

Routage dans les graphes cellulaires

Youssou Dieng, Cyril Gavoille

► **To cite this version:**

Youssou Dieng, Cyril Gavoille. Routage dans les graphes cellulaires. 9ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, May 2007, Ile d'Oléron, France. pp.91-94, 2007. <inria-00176944>

HAL Id: inria-00176944

<https://hal.inria.fr/inria-00176944>

Submitted on 5 Oct 2007

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.

Routage dans les graphes cellulaires

Youssou Dieng and Cyril Gavoille[†]

LaBRI, Université de Bordeaux, 351 cours de la Libération, 33405 Talence, France. {dieng,gavoille}@labri.fr

Il a été récemment établi que tout arbre à n sommets possède un schéma de routage de plus courts chemins utilisant des adresses, des en-têtes et des tables de routage de $O(\log n)$ bits. Il est ouvert de savoir si ce résultat optimal peut être étendu à des graphes plus généraux. En particulier Fraigniaud *et al.* (STACS '02) ont posé la question de savoir si ce résultat restait vrai pour les graphes planaires extérieurs, une classe de graphes planaires contenant les arbres.

Dans cet article, nous répondons positivement à cette question. Nous étendons le résultat à des graphes beaucoup plus généraux, appelés k -cellulaires. En fait, tout graphe est k -cellulaire pour une certaine valeur de k . De plus, dans le cas des graphes dessinés sur des surfaces de genre fini, les graphes k -cellulaires capturent le cas où tous les sommets peuvent être couverts par $O(k)$ faces disjointes du plongement, et le schéma de routage peut être alors construit en temps quadratique.

Keywords: routage compact, graphe planaires extérieurs

1 Introduction

Le routage point à point de messages est une fonctionnalité fondamentale des réseaux. Les composants d'un schéma de routage sont les tables de routage, les adresses, les en-têtes, et un algorithme de routage. Ce dernier prend comme paramètre l'en-tête du message arrivant et qui, en fonction de la table de routage du sommet courant, détermine le numéro de port (de l'arête) sur lequel le message doit être retransmis. Le premier en-tête du message est généré par l'émetteur à partir de sa table de routage et de l'adresse de la destination. Dans beaucoup de schémas l'en-tête se résume à l'adresse de destination.

Si la longueur des routes induites par le schéma est clairement un critère de qualité, l'espace requis par les tables de routage, la taille des adresses et des en-têtes, ou encore le temps de latence dans chaque routeur, sont autant de paramètres essentiels qui doivent être optimisés.

Beaucoup d'efforts ont été consacrés à la conception de schémas de routage équilibrant la charge de chaque routeur en terme de quantité d'information, l'idée que certains nœuds doivent jouer des rôles différents devenant une véritable faiblesse du schéma. D'autant plus que d'un point de vue théorique, il semble toujours possible de répartir l'information de routage équitablement sur tous les nœuds du réseau tout en gardant la même qualité sur la longueur des routes.

La conception de schémas de routage comportant un compromis taille des tables/longueur des routes passe par l'étude structurelle de la topologie sous-jacente. Par exemple, un des buts recherchés pour les réseaux planaires est de router à la « boussole », sans table de routage, uniquement à l'aide des coordonnées des sommets du réseau. Malheureusement, même pour des structures aussi simple que les triangulations cet objectif n'est pas atteignable : pour tout routage de plus courts chemins la taille des tables ou des adresses est au moins de $\Omega(n^{1/4})$ bits pour des triangulations de degré borné, et au moins $\Omega(n^{1/2})$ bits pour les réseaux à croissance bornée (et donc de dimension doublante bornée) [1].

Pour des structures arborescentes, des solutions avec un espace polylogarithmique existent. Pour les arbres, $O(\log n)$ bits suffisent [3, 2, 8] et cela s'étend aux graphes de largeur arborescente bornée avec $O(\log^2 n)$ [7]. Cependant la borne est connue pour être optimale seulement dans le cas des arbres [4]. Ce résultat pour les arbres est particulièrement intéressant en théorie du routage compact. Par exemple, le

[†]Supported by the projects "GéoComp" of the ACI Masses de Données, and the ANR blanche "Décomposition de Graphes".

