



HAL
open science

Algorithmes de placement de VNFs dans des contextes mono-et multi-proprétaire

Cedric Morin, Géraldine Texier, Christelle Caillouet, Gilles Desmangles,
Cao-Thanh Phan

► **To cite this version:**

Cedric Morin, Géraldine Texier, Christelle Caillouet, Gilles Desmangles, Cao-Thanh Phan. Algorithmes de placement de VNFs dans des contextes mono-et multi-proprétaire. ALGOTEL 2020 – 22èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Sep 2020, Lyon, France. hal-02873944

HAL Id: hal-02873944

<https://hal.science/hal-02873944v1>

Submitted on 18 Jun 2020

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.

Algorithmes de placement de VNFs dans des contextes mono- et multi-propritaire

Cédric Morin¹²³ et Geraldine Texier¹² et Christelle Caillouet⁴ et Gilles Desmangles³ et Cao-Thanh Phan²

¹*IMT Atlantique/IRISA/Adopnet, France*

²*BCOM, France*

³*TDF, France*

⁴*Université Côte d'Azur/13S/CNRS/Inria, France*

Le concept de Network Functions Virtualisation (NFV) offre aux opérateurs réseau la possibilité de fournir des services évolutifs et moins onéreux, libérés des contraintes propres aux composants matériels. Dans les réseaux 5G, les fonctions ne doivent pas seulement être déployées dans de vastes datacenters centraux, mais également dans les réseaux edge. Nous proposons un algorithme permettant de résoudre le problème de placement de chaînes de fonctions virtualisées en mettant l'accent sur la préservation de ces ressources rares pour répondre à un maximum de demandes. Lorsque le réseau est divisé entre plusieurs acteurs différents, réticents à partager les détails de leurs infrastructures, nous proposons une heuristique permettant à l'orchestrateur de placer les chaînes de fonctions virtualisées en se basant uniquement sur une vision abstraite des topologies. Nous montrons que cette approche peut également servir à réduire la complexité du problème dans les grands réseaux, qu'ils soient divisés entre plusieurs acteurs ou non. Nous analysons l'efficacité de notre algorithme et de l'heuristique sur un large éventail de paramètres et de topologies.

Mots-clefs : optimisation, placement de VNF, architecture multi-propritaire, architecture mono-propritaire

1 Introduction

Le concept de NFV propose de transformer les fonctions réseau habituellement implémentées par un matériel dédié en fonctions logicielles appelées Virtual Network Functions (VNFs) pouvant être exécutées sur des serveurs génériques. Dans l'architecture Management and Orchestration (MANO) [ETS14] de l'ETSI, le Network Functions Virtualisation Orchestrator (NFVO) doit placer une chaîne de VNFs (à la réception d'une demande de création de Network Service (NS)) en se basant sur les topologies fournies par les Virtualized Infrastructure Managers (VIMs). Nous traitons ici ce problème de placement, appelé Virtual Network Function Chain Placement Problem (VNF CPP). Nous nous intéressons d'abord au cas mono-propritaire, où les VIMs appartiennent à la même entité que le NFVO et lui transmettent leurs topologies en détail, puis au cas multi-propritaire, où le réseau est divisé en un ensemble de VIMs indépendants peu enclins à exposer leurs infrastructures internes. Cet article est un résumé de [MTC⁺19].

2 Placement dans une architecture mono-propritaire

L'objectif est de placer un maximum de requêtes à la volée, sans connaître les besoins des suivantes. Afin d'augmenter la probabilité de pouvoir servir les futures requêtes, notre stratégie épargne les ressources partout dans le réseau. Le but est d'éviter le rejet de NSs fortement contraints, par exemple ceux issus des nouveaux cas d'usage de la 5G, par manque de ressources atteignables dans le Edge. Nous faisons l'hypothèse que ces requêtes fortement contraintes sont majoritairement de petites tailles. De très grosse requêtes requérant des ressources Edge peuvent survenir, mais coûteraient cher à leur commanditaire. Ainsi nous associons à chaque ressource un

prix égal à l'inverse de la quantité disponible de cette ressource sur l'équipement considéré. Nous décrivons ce système sous forme d'un Integer Linear Programming (ILP) en utilisant les notations de la Table 1. En sortie le modèle attribue des valeurs aux ensembles de variables x et y . $x_{i,j}^{v,w}$ vaut 1 si le trafic entre $(v,w) \in V^2$ emprunte le lien de $(i,j) \in N^2$, 0 sinon. y_i^v vaut 1 si $v \in V$ est installée sur $i \in N$, 0 sinon. On suppose ici que l'ordre des VNFs dans la chaîne est strict.

