $n(n-1)/2$ sessions iBGP où n est le nombre de routeurs BGP présents dans l'AS. Cette solution, communément utilisée dans les petits AS, ne passe pas à l'échelle. En effet, chaque routeur va devoir maintenir une table par session, et devenir sensible au moindre événement dans le réseau. De plus, l'ajout d'un nouvel équipement dans une telle topologie devient rapidement lourd à configurer puisqu'il nécessite une intervention sur chaque routeur BGP. C'est pourquoi dans un AS de taille importante, les opérateurs réseaux ont recours aux confédérations BGP¹ [TRA 01], ou à la réflexion de route².

Dans cet article, nous nous focalisons sur la réflexion de route car c'est la technique la plus communément utilisée dans les grands AS. Un réseau utilisant la réflexion de route peut être victime de problèmes de routage dans deux cas.

1) Certains routeurs choisissent leur route en accord avec le MED. Le Multi Exit Discriminator est un attribut BGP permettant d'implémenter une forme de "cold potato routing". Ces problèmes de routage sont étudiés dans [GRI 02a] et peuvent être facilement évités en déployant sur chaque routeur BGP l'option *always-compare-med* ou *set-deterministic-med*.

2) Les routeurs choisissent leur route sur une étape de décision BGP implémentant le "hot potato routing" : *préférer une route apprise par eBGP sur une route apprise par iBGP*, et *préférer les routes BGP dont le point de sortie est le plus proche au sens IGP* [BUO 07].

1. Ceci consiste à subdiviser un AS en plusieurs sous-AS

2. Certains routeurs BGP, appelés route reflector (RR), peuvent repropager une partie des messages iBGP qu'ils reçoivent à des voisins iBGP.

Dans un AS de taille importante, les étapes de processus de décision implémentant le “hot potato routing” sont fréquemment déterminantes [TEI 04] (70% des routes dans le réseau tier-1 que nous avons étudié). Cependant, la réflexion de route ne garantit pas qu’un routeur apprenne systématiquement son point de sortie optimal pour une destination donnée. Idéalement, le routage devrait cependant converger vers le même état que celui obtenu dans un full mesh. Une telle topologie iBGP est dite *fm-optimale* [BUO 07]. Cette propriété permet en particulier de garantir un routage optimal, déterministe, et sans déflexion ou boucle de routage.

Dans cet article, nous cherchons à construire une topologie iBGP vérifiant les critères suivants :

- **Fm-optimalité** : on souhaite concilier les avantages offerts par le full mesh iBGP (du point de vue routage) et de la réflexion de route (en terme de configuration et de charge au niveau des routeurs). En outre ce critère garantit que chaque routeur choisit son meilleur point de sortie possible pour n’importe quelle destination. Comme nous allons le voir par la suite, valider ce critère dans une topologie de RR n’est pas évident, mais possible.

- **Validité** : prouver que chaque paquet parvient à atteindre son point de sortie est un problème NP-difficile. Cependant, le concept de *fm-optimalité* garantit qu’un réseau ne comporte pas de déflexion de routage et donc pas de boucle de routage.

- **Reliabilité** : on construit une topologie iBGP aussi proche que possible de la topologie IGP [XIA 03, GRI 02b]. On évitera autant que possible les sessions iBGP multi-saut, i.e. les sessions entre routeurs qui ne sont pas directement adjacents dans le réseau.

- **Robustesse** : la topologie construite doit rester valide en cas de panne IGP. Dans cette approche, nous allons construire une topologie qui reste *fm-optimale* pour tout cas de panne IGP simple (lien ou routeur).

- **Passage à l’échelle** : on construit une topologie iBGP avec aussi peu de sessions que possible.

Sauf erreur, l’approche que nous proposons est la seule qui garantit un bon comportement du réseau pour tout cas de panne simple. Qui plus est, elle garantit que les routages restent optimaux lors de ces pannes. L’approche que nous proposons permet de traiter efficacement des instances de grands réseaux, en particulier des réseaux d’opérateurs tier-1.

La section 2 introduit les notations utilisées dans le reste de l’article. La section 3 présente la méthode de résolution utilisée. La section 4 valide cette approche d’une part sur des instances aléatoires, et d’autre part sur deux réseaux réels. L’état de l’art est présenté dans la section 5.