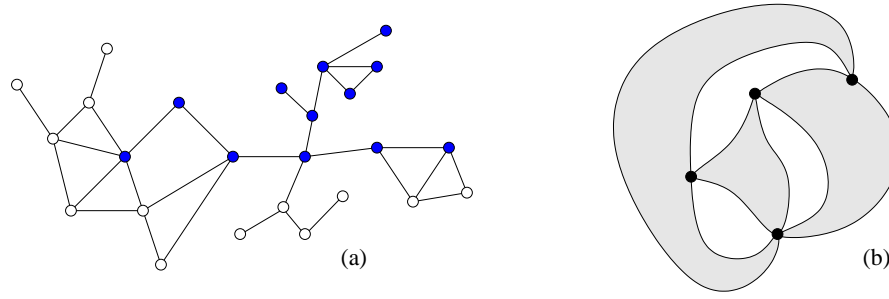


Fig. 1: (a) Une cellule où les sommets colorés forment un segment. (b) Trois cellules formant un graphe 9-cellulaire.

résultat quasi-optimal de Thorup et Zwick [8] est très dépendant du routage dans les arbres. Il énonce que tout graphe possède un schéma de routage avec des tables de taille approximativement $n^{1/k}$ par sommet garantissant des routes de longueur au plus $O(k)$ fois la distance, pour tout paramètre $k \geq 1$.

Fraigniaud *et al.* [4] ont posé explicitement la question de savoir si le schéma optimal des arbres pouvait être étendu à d'autres graphes, en particulier est-ce que tout graphe planaire extérieur, une surclasse des arbres (voir Fig. 1), possède un schéma de routage de plus courts chemins avec tables et adresses sur $O(\log n)$ bits ? La meilleure borne connue étant de $O(\log^2 n)$ bits [7].

Nous répondons positivement à cette question. Nous montrons également que si le graphe est décomposable en *cellules* (qui sont des graphes planaires extérieurs connexes) alors les tables de routage augmentent jusqu'à $O((k+1)\log n)$ bits/sommet où k représente le nombre total (avec multiplicité) des sommets partageant plusieurs cellules, les en-têtes et les adresses restant de taille $O(\log n)$ bits.

Il est important de noter que le schéma que nous présentons est *universel*, il s'applique à tout graphe en choisissant une valeur adéquate pour k . Pour le cas planaire (et de genre borné), il existe un algorithme quadratique pour calculer la décomposition en cellules [5] D'une certaine façon ce schéma universel exploite la nature géométrique de la topologie sous-jacente.

2 Préliminaires

Une *cellule* d'un graphe G est le plongement d'un sous-graphe connexe de G sur le plan sans croisement d'arête où tous les sommets bordent la face extérieure (cf. Fig. 1(a)). Donc chaque cellule d'un graphe est un graphe planaire extérieur connexe.

Une *couverture* d'un graphe G est un ensemble de sous-graphes de G tel que chaque arête de G soit contenue dans au moins un sous-graphe. On appelle *attachement* tout sommet appartenant à plusieurs cellules. Le *nombre d'attachements* d'une couverture est la somme, sur toutes les cellules, du nombre d'attachements des cellules. Finalement, un graphe *k-cellulaire* est un graphe connexe possédant une couverture en cellules de nombre d'attachements au plus k (cf. Fig. 1(b)).

Les graphes planaires extérieurs connexes sont donc 0-cellulaires, et réciproquement. Tout graphe connexe à m arêtes est $2m$ -cellulaire, chaque cellule correspondant à une arête et le nombre total d'attachements à la somme des degrés.

Une *partition k-cellulaire* est une partition en cellules de nombre d'attachements au plus k . Il est facile de voir que tout graphe k -cellulaire possède une partition k -cellulaire, car supprimer une arête partagée par plusieurs cellules d'une couverture reste une couverture cellulaire dont le nombre d'attachements n'est pas plus grand.

Un *segment* est un ensemble de sommets consécutifs d'une cellule obtenu en parcourant les sommets le long de la face extérieure.

Pour un arbre T enraciné en x , on note T_u le sous-arbre de T enraciné en u induit par les descendants de u dans T . Un résultat dû à Frederickson et Janardan [6] est que si T enraciné en x couvre un graphe planaire extérieur (donc une cellule), alors pour tout voisin u de x , T_u est un segment.

3 L'algorithme de routage

Considérons une partition k -cellulaire d'un graphe G connexe à n sommets.

Pour tout sommet x de G , soit $\text{ATTACH}(x)$ une table de routage construite à partir d'un arbre de plus courts chemins couvrant G enraciné en x dans laquelle seuls les sommets descendants d'au moins un attachement sont préservés. Donc, pour toute destination y , $\text{ATTACH}(x)[y]$ donne le numéro de port de l'arête menant vers y si dans l'arbre choisi par x le chemin utilise au moins un attachement.