Table 1: Notations

Nom	Description	Name	Description
V	VNFs à placer	N	Nœuds dans le réseau
$B^{v,w}$	Bande passante requise entre $(v,w) \in V^2$	$B_{i,j}$	Bande passante disponible sur le lien $(i,j) \in N^2$
S^v	RAM requise par $v \in V$	S_i	RAM disponible en $i \in N$
C^v	CPU requis par $v \in V$	C_i	CPU disponible en $i \in N$
T^v	Délai de traitement de $v \in V$	$D_{i,j}$	Délai du lien $(i,j) \in N^2$
Δ	Délai max de bout en bout	$D^{v,w}$	Délai max entre $(v,w) \in V^2$

La fonction objectif (1) est obtenue par scalarisation : nous sommes les prix des trois ressources consommables considérées (bande passante, RAM et CPU) et attribuons à chacun un coefficient (resp. α , β et γ). Le but est de minimiser cette somme. Les entrées et sorties de la chaîne sont représentées par deux VNFs fictives (*entryVNF* et *exitVNF*) sans contraintes de ressources, qui nous permettent de fixer les points d'entrée/sortie en imposant $y_{entryNode}^{entryVNF} = 1$ et $y_{exitNode}^{exitVNF} = 1$. (2) et (3) assurent que la chaîne est continue et ne contient pas de cycle entre deux VNFs. L'installation des VNFs est garantie par (4). Les capacités des éléments du réseau en termes de bande passante, CPU et RAM ne doivent pas être excédées (resp. (5), (6) et (7)) et les contraintes de délais doivent être respectées, aussi bien entre deux VNFs (8) que de bout en bout (9).

$$\min \alpha * \sum_{(v,w) \in V} \sum_{(i,j) \in N} x_{i,j}^{v,w} * \frac{B^{v,w}}{B_{i,j}} + \beta * \sum_i \sum_v y_i^v * \frac{S^v}{S_i} + \gamma * \sum_i \sum_v y_i^v * \frac{C^v}{C_i} \quad (1)$$

$$\sum_{j \in N} x_{j,i}^{v,w} - \sum_{j \in N} x_{i,j}^{v,w} + y_i^v - y_i^w = 0, \forall i \in N, \forall (v,w) \in V \quad (2)$$

$$\sum_{j \in N} x_{j,i}^{v,w} + y_i^v \leq 1, \forall i \in N, \forall (v,w) \in V \quad (3) \quad \sum_{i \in N} y_i^v = 1, \forall v \in V \quad (4)$$

$$\sum_{(v,w) \in V} x_{i,j}^{v,w} * B^{v,w} \leq B_{i,j}, \forall (i,j) \in N \quad (5) \quad \sum_{v \in V} y_i^v * C^v \leq C_i, \forall i \in N \quad (6)$$

$$\sum_{v \in V} y_i^v * S^v \leq S_i, \forall i \in N \quad (7) \quad \sum_{(i,j) \in N} x_{i,j}^{v,w} * D_{i,j} \leq D^{v,w}, \forall (v,w) \in V \quad (8)$$

$$\sum_{(v,w) \in V} \sum_{(i,j) \in N} x_{i,j}^{v,w} * D_{i,j} + \sum_{v \in V} \sum_{i \in N} y_i^v * T^v \leq \Delta \quad (9)$$

3 Placement dans une architecture multi-propriétaire

Le placement présenté Section 2 suppose une vision totale des ressources du réseau, ce qui n'est pas nécessairement le cas dans un contexte multi-propriétaire. Afin d'exposer leurs capacités sans pour autant révéler toute leur topologie les VIMs peuvent recourir à un mécanisme d'abstraction. Nous proposons donc un système en deux étapes formant notre heuristique : le NFVO place d'abord sa chaîne sur les ressources abstraites des VIMs, puis chaque VIM concerné remplace la portion de chaîne qui lui est allouée sur sa topologie interne, représentant un cluster indépendant. Les deux placements utilisent l'algorithme présenté précédemment. L'abstraction topologique est un sujet ancien dans le domaine des réseaux, qui a déjà motivé de nombreux travaux [ULNB07]. Cependant ces méthodes ne prennent pas en compte l'abstraction des ressources des nœuds (non considérées dans les réseaux traditionnels). Nous proposons une abstraction en étoile de chaque cluster. Le noyau est l'image du nœud ayant la plus grande centralité dans le cluster considéré. Il est relié par un lien direct à chaque nœud de bordure, eux-mêmes connectés aux clusters voisins. Pour chaque métrique de lien un algorithme de Dijkstra est effectué entre le nœud de bordure et le noyau dans