2. Terminologie

On note $G_{igp} = (V_{igp}, E_{igp})$ le graphe caractérisant la structure physique du réseau. Chaque sommet de V_{igp} représente un routeur, et chaque arc pondéré $(u, v) \in$

E_{igp} représente un lien physique et la métrique associée. On note $|u, v|$ la longueur du plus court chemin de u à v .

On note \mathcal{N} l'ensemble des routeurs de bordure de l'AS, et \mathcal{R} l'ensemble des routeurs implémentant le protocole BGP dans l'AS. En outre $\mathcal{N} \subseteq \mathcal{R}$. Le graphe $G_{bgp} = (V_{bgp}, E_{bgp})$ permet de représenter la topologie de réflexion de route déployée dans le réseau ($V_{bgp} = \mathcal{R}$). E_{bgp} rassemble l'ensemble des sessions iBGP installées. Chaque arc iBGP est étiqueté en accord avec les notations utilisées dans [FEA 04, GRI 02b]. Quand un routeur établit une session iBGP vers un autre routeur, on construit un arc étiqueté *up* (d'un client vers un RR), *down* (d'un RR vers un client), ou *over* entre deux routeurs (routeur de même niveau hiérarchique) (voir figure 1 et section 2).

Soit $\mathbb{L} = \{up, over, down\}$ l'ensemble des étiquettes iBGP, et $label : E_{bgp} \rightarrow \mathbb{L}$ la fonction qui retourne l'étiquette d'un lien donné. Soit $sym : \mathbb{L} \rightarrow \mathbb{L}$ la fonction qui retourne le label symétrique d'un label donné : $sym(up) = down$, $sym(down) = up$, $sym(over) = over$.

On dit qu'un chemin orienté de G_{bgp} est *valide* si celui-ci est constitué d'une suite de 0 ou plusieurs arcs *up*, suivi de 0 ou 1 arc *over*, suivi de 0 ou plusieurs arcs *down*.

A chaque fois que l'on considère une paire $(n, r) \in \mathcal{N} \times \mathcal{R}$ on suppose que :

- n est le plus proche ASBR de r au sens IGP pour un jeu de routes particulier ;
- la meilleure route choisie par chaque routeur est choisie sur le critère "privilégier le plus proche point de sortie appris" ;
- il existe un préfixe destination p annoncé par plusieurs routes concurrentes reçues via eBGP par des ASBR différents.

On cherche à garantir que r est toujours capable d'apprendre la route annoncée par son meilleur point de sortie n . Les points de sortie concurrents sous-optimaux sont donc décrits par l'ensemble : $\mathcal{N}(n, r) = \{n' \in \mathcal{N}, |r, n'| > |r, n|\}$.

S'il existe au moins un chemin iBGP valide de n à r tel que chaque routeur w de ce chemin choisit la route annoncée par n , alors r choisit la route annoncée par n . On appelle *routeur blanc* un routeur w vérifiant cette propriété. Ainsi, l'ensemble des routeurs blancs relatifs à une paire (n, r) se définit par : $\mathcal{W}(n, r) = \{w \in \mathcal{R} | \forall n' \in \mathcal{N}(n, r), |r', n| < |r', n'|\}$

On appelle *chemin blanc* tout chemin iBGP uniquement constitué de routeurs blancs. Si pour toute paire $(n, r) \in \mathcal{N} \times \mathcal{R}$, il existe au moins un chemin valide blanc, alors G_{bgp} est *fm-optimal*. On remarque en particulier que le concept de *fm-optimalité* est indépendant de la notion de préfixe. Ainsi, ce critère garantit un bon comportement du réseau pour *tout* jeu de routes BGP concurrentes.

La figure 1 illustre les concepts que nous venons d'introduire. Dans cet exemple, (y, rr, z) est un chemin iBGP valide de y à z . Cependant $rr \notin \mathcal{W}(y, z)$, donc (y, rr, z) n'est pas un chemin blanc. En effet, rr choisit la route annoncée par x (si x en reçoit une). Ainsi rr est susceptible de ne pas propager la route annoncée par y vers z .