Pour toute cellule C de la partition, on choisit un arbre enraciné R^C couvrant³ C , de plus courts chemins dans C . Notons que R^C n'est pas forcément un arbre de plus courts chemins de G . Pour tout sommet x de C , on note $\text{TABLECELL}(x, C)$ une table de routage construite à partir d'un arbre enraciné en x couvrant C et de plus courts chemins dans C dans lequel on supprime les sous-arbres issus des voisins de x ne contenant que des descendants de x dans R^C . Donc, pour toute destination $y \notin R_x^C$, $\text{TABLECELL}(x, C)[y]$ donne le numéro de port de l'arête menant vers y .

L'algorithme de routage présenté ci-dessous est élémentaire, l'originalité venant de son implémentation.

Routage de $x \in G$ vers $y \in C$:

1. si $y \in \text{ATTACH}(x)$, retourner $\text{ATTACH}(x)[y]$, sinon
2. si $y \in R_x^C$, router dans l'arbre R^C , sinon
3. retourner $\text{TABLECELL}(x, C)[y]$.

Montrons que le routage est de plus courts chemins. Notons qu'il suffit de montrer que la première étape du routage en x retourne une arête sur un plus court chemin de x à y . C'est évidemment vrai dans le cas 1, si $y \in \text{ATTACH}(x)$. Si $y \notin \text{ATTACH}(x)$ cela signifie qu'il existe un plus court chemin entre x et y complètement contenu dans la même cellule, disons C , tout chemin entre deux cellules devant sinon intersecter un attachement. Il suit qu'un plus court chemin entre x et y dans C l'est aussi dans G . Donc par construction de l'arbre R^C , le routage dans le cas 2 est selon un plus court chemin. Finalement, dans le cas 3 le routage est aussi selon un plus court chemin dans C , donc dans G .

4 Implémentation

Il a été démontré dans [3, 8] que tout arbre à n sommets possède un schéma de routage utilisant des tables, des en-têtes et des adresses de $O(\log n)$ bits⁴. Pour tout arbre R , on notera $\lambda(x, R)$ la concaténation de l'adresse et de la table de x de ce schéma de routage appliqué à R si bien que le routage de x à y peut être déterminé par la seule donnée du couple $(\lambda(x, R), \lambda(y, R))$.

Tout sommet x de G possède un indice $\in \{1, \dots, n\}$ obtenu comme suit : les sommets d'attachements sont numérotés en premier dans un ordre quelconque, les autres sommets étant numérotés consécutivement selon un parcours bordant la face extérieure de chaque cellule.

L'adresse du sommet x est composée du triplet $(i, C, \lambda(x, R^C))$, où i est l'indice de x et C une cellule arbitraire contenant x . Les en-têtes sont simplement composés des adresses, qui sont donc sur $O(\log n)$ bits.

La taille des tables de routage du sommet x est donnée par les tables $\text{ATTACH}(x)$ et $\text{TABLECELL}(x)$, ainsi que $\lambda(x, R^C)$ pour toutes les cellules C contenant x . Il y en a plusieurs si x est un attachement. Il faut aussi stocker une table indiquant la plage des numéros de ports utilisés pour chaque cellule C contenant x . Notons que pour $k \neq 0$ il ne peut avoir plus de cellules que le nombre d'attachements. Autrement dit, x appartient à au plus k cellules. Donc les plages de degrés ainsi que les données $\lambda(x, R^C)$ ne coûtent que $O(k \log n)$ pour la table de x . Reste à évaluer les tailles des tables $\text{ATTACH}(x)$ et $\text{TABLECELL}(x)$.

Les deux tables sont implémentées de manière similaire en compactant toutes les destinations utilisant la même arête (c'est-à-dire le même numéro de port) sous forme d'intervalles d'entiers consécutifs. Pour cela

³ R^C existe car C est connexe.

⁴ Ce schéma permette également de déterminer si la destination est un descendant du sommet courant, c'est-à-dire si la route ne passe pas par le père. Dans ce papier, on suppose que les numéros de ports de x peuvent être permutés dans d'intervalle $[1, \text{deg}(x)]$ afin d'optimiser la table de routage. Dans le cas où les numéros de ports ne peuvent être permutés, alors la taille des étiquettes $\lambda(x, R)$ augmente jusqu'à $O(\log^2 n / \log \log n)$.

on utilise l'indice des sommets. L'appartenance de y à un segment X donné se fait simplement en testant si l'indice de y est contenu dans un intervalle d'entier (modulo $|C|$) qui code X .

