



HAL
open science

Complexité du calcul de chemins dans les réseaux multicouches

Mohamed Lamine Lamali, Nasreddine Fergani, Johanne Cohen, Hélia Pouyllau

► **To cite this version:**

Mohamed Lamine Lamali, Nasreddine Fergani, Johanne Cohen, Hélia Pouyllau. Complexité du calcul de chemins dans les réseaux multicouches. ALGOTEL 2017 - 19èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, May 2017, Quiberon, France. hal-01516573

HAL Id: hal-01516573

<https://hal.science/hal-01516573v1>

Submitted on 1 May 2017

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.

Complexité du calcul de chemins dans les réseaux multicouches[†]

Mohamed Lamine Lamali¹ et Nasreddine Fergani² et Johanne Cohen³ et Hélia Pouyllau⁴

¹LaBRI – Université de Bordeaux. France

²Queen Mary University of London. UK.

³LRI, Univ. Paris-Sud, CNRS, Université Paris-Saclay. France.

⁴Thales Research & Technology. France.

Les réseaux d'opérateurs sont constitués de différentes couches où plusieurs protocoles coexistent. Le passage d'un protocole à un autre peut se faire par conversion (changement d'en-tête des paquets), encapsulation (les paquets sont insérés dans d'autres paquets) ou désencapsulation (les paquets sont retirés d'autres paquets). Actuellement, la plupart de ces réseaux possèdent un plan de contrôle pour chaque couche pour gérer le routage, ce qui induit des chemins sous-optimaux et une surutilisation des ressources du réseau. L'idéal serait d'unifier les plans de contrôle et d'effectuer un calcul de chemins qui engloberait toutes les couches et tous les protocoles, aboutissant ainsi à un chemin globalement optimal. Cela nécessite la conception d'un algorithme qui prenne en compte les encapsulations de protocoles et les contraintes qu'ils impliquent. L'approche par la théorie des langages [LPB13] a donné le premier algorithme polynomial pour le calcul du plus court chemin prenant en compte les encapsulations de protocoles. Dans cet article, nous généralisons largement l'approche de [LPB13] en proposant un algorithme qui calcule le chemin de poids minimum selon n'importe quelle métrique additive, qui prend en compte tous les types de fonctions d'adaptation et dont la complexité est plus basse. Nous montrons également que le problème est NP-difficile sous contrainte de bande passante, même sous certaines restrictions sur le réseau.

La version complète de cet article est [LFCP16] (<https://arxiv.org/abs/1601.01786>).

Mots-clés : Réseaux multicouches, Calcul de chemins, Automates à Pile, NP-complétude

1 Introduction

Aujourd'hui, la plupart des réseaux de télécommunication comportent plusieurs couches et plusieurs protocoles, parfois sur des technologies différentes. Pour passer d'une couche à une autre ou d'un protocole à un autre, certains routeurs possèdent des *fonctions d'adaptation* : elles peuvent être des *conversions* (ex., IPv4→IPv6 avec changement d'en-tête), des *encapsulations* (ex., une trame Ethernet insérée dans un paquet IP/MPLS, comme spécifié par l'architecture Pseudo-Wire) ou des *désencapsulations*. Pour calculer un chemin de bout en bout tout en prenant en compte cette hétérogénéité protocolaire, il est nécessaire de disposer d'algorithmes capables de prendre en compte ces fonctions d'adaptation, celles-ci pouvant être imbriquées. L'utilité d'un tel algorithme serait de pouvoir unifier les plans de contrôle des différentes couches (par exemple la couche optique et la couche IP), mais plus généralement d'effectuer un calcul de chemin dans tout contexte où les encapsulations et/ou conversions interviennent : mise en place de tunnels, protocoles de sécurité (ex., IPsec), réseaux privés virtuels, etc.

Les algorithmes de calcul de chemins classiques (Dijkstra, Floyd-Warshall, etc.) ne sont pas utilisables dans ce contexte car ils ne prennent pas en compte les fonctions d'adaptation. Certains travaux proposent des algorithmes spécifiques au problème (ex., [KD09]) mais ils procèdent par recherche exhaustive et sont exponentiels. L'approche par la théorie des langages [LPB13] a donné le premier algorithme polynomial pour le calcul du plus court chemin prenant en compte les encapsulations de protocoles. Néanmoins, cet

[†]Ce travail a été réalisé quand les deux premiers auteurs étaient à Nokia Bell Labs.

algorithme ne minimise que le nombre de lien ou d'encapsulations dans un chemin, de plus il ne prend pas en compte les conversions. Dans cet article, nous généralisons largement l'approche de [LPB13] en proposant un algorithme qui calcul le chemin de poids minimum selon n'importe quelle métrique additive, qui prend en compte les conversions de protocoles et dont la complexité est substantiellement plus basse que la version précédente. Notre approche consiste à modéliser le réseau en automate à pile pondéré, puis de le convertir en grammaire pondérée et de générer le mot ayant l'arbre de dérivation de poids minimum dans cette grammaire. Ce mot permettra de trouver le chemin recherché. Mais ces résultats ne sont valables que sous hypothèse de bande passante illimitée. Dans la deuxième partie de cet article, nous montrons que le même problème devient NP-difficile si on y ajoute une contrainte de bande passante.