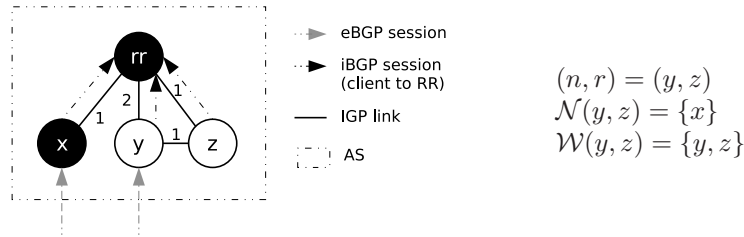


Figure 1 – Un exemple de routage sous-optimal : le trafic émis par z suit le chemin IGP (z, rr, x) au lieu de (z, y) .

3. Comment construire une topologie iBGP fm-optimale ?

Nous allons détailler dans cette partie comment résoudre le problème de conception iBGP que nous avons défini. Les paramètres requis pour ce problème sont l'ensemble des ASBR (\mathcal{N}), l'ensemble des routeurs BGP (\mathcal{R}), et la topologie IGP (G_{igp}). Notre approche part du principe que $V_{igp} = V_{bgp}$ ce qui est généralement le cas en pratique, et calcule un ensemble idéal de sessions iBGP (E_{bgp}). On modélise ce problème sous forme d'un programme linéaire en nombre entier (PLNE). Cependant, les contraintes ne peuvent pas être énumérées de manière exhaustive pour un réseau de taille importante. On génère donc dynamiquement les contraintes utiles au problème (décomposition de Benders).

1) Dans un premier temps nous n'allons pas tenir compte des pannes (section 3.1). On s'intéresse donc uniquement au comportement du réseau dans son *régime nominal*. Pour chaque paire $(n, r) \in \mathcal{N} \times \mathcal{R}$, on construit un problème satellite, satisfait si et seulement si un chemin valide blanc existe de n à r . Les problèmes satellites permettent d'alimenter le jeu de contraintes du PLNE (problème maître).

2) Dans un second temps (section 3.2), on explique comment introduire les contraintes de robustesses aux pannes. On construit pour cela autant de satellites (n, r, f) (où f désigne une panne) que nécessaire. Il est nécessaire d'agréger les satellites ainsi engendrés, sans quoi le problème est trop grand pour être résolu.

Nous n'allons pas considérer les sessions *over* dans ce problème, bien que notre modèle le permette. Ceci permet en outre de limiter le phénomène de dégénérescence du problème. En effet, chaque session *over* peut être transformée en session *up* ou *down* sans remettre en cause la validité de la topologie calculée.

3.1. Cas nominal

3.1.1. Problème maître

Pour chaque session iBGP candidate (u, v) , $(u, v) \in \mathcal{R}$, $u \neq v$, on définit deux variables booléennes : $up(u, v)$ (égale à 1 si $label(u, v) = up$, 0 sinon); $down(u, v)$ (égale à 1 if $label(u, v) = down$, 0 sinon).

On définit une fonction objectif F permettant de construire une topologie iBGP aussi proche que possible de la topologie IGP, tout en minimisant le nombre de session iBGP installées :

$$F = \min\left(\sum_{(u,v) \in \mathcal{R}} (R(u,v) \cdot (up(u,v) + down(u,v)))\right)$$

où $R(u,v)$ est égal au nombre de sauts IGP nécessaires pour établir une session iBGP de u à v .

Le problème maître comporte deux types de contraintes :

– **Les contraintes de domaine** : chaque paire de routeur $(u,v) \in \mathcal{R} \times \mathcal{R}$ est reliée par au plus 1 session iBGP. De plus $label(u,v) = sym(label(v,u))$. Pour chaque session iBGP candidate (u,v) on introduit les contraintes suivantes :

- $\forall u,v \in \mathcal{R}, up(u,v) + down(u,v) \leq 1,$
- $\forall u,v \in \mathcal{R}, up(u,v) = down(v,u).$

– **Les contraintes Max-flow Min-cut** : au début, cet ensemble de contrainte est vide. Il sera alimenté à chaque itération par les problèmes satellites (voir section 3.1.2).

