

Routage inter-domaine en mode connecté

Stefano Secci^{1,2}, Jean-Louis Rougier¹, Achille Pattavina²

¹Institut TELECOM - TELECOM ParisTech - CNRS LTCI UMR 5141. 37/39 rue Dareau, 75014, Paris, France.

²Dip. di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milan, Italy.

E-mails: {secci,rougier}@telecom-paristech.fr, {secci,pattavina}@elet.polimi.it

Abstract—Nous nous plaçons dans le cadre d’une alliance d’opérateurs et nous nous intéressons au routage inter-domaine au sein de l’alliance, avec support de pre-computation et de métriques directionnelles de qualité de service. Nous proposons un algorithme de recherche en largeur à profondeur limitée. Nous mettons en évidence les bonnes performances de cet algorithme en terme de temps de calcul et d’optimalité.

I. INTRODUCTION

Des travaux récents au sein de l’IETF permettent d’étendre la technologie MPLS pour la configuration de tunnels inter-domaine. Cette technologie nécessite le support d’une architecture de calcul de chemin inter-Autonomous System (AS). Ce dernier point a fait l’objet d’études dans le cadre du groupe de travail sur les PCEs (Path Computation Element) [1], éléments de calcul qui collaborent pour le calcul de chemins inter-AS. Dans telle architecture, nous pouvons mettre en évidence deux problèmes : (i) des méthodes distribuées sont décrites pour le calcul du chemin de bout en bout, mais elles requièrent la liste des ASs à traverser. Comment la chaîne d’AS est déterminée n’est pas spécifié; (ii) la mise en place d’un tunnel inter-AS requiert une instanciation et une activation préliminaire, pour assurer la facturation et pour gérer le filtrage des requêtes de routage et de signalisation aux frontières des domaines.

Au sein du projet ANR ACTRICE, une solution architecturale permettant de résoudre ces problèmes a été définie [2], dans un contexte économiquement viable applicable à une alliance de fournisseurs d’accès Internet - et non pas à l’Internet globale - qui désirent collaborer pour l’offre de services de réseau inter-domaine en mode connecté. À l’intérieur de l’alliance, une infrastructure de découverte de service permet à chaque partenaire de publier ses éléments de service. Ces derniers annoncent des spécifications de niveau de service (en terme de contraintes de qualité de Service (QoS)) et de coût de transit. Afin de bien modéliser les relations parfois complexes entre ASs, *les spécifications de service doivent être directionnelles*: chaque AS peut annoncer un niveau de service et un coût de transit différent par direction de transit, c.à.d. par pair (point d’entrée, point de sortie) de son réseau. L’architecture fonctionnelle complète (éléments fonctionnels, extensions protocolaires) est présentée dans [2].

II. ROUTAGE INTER-AS EN MODE CONNECTÉ AVEC MÉTRIQUES DIRECTIONNELLES

Dans telle architecture, se pose donc le problème de sélectionner la chaîne ou l’arbre d’ASs sur lequel le tunnel, point-à-point ou point-à-multipoint, sera configuré. C’est un problème de calcul de plus court chemin sous contraintes, donc NP-difficile, pour lequel nous proposons une heuristique.

Nous disposons d’éléments de service décrit par un coût directionnel $c(e, t, s)$ et par des métriques additives directionnelles $m_i(e, t, s)$, où e est l’AS d’entrée, t l’AS de transit et s l’AS de sortie. Chaque métrique directionnelle s’applique donc à une couple d’arêtes adjacentes, une entrante (e, t) et une sortante (t, s) au et du noeud représentant l’AS de transit t . Il ne s’agit donc pas d’un graphe classique avec pondération sur les arêtes, mais d’un graphe pesé sous métriques directionnelles.

Nous pourrions utiliser une transformation pour se ramener à un graphe pondéré classique, et ce afin de pouvoir réutiliser les algorithmes existants de plus court chemin sous contraintes (un noeud d’AS doit alors être “éclaté” en V noeuds distincts, où V est le nombre de AS voisins). En étudiant les caractéristiques d’un graphe d’ASs potentiellement intéressés à coopérer au sein d’une alliance pour ce type de service (et donc ceux ASs majoritairement connectés, “Tier 1” et “Tiers 2”), nous avons trouvé que le degré d’interconnexion moyen des noeuds peut être approximé avec $\bar{V} = N^{1/3}$, où N est le nombre de noeuds d’AS [4] dans le graphe. Les algorithmes classiques à complexité $O(N^3)$ pour graphes pesés sur des arêtes simples, pour le graphe transformé assumeraient une complexité $O((N \cdot N^{1/3})^3 = N^4)$ [3]. Une alternative consiste à trouver des algorithmes permettant d’utiliser directement les métriques directionnelles, sans transformation du graphe. Dans ce cas, des algorithmes de recherche à la source s’imposent pour leur capacité de suivre la direction des métriques directionnelles.