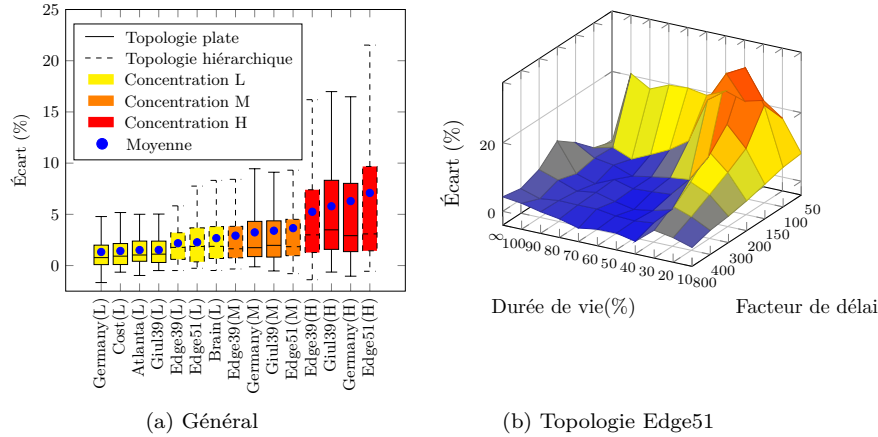


Figure 1: Performances de l'ILP contre l'algorithme de référence dédié à la bande passante

le cluster, et le meilleur score sert de métrique dans l'étoile. Chaque nœud de bordure conserve ses ressources, et le noyau agrège toutes les autres. Dans la fonction objectif cependant, le noyau affiche autant de ressource que le nœud réel qui en possède le plus, afin de ne pas le rendre trop attractif comparé aux nœuds de bordure. Bien que simple, cette méthode d'abstraction - adaptée ici pour prendre en compte les ressources des nœuds - a déjà démontré son efficacité [KK00]. On peut également appliquer cette heuristique au cas mono-proprétaire pour accélérer la résolution du placement, le NFVO réalisant alors en interne une découpe artificielle du réseau en clusters.

4 Simulations et résultats

Nous évaluons notre solution avec le solveur Gurobi et 12 cœurs logiques Intel Xeon E5-2630, en commençant par le cas mono-proprétaire. L'algorithme est testé en faisant varier de multiples paramètres. Nous considérons plusieurs topologies qui peuvent être réparties en deux familles : plates[†] et hiérarchiques sur 3 niveaux (Edge39, Edge51). Pour les nœuds, nous considérons 3 niveaux de concentration de ressources : L (low), M (medium) et H (high). En L, tous les nœuds ont les mêmes ressources (80 CPU, 100 RAM), en M les nœuds centraux en ont d'avantage que les nœuds des extrémités, et en H cette différence est accentuée. La quantité totale de ressources dans le réseau reste toujours la même. En ce qui concerne les requêtes, leur tolérance au délai est fonction d'un paramètre nommé facteur de délai. Plus celui-ci est faible plus la tolérance est faible. Chaque NS possède également une durée de vie indiquant le temps qu'il va passer dans le réseau. Si les NS ont une durée de vie assez longue on observe que le réseau va atteindre un point d'inflexion marquant une chute brutale du taux d'acceptation des nouvelles requêtes (saturation). La durée du vie d'un NS est exprimée en pourcentage par rapport à ce point d'inflexion. Plus le pourcentage est élevé, plus le réseau sera saturé. Enfin plusieurs paramètres sont tirés aléatoirement : les chaînes contiennent entre 2 et 5 VNFs, les VNFs consomment entre 0 et 10 unités de chaque type de ressource, les connexions consomment entre 1 et 10 unités de bande passante. Afin de faciliter la lecture des résultats nous ne conservons que les requêtes pouvant être placées si le réseau est vide. Les coefficients α , β et γ sont déterminés empiriquement et valent respectivement 1, 4 et 7.