A chaque itération it , le problème maître est résolu et propose aux problèmes satellites la solution qu'il vient de calculer. Chaque satellite violé remonte une contrainte Max-flow Min-cut qui sera insérée dans le problème maître. Au fil des itérations, le jeu de contraintes s'enrichit jusqu'à ce que l'ensemble des satellites soient satisfaits. Dès lors, il existe pour chaque paire (n,r) au moins un chemin blanc valide. Cette ultime résolution du problème maître permet de calculer une topologie *fm-optimale* tout en minimisant la fonction objectif F .

3.1.2. Problèmes satellites

Afin de ne construire que des chemins iBGP valides, on réutilise la transformation de graphe proposée dans [BUO 07]. Chaque sommet de V_{bgp} est transformé en un *meta noeud* composé de deux sommets (appelés *noeud source* et *noeud cible*) et un arc (appelé *arc interne*), comme montré sur la figure 2. La manière dont deux *meta noeuds* sont connectés découle directement du type de session iBGP établie entre les deux routeurs correspondants. Dans le graphe étendu, on ne peut construire que des chemins valides. On appelle *meta arc* tout arc connectant deux sommets appartenant à deux *meta noeuds* différents. Chaque *meta arc* correspond à un type de session iBGP établie entre deux routeurs. On note $[u,v,rel]$ le *meta arc* allant du *meta noeud* u au *meta noeud* v et portant la relation iBGP rel . Chaque chemin valide de G_{bgp} allant de $s \in V_{bgp}$ à $t \in V_{bgp}$ correspond ainsi à un chemin du graphe étendu allant de s_{src} à t_{dst} , où s_{src} est le *noeud source* de s et t_{dst} le *noeud cible* de t .

Un problème satellite permet de vérifier si un chemin valide blanc existe pour une paire (n,r) donnée. Un tel problème peut se modéliser à l'aide d'un problème de flot dans un graphe noté $G_w(n,r)$. Chaque sommet de ce graphe appartient à $\mathcal{W}(n,r)$ (voir section 2). Afin de réduire le nombre de sessions iBGP candidates, on considère uniquement les sessions iBGP (u,v) vérifiant la propriété $|n,u| \leq |n,v|$ et

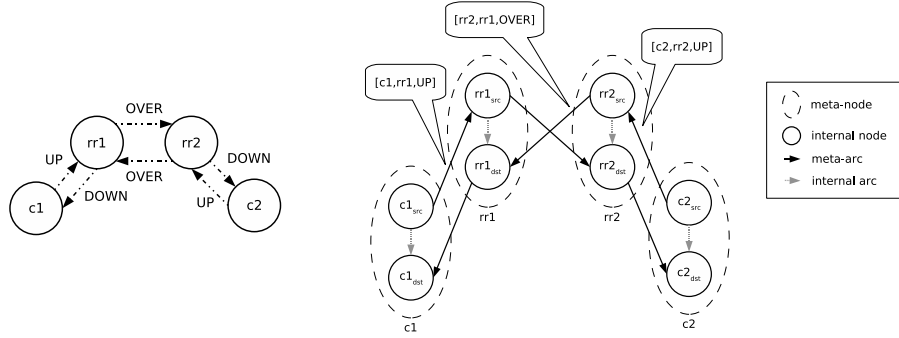


Figure 2 – Un exemple de graphe iBGP et son graphe étendu correspondant. Le chemin iBGP valide (c_1, rr_1, rr_2, c_2) dans G_{bgp} correspond au chemin $(c1_{src}, rr1_{src}, rr2_{src}, rr2_{dst}, c2_{dst})$ dans G_{bgp}^{ext} .

$|v, r| \leq |u, r|$. Ainsi, un message BGP se rapproche toujours de sa destination r et s'éloigne de sa source n au cours de la propagation iBGP. Ceci permet d'acheminer plus rapidement les messages de n vers r . Le graphe $G_w(n, r) = (\mathcal{W}(n, r), E_w(n, r))$ regroupe donc l'ensemble des chemins iBGP blancs aptes à satisfaire efficacement une paire (n, r) donnée.

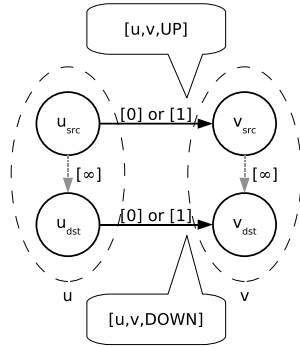


Figure 3 – Deux sessions iBGP

candidates : $label(u, v) = up$ ou $label(u, v) = down$.