Nous avons donc défini un algorithme de recherche à la source de plus court chemin sous contraintes pour des tunnels inter-AS. Le cas point-à-point est une dégénération du cas point-à-multipoint à une seule destination. Cet algorithme est composé de deux étapes, une première d’énumération de chemins faisables, et une deuxième de couplage optimal entre les chemins énumérés pour atteindre les destinations.

Nous décrivons brièvement la première étape. À partir de l’AS de source on procède à une recherche en largeur et à profondeur limitée de chemins faisables. Etant donnée la nature sans échelle du graphe d’ASs, la profondeur de la recherche peut être limitée à 8 sauts sans crainte d’inoptimalités, même dans des alliances de très grande taille [4]. Un chemin atteignant la destination d est considéré comme faisable s’il respecte les contraintes additives et si son coût ne dépasse pas le coût seuil $v(d)$ courant pour d . Le seuil est mis à jour à chaque nouvelle énumération, et est calculé seulement après F chemins énumérés. Il s’agit de la moyenne du coût des chemins énumérés pour d ayant une variance de la moyenne plus petite que la moyenne des variances (ayant donc un coût tellement grand que le chemin ne sera pas retenu probablement dans la solution finale). De cette façon le seuil suit une tendance décroissante. Il est à noter que les chemins

comportant moins de sauts sont privilégiés par cette technique, le seuil étant potentiellement plus grand aux premiers sauts. Grâce à l'utilisation de ce seuil, de nombreuses branches sont coupées ce qui permet une réduction du temps de calcul.

Finalement, le coût projeté, i.e. coût courant + coût du plus court chemin du noeud courant vers la destination, peut être utilisé pendant l'exploration du graphe, afin d'éliminer plus de branches. Une branche est éliminée lorsque le coût projeté vers chaque destination est plus grand que le seuil correspondant à cette destination. Cette opération est possible si on connaît les coûts des plus courts chemins (sans contraintes) de tout noeud vers tout noeud (par exemple par un Floyd-Warshall sur le graphe transformé). Cette opération est certes coûteuse, mais elle permet ensuite une exploration beaucoup plus efficace du graphe. De plus, n'étant pas dépendante des contraintes de QoS des requêtes, il faut noter que cette opération peut être pré-exécutée grâce à serveurs de computation supposés disponibles dans l'architecture de routage.

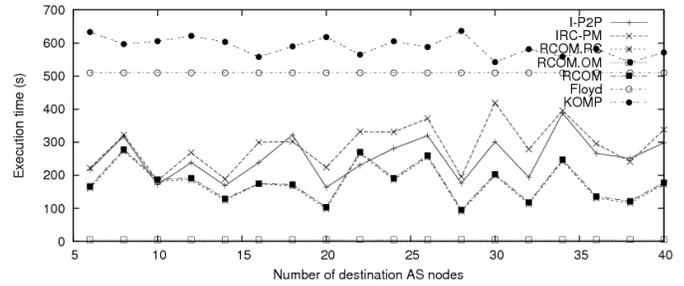
La recherche en largeur avec une profondeur limitée à $h=8$ [4], l'algorithme de recherche aurait une complexité temporelle de $O(\bar{V}^h = N^{\frac{1}{3}h}) < O(N^3)$ pour le pire cas - en approximant la base \bar{V} (facteur d'élagage du noeud) à $\sqrt[3]{N}$ (nombre moyen de voisins dans un graphe d'AS à N noeuds) comme mentionné auparavant. Grâce à l'élagage, l'algorithme d'énumération est beaucoup plus rapide comme le montre les expérimentations. Il est à noter que les heuristiques pour le routage multipoint avec contraintes additives de QoS dont nous avons connaissance n'offrent pas la possibilité de diviser la tâche en différentes phases, de manière à utiliser le pré-calcul. En effet, les contraintes de QoS sont généralement utilisées dès les premières étapes, contrairement à l'approche présentée.

Une fois les routes énumérées, on procède à leur couplage pour en extraire l'arbre d'interconnexion. Nous proposons pour cela une formulation à programmation linéaire à nombres entiers qui minimise le coût de l'arbre en garantissant que toute destination est atteinte [3]. À cette étape, il n'est plus nécessaire de vérifier les contraintes de QoS car on utilise des routes réalisables sélectionnées lors de la phase précédente.