Par abus de langage, nous dirons qu'un *protocole a est encapsulé dans (resp. converti en) protocole b* pour signifier que les paquets du protocole *a* sont encapsulés dans (resp. convertis en) paquets du protocole *b*.

2 Modèle

Convention de notation. Les lettres en minuscule désignent des protocoles (ex., a, b, c, x, y) ou des fonctions (ex., f, h, ℓ). Les lettres en majuscule désignent des sommets ou des liens (ex., U, V, E) et les lettres calligraphiées désignent des ensembles (ex., $\mathcal{G}, \mathcal{V}, \mathcal{E}$).

Un réseau multicouche est un quadruplet $\mathcal{N} = (\mathcal{G}, \mathcal{A}, \mathcal{F}, h)$ où :

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est un graphe orienté modélisant la topologie du réseau. Les sommets dans \mathcal{V} représentent les routeurs et les arcs \mathcal{E} représentent les liens physiques.
- L'alphabet $\mathcal{A} = \{a, b, c, \dots\}$ représente l'ensemble des protocoles disponibles dans le réseau (mais pas forcément à chaque sommet).
- Pour chaque sommet $U \in \mathcal{V}$, $\mathcal{F}(U)$ est l'ensemble des fonctions d'adaptation disponibles sur le sommet U . Les fonctions d'adaptation peuvent être :
 - *Conversion* : Un protocole a est converti en un protocole b par changement d'en-tête. Cette opération est notée $(a \rightarrow b)$.
 - *Fonction passive* : Le protocole a est réémis tel qu'il est reçu. C'est un cas spécial de conversion (neutre) et est notée $(a \rightarrow a)$.
 - *Encapsulation* : Un protocole a est encapsulé dans un protocole b . Cette opération est notée $(a \rightarrow ab)^\ddagger$.
 - *Désencapsulation* : Un protocole a est désencapsulé d'un protocole b . Cette opération est notée $(a \rightarrow ab)$.
- $h : \mathcal{V} \times \mathcal{F} \times \mathcal{V} \rightarrow \mathfrak{R}_+$ est une fonction de poids. $h(U, f, V)$ (où $U, V \in \mathcal{V}$ et $f \in \mathcal{F}(U)$) est le coût de prendre le lien (U, V) en utilisant la fonction d'adaptation f sur U . La fonction h permet de représenter n'importe quelle métrique additive sur les liens et/ou les fonctions d'adaptation.

2.1 Faisabilité d'un chemin

Un chemin de la source S à la destination D est une séquence $Sf_0U_1f_1U_2f_2\dots U_nf_nD$ où chaque U_i est un sommet et chaque f_i est une fonction d'adaptation (f_0 étant une fonction fictive). Un tel chemin est faisable si :

1. La séquence $SU_1U_2\dots U_nD$ est un chemin (au sens classique) dans $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ et chaque $f_i \in \mathcal{F}(U_i)$;
2. Chaque protocole encapsulé doit être désencapsulé plus loin dans le chemin en respectant l'ordre des encapsulations. La continuité protocolaire doit être assurée (i.e., si le chemin contient une fonction, f_i t.q. $f_i = (a \rightarrow b)$, $a, b \in \mathcal{A}$, alors $f_{i+1} = (b \rightarrow a')$ ou $f_{i+1} = (b \rightarrow ba')$ ou $f_{i+1} = (a' \rightarrow a'b)$, $a' \in \mathcal{A}$).

En fait, les séquences de protocoles des chemins faisables dans un réseau peuvent être caractérisées comme un langage bien parenthésé [LPB13]. Il peut parfois être utile de considérer un chemin comme étant une suite de triplets (U_i, f_i, U_{i+1}) , $i = 0 \dots n$, où (U_i, U_{i+1}) est un lien et f_i une fonction d'adaptation sur U_i . Le poids d'un chemin \mathcal{P} serait défini par $h(\mathcal{P}) = \sum_{i=0}^n h(U_i, f_i, U_{i+1})$, avec $U_0 = S$ et $U_{n+1} = D$.