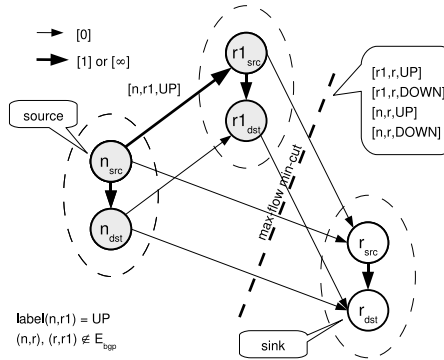


Figure 4 – La coupe min-cut max-flow engendrée :

$$up(r_1, r) + down(r_1, r) + up(n, r) + down(n, r) \geq 1.$$

On construit ensuite le graphe étendu $G_w^{ext}(n, r)$ correspondant. On note n_{src} le noeud source du meta noeud n et r_{dst} le noeud cible du meta noeud r . On installe l'ensemble des meta arcs correspondant à l'ensemble des sessions iBGP candidates. La capacité installée sur chaque meta arc (i, j) dépend des sessions installées :

- Si i et j appartiennent au même meta noeud, on installe une capacité infinie.
- Dans le cas contraire, on installe une capacité nulle ou égale à 1. Soient $rel \in$

$\{up, down\}$ le type de session iBGP correspondant à l'arc (i, j) et r_i (resp. r_j) le *meta-noeud* correspondant à i (resp. j). Si une session iBGP de type *rel* est installée de r_i à r_j , on installe sur (i, j) une capacité égale à 1 (0 autrement). En outre, au plus un *meta arc* du *meta noeud* r_i au *meta noeud* r_j est de capacité égale à 1.

Si le flot maximal de n_{src} (la source) à r_{dst} (la destination) est supérieur ou égal à 1, alors la paire (n, r) est satisfaite. Autrement aucune unité de flot ne parvient à atteindre la destination. Dans ce cas, on cherche la coupe minimale de flot maximale. On note $C(n, r, it)$ l'ensemble des *meta arcs* coupés durant l'itération. On insère ensuite dans le problème maître la contrainte linéaire max-flow min-cut :

$$\sum_{[r_i, r_j, rel] \in C(n, r, it)} (rel(r_i, r_j)) \geq 1.$$

La figure 4 permet de visualiser une telle coupe. Dans cet exemple $\mathcal{W}(n, r) = \{n, r_1, r\}$. Les itérations précédentes ont conduit à installer une session *up* de n vers r_1 et pas de session entre r_1 et r , et entre n et r . Le satellite (n, r) est ici violé. Il engendre la contrainte $up(r_1, r) + down(r_1, r) + up(n, r) + down(n, r) \geq 1$, insérée à l'itération suivante dans le problème maître.

3.2. Cas des pannes IGP

Cette section explique comment tenir compte des pannes IGP (de lien ou de routeur). Rappelons qu'une session BGP s'établit le long du plus court chemin IGP du routeur source au routeur destination de la session. Lorsqu'une panne IGP survient, chaque routeur met à jour ses plus courts chemins IGP et, dans le cas où la panne a impacté certaines sessions iBGP ces dernières sont remontées automatiquement. Si la connectivité IGP est perdue entre les deux routeurs BGP, la session BGP tombe. On note une panne IGP f une panne impactant un ou plusieurs équipements (lien ou routeur), et ϕ la panne vide.

Pour chaque paire f on recalcule le coût IGP entre chaque paire de routeur appartenant à une même composante connexe IGP. Il suffit ensuite d'appliquer le raisonnement du cas nominal au sein de chacune de ces composantes connexes. Soit une paire (n, r) telle que n et r appartiennent à une même composante connexe IGP C . On considère dans $G_w(n, r, f)$ uniquement les sommets blancs appartenant à C . En effet, une session iBGP ne peut s'établir que si les deux routeurs appartiennent à la même composante connexe IGP. Pour chaque couple (n, r) on se limite donc aux pannes f telles que n et r appartiennent toujours à la même composante connexe, et n'impactant ni n , ni r . Ce formalisme est suffisamment générique pour prendre en compte n'importe quel jeu de pannes simples ou multiples passé en paramètre du problème.