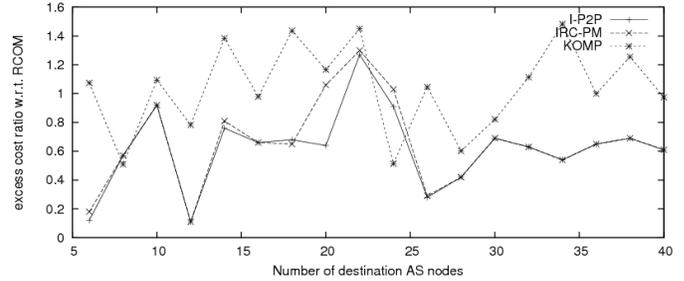
III. PERFORMANCES

Nous avons testé l'algorithme proposé sur un graphe composé dans une alliance de très grande taille (643 ASs), afin de pouvoir tester l'algorithme dans des conditions difficiles. La topologie est construite à partir de la topologie réelle de l'Internet. Nous avons effectué ces tests dans le cas d'une seule contrainte additive (le délai). Les délais, les capacités et les coûts de transit ont été engendrés d'une façon cohérente au degré d'interconnexion de chaque AS de manière à refléter la structure hiérarchique de l'Internet [4].

En figure (a) nous comparons notre algorithme (RCOM) à d'autres bien connus algorithmes de routage multi-cast avec QoS et mono-contrainte: l'algorithme de Kompella (KOMP) [5], l'algorithme de Zhu et al. (I-P2P) [6]. Pour le même graphe pesé, nous avons expérimenté l'algorithme avec un nombre de destination croissant. Nous montrons aussi les temps d'exécution d'un autre algorithme (IRC-PM) qui consiste en une utilisation itérative de la première étape d'énumération pour une seule destination, en mettant à chaque itération à zéro les métriques du chemin sélectionné pour la destination précédente. Le temps d'exécution de l'algorithme



(a) Temps d'exécution



(b) Ecart de coût par rapport à RCOM

de Floyd-Warshall (Floyd) [7] relatif au pré-calcul est présenté séparément. Nous indiquons aussi le temps d'exécution de l'algorithme d'énumération (RCOM.RC) et de l'algorithme de couplage (RCOM.OM). En figure (b) nous montrons l'écart de coût de la solution obtenue par ces algorithmes par rapport à la solution obtenue par RCOM, qui s'est démontrée toujours comme la plus proche de l'optimum [3].

Nous pouvons observer que le temps d'exécution de KOMP est borné inférieurement par celui de Floyd-Warshall (KOMP implémente une extension de Floyd avec contraintes). En prenant en compte l'étape préliminaire de Floyd, RCOM a un temps d'exécution comparable à KOMP. Néanmoins, si cette étape est pré-calculée, RCOM est pratiquement toujours le plus rapide, en donnant pourtant des coûts toujours plus faibles. Une augmentation du nombre de destinations ne semble pas aggraver sensiblement la complexité de RCOM. RCOM avec pré-calcul semble donc une approche intéressante en terme de complexité. En terme d'optimalité, RCOM borne inférieurement les autres algorithmes et le gap d'optimalité entre RCOM et les autres algorithmes dépasse fréquemment le 50%.

REFERENCES

- [1] A. Farrel, J.P. Vasseur, J. Ash, "A Path Computation Element (PCE)-based architecture", RFC 4655
- [2] R. Douville, J.-L. Le Roux, J.-L. Rougier, S. Secci, "A Service Plane over the PCE Architecture for Automatic Multi-Domain Connection-Oriented Services", *IEEE Communications Magazine*, Vol. 47, No.6, June 2008.
- [3] S. Secci, J.-L. Rougier, A. Pattavina, "AS Tree Selection for Inter-Domain Multipoint MPLS Tunnels", in *Proc. of IEEE ICC 2008*.
- [4] S. Secci, J.-L. Rougier, A. Pattavina, "On the Selection of Optimal Diverse AS-Paths for Inter-Domain IP/(G)MPLS Tunnel Provisioning", in *Proc. of 4th IEEE IT-NEWS (QoS-IP) 2008*.
- [5] V.P. Kompella et al., "Multicast routing for multimedia communication", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 3 (1993).
- [6] Q. Zhu et al., "A source-based algorithm for delay-constrained minimum-cost multicasting", in *Proc. of INFOCOM 1995*.
- [7] R.W. Floyd, "Algorithm 97: Shortest Path", *Communications of the ACM*, Vol. 5 (6) (1962)