‡. Il est à noter qu'un protocole peut être encapsulé dans lui-même (ex. IP dans IP, spécifié dans la RFC 2003).

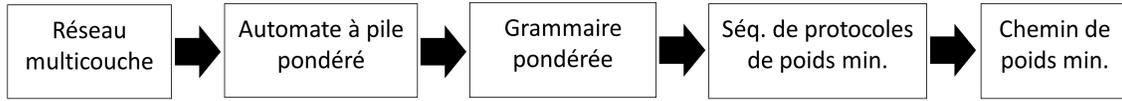


FIGURE 1: Approche utilisée pour le calcul du chemin de poids minimum.

3 Calcul du plus court chemin sans contrainte de bande passante

Le problème à résoudre est donc le suivant :

$$\min_{\mathcal{P}} h(\mathcal{P}) = \sum_{(U,f,V) \in \mathcal{P}} h(U,f,V) \text{ t.q. } \mathcal{P} \text{ est un chemin faisable de } S \text{ à } D \quad (1)$$

3.1 Un algorithme polynomial pour le calcul de chemin

Notre approche pour résoudre le problème est la suivante :

1. Modéliser l'ensemble des protocoles comme un alphabet et le réseau multicouche comme un automate à pile pondéré (AAPP) :
 - La pile de l'AAPP représentera la pile des protocoles encapsulés.
 - Chaque encapsulation (resp. désencapsulation) sera transformée en empilement (resp. dépilement) dans l'AAPP.
 - Les conversions seront transformées en transitions sans accès à la pile.
 - Le poids de chaque triplet (U, f, V) , où (U, V) est un lien et f une fonction d'adaptation utilisée sur U sera répercuté sur le poids de la transition correspondante dans l'AAPP. Par exemple, si l'on veut minimiser le nombre d'encapsulations dans le chemin, on posera $h(U, f, V) = 1$ si f est une encapsulation, 0 sinon, et ce pour tous les triplets.

La complexité de cette étape est en $O(|\mathcal{A}|^3 \times |\mathcal{E}|)$.

2. L'AAPP est converti en grammaire à contexte libre pondérée (GCLP) selon une méthode classique [HMU06]. Chaque transition est transformée en ensemble de règles de production, le poids de la transition se répercutera sur ces règles. La complexité de cette étape est en $O(|\mathcal{A}|^5 \times |\mathcal{V}|^2 \times |\mathcal{E}|)$.
3. Grâce à une généralisation aux grammaires de l'algorithme de Dijkstra par Knuth [Knu77], nous pouvons obtenir le mot généré par la GCLP ayant l'arbre de dérivation de poids minimal. Ce mot correspond à la séquence de protocoles du chemin à calculer. Cet algorithme a une complexité en $O(|\mathcal{A}|^5 \times |\mathcal{V}|^2 \times |\mathcal{E}|)$ s'il est implémenté avec des tas de Fibonacci.
4. A partir de la séquence de protocoles de poids minimum, un algorithme de recherche de motifs retrouve la suite de sommets et de fonctions d'adaptation qui correspond à la séquence de poids minimum, cette suite est le chemin que l'on recherche. Cette étape a une complexité en $O(|\mathcal{T}_{\mathcal{P}}| \times |\mathcal{V}| \times |\mathcal{E}|)$, où $|\mathcal{T}_{\mathcal{P}}|$ est la longueur (nombre de liens) du chemin.

Cette approche est illustrée par la Figure 1.

4 Calcul du plus court chemin sous contrainte de bande passante

Dans la section précédente, nous avons proposé un algorithme polynomial pour le calcul de chemins de poids minimum dans un réseau multicouche. Néanmoins, les seuls chemins faisables dans un tel réseaux peuvent comporter des boucles (nécessaires dans certaines configurations pour accumuler des encapsulations). Or, la présence de boucles peut modifier fondamentalement le problème. Si la bande passante est supposée illimitée, on peut passer autant de fois que l'on veut par le même lien, mais si la bande passante est limitée, ce n'est plus possible. Imaginons un lien de capacité 100Mbps et une chemin nécessitant 50Mbps, le chemin pourra passer par ce lien deux fois, mais pas trois. Le problème devient donc le suivant :