Si l'on construit tous les satellites (n, r, f) , on constate que certains sont redondants. Par exemple, si f n'affecte en rien la paire (n, r) , alors il est inutile d'introduire le problème satellite (n, r, f) dans le problème. En effet, celui-ci introduira les mêmes contraintes que (n, r, ϕ) . De manière plus générale, considérons deux pannes f, f' et

une paire (n, r) . Soient $G_w(n, r, f)$ et $G_w(n, r, f')$ les deux graphes correspondants. Si $G_w(n, r, f) \subseteq G_w(n, r, f')$, alors les contraintes introduites par $G_w(n, r, f)$ seront toujours moins restrictives que celles introduites par $G_w(n, r, f')$. On peut donc supprimer sans risque le problème satellite (n, r, f') .

4. Résultats

Nous avons d'abord appliqué notre approche sur de petites topologies (voir section 4.1) dont une celle du réseau GEANT. Nous avons ensuite résolu le problème sur le réseau d'un grand opérateur tier-1 (section 4.2). Nous avons calculé pour chaque instance calculé deux topologies iBGP : la première ne tient pas compte des cas de panne, la seconde est robuste à tout cas de panne simple IGP.

4.1. Petites topologies

Cette partie présente les résultats obtenus dans le cas nominal et le cas robuste aux pannes simples sur 5 topologies de petite taille :

- La topologie du réseau GEANT de 2004³.
- 4 topologies générées par iGen⁴. iGen permet de générer un ensemble de points aléatoire sur un ou plusieurs continents, et les connecte en utilisant différentes heuristiques de conception de réseau [CAH 98]. Nous avons généré 4 réseaux de 25 noeuds. NA correspond à une topologie IGP de réseau pour le continent Nord américain, W à une topologie mondiale. Nous avons utilisé deux heuristiques de maillage : la triangulation de Delaunay (D) et une juxtaposition de deux arbres couvrants disjoints (2T).

Afin de se placer dans le cas le plus difficile, on suppose que chaque routeur est un routeur de bordure ($\mathcal{N} = \mathcal{R} = V_{igp}$). C'est la forme la plus contrainte du problème. En effet, les topologies iBGP calculée restent *fm-optimales* même si seuls certains routeurs sont effectivement des routeurs de bordure. Plus précisément, une topologie iBGP calculée pour un plus petit jeu d'ASBR comporterait moins de session iBGP. Le tableau 1 présente les résultats obtenus avec et sans prise en compte des pannes (voir les deux lignes du bas) pour les 5 topologies que nous venons de présenter.

GEANT utilise un full mesh iBGP et requiert la configuration de 462 sessions iBGP orientées entre ses 22 routeurs. La décomposition de Benders montre que 74 sessions pourrait suffire pour obtenir le même routage dans le cas nominal, et 170 pour garantir la *fm-optimalité* du réseau en cas de panne simple. Le nombre de session iBGP double lorsque la topologie doit rester robuste aux pannes simples IGP (voir dernière ligne du tableau 1). Néanmoins les topologies ainsi construites comportent 3 fois moins de sessions que le full mesh iBGP correspondant.

3. <http://www.geant.net>

4. <http://www.info.ucl.ac.be/~bqu/igen/>

		GEANT	NA-D	NA-2T	W-D	W-2T
Graphe d'entrée	$ V_{bgp} = V_{igp} $	22	25	25	25	25
	$ E_{igp} $	72	128	96	130	96
	$ E_{bgp} $ in f.m.	462	600	600	600	600
Sans panne	$ E_{bgp} $	74	80	72	100	64
Avec panne	$ E_{bgp} $	172	168	146	194	126

Tableau 1 – Les solutions trouvées pour les petites topologies.