Nous comparons notre algorithme avec une solution qui ne cherche qu'à optimiser la consommation de bande passante (mise en oeuvre par notre algorithme avec les paramètres $\alpha = 1$, $\beta = 0$ et $\gamma = 0$). Les résultats présentés Figure 1 affichent l'écart en nombre de requêtes acceptées entre notre algorithme et cet algorithme de référence. La Figure 1a montre que la performance ne dépend pas de la forme de la topologie, mais de la concentration des ressources : lorsque les ressources sont concentrées, ce qui est le cas dans les réseaux réels, notre algorithme affiche de meilleurs résultats. Cependant, même dans cette configuration favorable de forts écarts sont observés. Pour les com-

[†]. Ces architectures peuvent être trouvées à l'adresse <http://sndlib.zib.de>

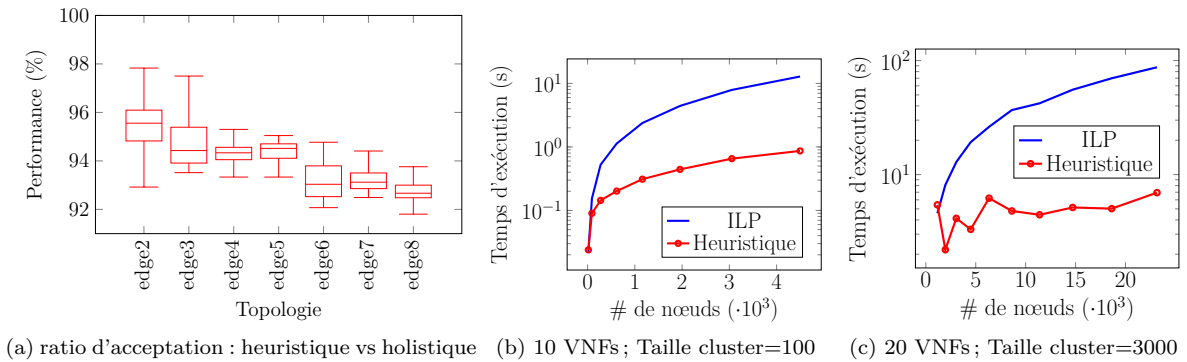


Figure 2: Performances de l'heuristique contre l'holistique

prendre la Figure 1b détaille les résultats sur une topologie. On constate que notre algorithme est très performant pour traiter des requêtes présentant des contraintes de délai assez fortes sur un réseau moyennement chargé. Ceci correspond à notre cas d'usage impliquant des requêtes parfois fortement contraintes dans des réseaux correctement dimensionnés (ni sous utilisés, ni surchargés).

Nous analysons ensuite Figure 2 les performances de l'heuristique contre l'algorithme holistique (*i.e.* : sans processus d'abstraction). La Figure 2a présente la quantité de requêtes acceptées entre la première requête et le point d'inflexion, où chaque VIM gère un cluster contenant au plus 100 noeuds qu'il abstrait suivant le processus décrit Section 3 avant de le présenter au NFVO, dans des topologies de type hiérarchique de plus en plus grandes. On constate que l'heuristique affiche de bonne performances en acceptant systématiquement plus de 90% des requêtes. Au niveau du temps d'exécution, on constate Figure 2b que l'heuristique est beaucoup moins sensible à l'augmentation de taille du réseau que l'algorithme holistique. Cet aspect est amplifié lorsque la taille du réseau, de la chaîne de VNF à placer et des clusters augmente (Figure 2c).

5 Conclusion

Afin de résoudre le problème du placement de chaînes de fonctions réseau virtualisées nous avons d'abord proposé, dans le cadre mono-propritaire, un algorithme visant à satisfaire un nombre maximum de requêtes via une gestion fine des ressources, puis une heuristique permettant au VIM d'exposer une vue abstraite de son infrastructure pour permettre le placement dans un contexte multi-propritaire. Pour les travaux futurs, nous prévoyons de construire l'abstraction de manière récursive et d'étudier la ré-optimisation hors ligne du placement et la mutualisation de VNFs.

Références

- [ETS14] ETSI GS NFV-MAN 001. Network Functions Virtualisation (NFV); Management and Orchestration. 12 2014. Version 1.1.1.
- [Fre77] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1) :35–41, March 1977.
- [KK00] Turgay Korkmaz and Marwan Krunz. Source-oriented topology aggregation with multiple QoS parameters in hierarchical networks. *ACM Transactions on Modeling and Computer Simulation*, 10(4) :295–325, October 2000.
- [MTC⁺19] Cédric Morin, Geraldine Texier, Christelle Caillouet, Gilles Desmangles, and Cao-Thanh Phan. VNF placement algorithms to address the mono- and multi-tenant issues in edge and core networks. In *8th IEEE International Conference on Cloud Networking (IEEE CloudNet 2019)*, pages 1–6. IEEE, 2019.
- [ULNB07] Suleyman Uludag, King-Shan Lui, Klara Nahrstedt, and Gregory Brewster. Analysis of topology aggregation techniques for QoS routing. *ACM Computing Surveys (CSUR)*, 39(3) :7, 2007.