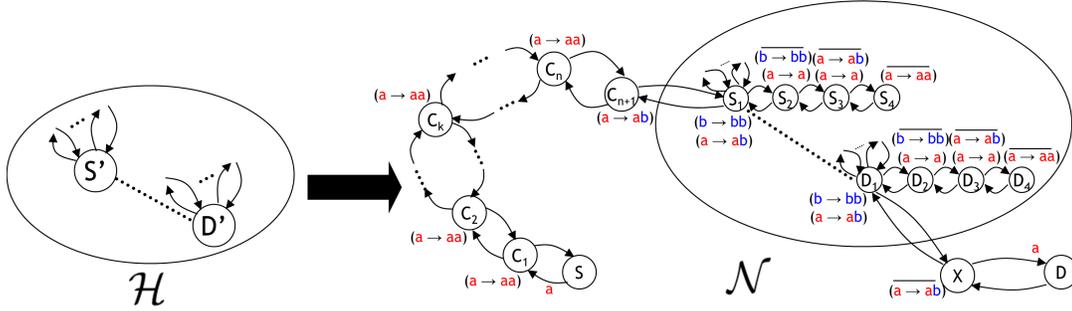


FIGURE 2: Réduction du chemin hamiltonien dans un graphe symétrique au problème du chemin faisable.

$$\min h(\mathcal{P}) = \sum_{(U,f,V) \in \mathcal{P}} h(U, f, V)$$

$$t.q. \begin{cases} \mathcal{P} \text{ est un chemin faisable de } S \text{ à } D \\ \min_{E \in \mathcal{P}} \frac{q_b(E)}{nb(E)} \geq q_b^{min} \end{cases} \quad (2)$$

Où $nb(E)$ est le nombre de fois où E est parcouru par le chemin \mathcal{P} , $q_b(E)$ est la capacité de E et q_b^{min} est la contrainte de bande passante. Dans un réseau classique, un chemin ne doit pas boucler, et donc la contrainte de bande passante est simplement résolue en supprimant les liens dont la capacité n'est pas suffisante. Mais dans un réseau multicouche, ce n'est plus possible, car il se peut que la capacité d'un lien soit suffisante pour un passage du chemin, mais pas plusieurs. Il est impossible de savoir à l'avance (avant de calculer un chemin) combien de fois il va passer par un certain lien. En fait, la contrainte de bande passante rend le problème NP-difficile.

Proposition 1 *Le problème de décision associé au problème (2) est NP-difficile, même avec deux protocoles et sur un graphe symétrique*[§].

Idée de la preuve : Soit \mathcal{H} un graphe orienté symétrique, la preuve consiste à construire à partir de \mathcal{H} , un réseau multicouche \mathcal{N} symétrique et ayant deux protocoles, tel qu'il existe un chemin faisable dans \mathcal{N} du sommet S au sommet D si et seulement si il existe un chemin hamiltonien dans \mathcal{H} du sommet S' au sommet D' . Une première étape consiste à ajouter une « queue » à \mathcal{H} pour accumuler un certain nombre d'encapsulations $(a \rightarrow aa)$, puis de le forcer à passer une et une seule fois par les sommets qui permettent la désencapsulation du protocole a . La contrainte de bande passante assure que chaque lien est pris au plus une seule fois. La Figure 2 illustre cette réduction. Comme l'existence d'un chemin hamiltonien dans \mathcal{H} est un problème NP-complet, celui de l'existence d'un chemin faisable dans \mathcal{N} est NP-difficile. \square

Concernant l'appartenance du problème à NP (et donc sa NP-complétude), la question reste ouverte. En effet, même si un chemin faisable semble constituer un certificat polynomial pour le problème, il n'est pas encore prouvé que le chemin faisable le plus court a un nombre polynomial de liens.

Références

- [HMu06] J. E. Hopcroft, R. Motwani, and J. D. Ullman. From PDA's to Grammars. In *Introduction to Automata Theory, Languages, and Computation*, chapter 6.3.2, pages 247–251. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.
- [KD09] F. A. Kuipers and F. Dijkstra. Path selection in multi-layer networks. *Computer Communications*, 2009.
- [Knu77] D. E. Knuth. A Generalization of Dijkstra's Algorithm. *Inf. Process. Lett.*, 6(1) :1–5, 1977.
- [LFCP16] M. L. Lamali, N. Fergani, J. Cohen, and H. Pouyllau. Path computation in multi-layer networks : Complexity and algorithms. In *INFOCOM 2016, San Francisco, CA, USA, April 10-14, 2016*, pages 1–9, 2016.
- [LPB13] M. L. Lamali, H. Pouyllau, and D. Barth. Path computation in multi-layer multi-domain networks : A language theoretic approach. *Computer Communications*, 36(5) :589–599, 2013.

[§]. Nous définissons un graphe orienté symétrique comme étant un graphe où un lien (U, V) existe si et seulement si le lien (V, U) existe également. La plupart des réseaux de télécommunication sont symétriques dans ce sens.