À tout sous-ensemble X de sommets d'une cellule, on associe $\text{seg}(X)$ le plus petit nombre de segments dont l'union est X . Nous avons le résultat clé suivant :

Lemme 1 *Soit C une cellule de G , x un sommet de G , et T un arbre de G enraciné en x et couvrant C . Soit A l'ensemble des attachements de C distincts de x et soit U l'ensemble des voisins u de x tel que $T_u \cap A \neq \emptyset$. Alors $\sum_{u \in U} \text{seg}(T_u \cap C) \leq 2|A| - 1$.*

Autrement dit, la partie de $\text{ATTACH}(x)$ concernant la cellule C peut se compacter en $O(|A|)$ intervalles d'indices (le nombre n'étant pas exactement $2|A| - 1$ car les attachements sont numérotés avant tous les autres sommets et peuvent ainsi ajouter des intervalles). Au total, $\text{ATTACH}(x)$ s'implémente avec $O(k)$ intervalles d'indices, la somme des attachements de G étant de k par définition.

La recherche de $y \in \text{ATTACH}(x)$ peut être effectuée par une recherche dichotomique en temps $O(\log |A|) = O(\log k)$ en rangeant les intervalles par cellules, la cellule C de y étant codée dans l'adresse de y .

Le résultat suivant nous indique qu'en fait $\text{TABLECELL}(x)$ peut être implémentée suivant le même principe avec 5 intervalles indices seulement.

Lemme 2 *Soit R un arbre enraciné couvrant une cellule C et de plus courts chemins. Alors, tout sommet x a au plus 5 voisins u pour lesquels il existe un sommet $y \notin R_x$ tel que la distance dans C entre x et y augmente si l'arête $\{x, u\}$ est supprimée.*

L'adresse de y contient C et $\lambda(y, R^C)$. Donc en temps constant le sommet x peut décider si $y \in R_x^C$ et le cas échéant calculer la route à l'aide de $\lambda(x, R^C)$. Nous avons vu que la recherche de y dans $\text{ATTACH}(x)$, étant donné C , en temps $O(\log k)$, tous les autres tests étant constants. Nous avons donc montré :

Théorème 1 *Tout graphe k -cellulaire possède un schéma de routage de plus courts chemins utilisant des tables de $O((k+1)\log n)$ bits par sommets, des adresses et des en-têtes de $O(\log n)$ bits. Le temps de décision est de $O(\log k)$ par sommet.*

Le théorème 1 s'étend au modèle *ports fixés*, c'est-à-dire le cas où les numéros de ports sont fixés avant la construction des tables et des adresses, et ne peuvent être changés. Alors, les termes « $\log n$ » de l'énoncé précédent devant être remplacés par « $\log^2 n / \log \log n$ ».

5 Conclusion et perspectives

Le schéma présenté peut être étendu aux graphes valués k -cellulaires avec les mêmes performances, moyennant un algorithme et une implémentation plus sophistiqués pour le routage à l'intérieur d'une cellule (dans le cas valué le lemme 2 devient faux).

Notons que c'est une propriété combinatoire qui est au cœur de notre schéma, en particulier la décomposition cellulaire n'impose aucune restriction sur la métrique des cellules : la distance entre deux sommets d'une cellule est sans relation avec celle dans le graphe en entier. Il serait intéressant d'étendre le schéma à d'autres graphes, et particulier aux graphes de largeur arborescente bornée, mais la taille des séparateurs ne semble pas suffisante pour être exploité dans le cadre du routage.

Une autre piste est d'étendre le schéma aux graphes excluant certaines sous-structures, le fait que les graphes planaires extérieurs excluent $K_{2,3}$ se révélant fondamental dans notre schéma.

References

- [1] I. ABRAHAM, C. GAVOILLE, A. V. GOLDBERG, AND D. MALKHI, *Routing in networks with low doubling dimension*, in 26th International Conference on Distributed Computing Systems (ICDCS), IEEE, July 2006.
- [2] P. FRAIGNIAUD AND C. GAVOILLE, *Comment router dans un arbre ?*, in 3^{èmes} AlgoTel, May 2001, pp. 85–88
- [3] P. FRAIGNIAUD AND C. GAVOILLE, *Routing in trees*, in 28th ICALP, vol. 2076 of LNCS, July 2001, pp. 757–772.
- [4] P. FRAIGNIAUD AND C. GAVOILLE, *A space lower bound for routing in trees*, in 19th STACS 2002, pp. 65–75.
- [5] G. FREDERICKSON, *Planar graph decomposition and all pairs shortest paths*, JACM, 38 (1991), pp. 162–204.
- [6] G. FREDERICKSON AND R. JANARDAN, *Designing networks with compact routing tables*, Algorithmica, 3 (1988).
- [7] D. PELEG, *Proximity-preserving labeling schemes*, Journal of Graph Theory, 33 (2000), pp. 167–176.
- [8] M. THORUP AND U. ZWICK, *Compact routing schemes*, in 13th Annual ACM SPAA, July 2001, pp. 1–10.