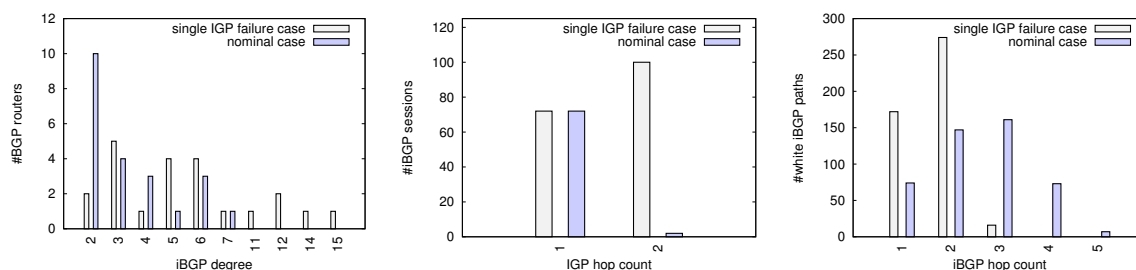


Figure 5 – Les propriétés des topologies iBGP calculées pour le réseau GEANT (cas nominal et cas robuste aux pannes simples).

Afin de caractériser la qualité des topologies calculées, nous utilisons trois indicateurs. Nous n'avons reporté que les résultats pour le réseau GEANT, car les topologies IGEN conduisent à des observations similaires (voir figure 5)

1) Distribution du degré de chaque sommet iBGP : lorsqu'un routeur établit un nombre important de session iBGP, une quantité importante de mémoire est consommée (une table par session). Le graphe de gauche indique qu'un routeur a au plus 7 voisins iBGP dans le cas nominal, contre 15 dans le cas robuste aux pannes simples.

2) Distribution de la longueur de chaque chemin blanc : nous avons calculé pour chaque paire source destination (n, r) le nombre de sauts iBGP requis. De cette valeur découle la rapidité à laquelle les messages iBGP seront traités et diffusés. Ainsi plus un chemin est long, moins l'information se diffuse rapidement. Le graphe du milieu reporté montre que dans la plupart des chemins iBGP comportent moins de deux bonds iBGP.

3) Concordance avec la topologie IGP : pour chaque session iBGP, on calcule le nombre de routeurs traversés. Idéalement la topologie iBGP devrait être aussi proche que possible de la topologie IGP [XIA 03, GRI 02b] (soit un seul bond). Le graphe de droite montre que c'est le cas pour la très grande majorité des sessions iBGP, en particulier pour le cas nominal.

4.2. Application sur le réseau de coeur d'un opérateur tier-1

Nous avons appliqué notre méthode sur le réseau d'un opérateur tier-1 composé de plusieurs centaines de routeurs BGP. Les topologies calculées sont assez éloignées

de la topologie utilisée. Ceci sous-entend donc une migration complexe à mettre en oeuvre, mais ce n'est pas très surprenant. En effet, la règle de conception couramment utilisée (3 étages hiérarchiques de RR, avec en haut de la topologie les routeurs intercontinentaux, ensuite les routeurs continentaux, et en bas les routeurs nationaux) est simple à appréhender, mais ne conduit pas à une topologie iBGP robuste et efficace.

On remarque que la topologie correspondant au cas nominal comporte 45% moins de session que la solution déployée dans le réseau et elle est en plus *fm-optimale*. En outre, chaque routeur établit moins de sessions iBGP. Les chemins blancs sont un peu plus longs que dans réseau actuel. La topologie robuste comporte 25% sessions supplémentaires à celle déployée dans le réseau, mais elle est *fm-optimale* y compris en cas de panne simple. La distribution des degrés des routeurs iBGP et la distribution des chemins blancs est proche de la solution déployée dans le réseau.

5. Etat de l'art

[DUB 99] est le premier article à mettre en évidence qu'un réseau peut être victime de boucles de routage si la topologie iBGP est mal définie. [GRI 02b] propose des conditions permettant d'éviter l'apparition de boucles de routage : 1) les RR devraient préférer les routes apprises par leur clients sur les autres, 2) chaque plus court chemin IGP devrait être valide au sens iBGP. Ces deux conditions sont cependant très restrictives. [VUT 06] propose une approche permettant de s'affranchir de la première condition.

[XIA 03] propose une formulation du problème de conception iBGP. Les auteurs introduisent deux critères (Expected Lifetime et Expected Session Loss) apportant une certaine forme de robustesse. Ils construisent ensuite une topologie iBGP sur deux niveaux hiérarchiques tout en optimisant ces deux critères. [VUT 06] propose une méthode de séparation de graphe permettant de construire itérativement des étages de RR. La topologie ainsi construite ne comporte pas de boucle de routage en régime nominal. [RAW 06] présente en détail les problèmes de routages communément rencontrés dans un réseau utilisant la réflexion de route. Cet article propose quelques conditions permettant d'éviter l'apparition de déflexion de route et d'oscillations de routage dûes au MED. La méthode proposée permet de construire une topologie de RR à deux niveaux et minimise la distance IGP entre deux voisins iBGP.

Aucune de ces approches ne permet de garantir que la topologie iBGP reste valide en cas de panne simple IGP. En outre une telle panne pourrait provoquer l'apparition de boucle de routage. De plus, seul [VUT 06] permet de construire des topologies composées de plus de deux étages de RR.

6. Conclusion

Dans cet article nous avons proposé une méthode permettant de construire une topologie iBGP de réflexion de route. D'une part, notre topologie passe à l'échelle en tirant parti de la réflexion de route. D'autre part, le réseau se comporte comme si un full mesh avait été déployé et apporte les mêmes garanties de routage, y compris en

cas de panne simple d'équipement. Notre approche est la première à apporter de telles garanties. Globalement, assurer la robustesse à chaque cas de panne simple requiert l'installation de deux fois plus de sessions iBGP.

Nous avons vu qu'iBGP diffuse les informations de routage dans un AS. Cette diffusion dépend d'une part du graphe de diffusion (donc du placement des sessions iBGP) et du mécanisme de propagation utilisé. Nous avons ici cherché à résoudre les problèmes de routage en optimisant ce graphe de diffusion. Cependant, ce graphe est fortement dépendant de la topologie IGP. Idéalement, le protocole iBGP ne devrait pas induire de telles difficultés de conception. Nous cherchons actuellement à modifier le protocole iBGP lui-même afin de pouvoir configurer un réseau iBGP à l'aide de règles de conception simples. Nous tenons à remercier Olivier Klopfenstein et Jean - Luc Lutton pour leur aide précieuse.

7. Bibliographie

- [BUO 07] BUOB M., MEULLE M., UHLIG S., « Checking for optimal egress points in iBGP routing », Proc. of the 6th IEEE International Workshop on the Design of Reliable Communication Networks (DRCN 2007), October 2007.
- [CAH 98] CAHN R., *Wide area network design : concepts and tools for optimization*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [DUB 99] DUBE R., « A comparison of scaling techniques for BGP », *SIGCOMM Comput. Commun. Rev.*, vol. 29, n^o 3, 1999, p. 44–46.
- [FEA 04] FEAMSTER N., WINICK J., REXFORD J., « A Model of BGP Routing for Network Engineering », *ACM Sigmetrics - Performance 2004*, New York, NY, June 2004.
- [GRI 02a] GRIFFIN T., WILFONG G. T., « Analysis of the MED Oscillation Problem in BGP », *ICNP '02 : Proceedings of the 10th IEEE International Conference on Network Protocols*, Washington, DC, USA, 2002.
- [GRI 02b] GRIFFIN T. G., WILFONG G., « On the correctness of iBGP configuration », *Proc. of ACM SIGCOMM*, August 2002.
- [HAL 00] HALABI B., PHERSON D. M., *Internet Routing Architectures (2nd Edition)*, Cisco Press, January 2000.
- [RAW 06] RAWAT A., SHAYMAN M. A., « Preventing persistent oscillations and loops in iBGP configuration with route reflection », *Computer Networks*, , 2006, p. 3642-3665.
- [REK 95] REKHTER Y., LI T., « A Border Gateway Protocol 4 (BGP-4) », RFC 1771, 1995.
- [TEI 04] TEIXEIRA R., GRIFFIN T., VOELKER G., SHAIKH A., « Network Sensitivity to Hot Potato Disruptions », *Proc. of ACM SIGCOMM*, August 2004.
- [TRA 01] TRAINA P., MCPHERSON D., SCUDDER J., « Autonomous System Confederations for BGP », RFC 3065, February 2001.
- [VUT 06] VUTUKURU M., VALIANT P., KOPPARTY S., BALAKRISHNAN H., « How to Construct a Correct and Scalable iBGP Configuration », *IEEE INFOCOM*, Barcelona, Spain, April 2006.
- [XIA 03] XIAO L., WANG J., NAHRSTEDT K., « Optimizing iBGP route reflection network », *IEEE INFOCOM*, 2003.