Optimisation pour le Provisionnement de Chaînes de Services Réseau

Nicolas Huin, Brigitte Jaumard, Frédéric Giroire

Optimisation pour le Provisionnement de Chaînes de Services Réseau

Nicolas Huin[1] et Brigitte Jaumard[1] et Frédéric Giroire[2]

[1] *Computer Science and Software Engineering, Concordia University, Montreal (QC) Canada*
[2] *Université Côte d'Azur, CNRS, Sophia Antipolis, France*

La virtualisation des réseaux, portée en partie par l'initiative virtualisation des fonctions de réseau (Network Function Virtualization - NFV), apporte une plus grande flexibilité aux opérateurs de réseaux ainsi qu'une baisse des coûts de gestion. Les fonctions réseau exécutées sur du matériel dédié peuvent maintenant être exécutées par des serveurs virtuels. Un service correspond alors à une séquence de fonctions, appelée chaîne de services, à effectuer dans un ordre précis sur l'ensemble des flots du service. Le problème considéré est alors de satisfaire les requêtes des chaînes de services tout en trouvant le meilleur compromis entre l'utilisation de la bande passante et le nombre d'emplacements pour héberger les fonctions réseau.

Nous proposons un modèle de génération de colonnes pour le routage et le placement de chaînes de service. Nous montrons à l'aide d'expérimentations poussées que nous pouvons résoudre le problème de façon optimale en moins d'une minute pour des réseaux ayant une taille allant jusqu'à 65 nœuds et 16000 requêtes. Nous étudions aussi le compromis entre l'utilisation de la bande passante et le nombre de nœuds capables d'héberger des fonctions réseau.

1 Introduction

Ces dernières années, les opérateurs de réseaux de télécommunications ont vu l'arrivée de nouveaux paradigmes promettant de faciliter la gestion de leurs réseaux. Parmi ces paradigmes, l'initiative NFV (Network Function Virtualization) vise à virtualiser les fonctions réseau. En effet, dans les réseaux actuels, les fonctions réseau telles que les pare-feu, les inspecteurs de paquets ou les optimiseurs vidéos sont effectués grâce à du matériel dédié, propriétaires et fermés, qui souffre de coûts opérationnels importants et d'un manque de flexibilité. En virtualisant les fonctions réseaux, les opérateurs peuvent déployer les fonctions sur du matériel générique en utilisant des serveurs virtuels, permettant ainsi une plus grande flexibilité dans la gestion et la création de nouveaux services pour les clients.

Dans un réseau, chaque service est associé à un ensemble de fonctions qui sont appliquées aux requêtes utilisant ce service. Lorsque les fonctions doivent être appliquées dans un ordre spécifique, on parle alors de *Chaîne de Services*. Par exemple, un service de voix sur IP peut demander l'exécution d'un pare-feu suivi d'un contrôle d'accès. Dans cet article, nous considérons le problème du *Placement de Chaînes* qui consiste à placer les fonctions sur le réseau de sorte à ce que les flots puissent parcourir les fonctions correspondantes à leur service dans le bon ordre. Plusieurs variantes du problème existent avec des objectifs tels que la minimisation du nombre de nœuds pouvant héberger des fonctions [RBR$^+$15] ou bien le nombre de multiplications/copies de fonctions [LBB$^+$15] mais nous considérons ici la minimisation de la bande passante. À notre connaissance, nous sommes les premiers à proposer *une résolution optimale du problème grâce à un modèle de génération de colonnes sur des jeux de données de moyennes et grandes tailles avec un trafic de type tout-à-tout*. Les précédents travaux proposent soit des heuristiques basées sur des formulations mathématiques [GHC$^+$15, RBR$^+$15], soit des formulations exactes [LBB$^+$15, GJTM18] mais qui ne résolvent que des petits jeux de données du problème. Finalement, nous considérons aussi un scénario "statique" où toutes les requêtes sont approvisionnées en même temps. Ce type de scénario est très utile lors de la reconfiguration ou la défragmentation d'un réseau ou lors de l'évaluation de la qualité de solutions associées à des scénarios dynamiques.

Nous présentons formellement le problème, le graphe en couches nécessaire pour la résolution ainsi que le modèle de génération de colonnes dans la Section 2. Nous étudions dans la Section 3 la performance de notre modèle, puis nous utilisons notre modèle pour étudier le compromis entre le nombre de nœuds capable d'héberger des fonctions et la bande passante requise.

2 Modèle

Le réseau est représenté par un graphe $G = (V, L)$ où V représente l'ensemble des nœuds du réseau et L l'ensemble des liens. L'ensemble des requêtes est donné par $\mathcal{SD}$, et elles sont caractérisées par une source v_s, une destination v_d, une chaîne de fonctions c et une bande passante requise D^c_{sd}. Soit F l'ensemble des fonctions réseau virtuelles (VNFs). Chaque chaîne de service c est définie par une séquence de n_c fonctions virtuelles qui peuvent être répétées. C note l'ensemble des chaînes. Le nombre de cœurs CPU requis par une fonction f par unité de bande passante est égal à Δ_f. Seul un sous-ensemble de nœuds $V^{\text{VNF}} \subseteq V$ peut héberger des fonctions virtuelles †. Chaque nœud $v \in V^{\text{VNF}}$ possède une capacité en nombre de cœurs CPU, noté CAP_v, qui sont utilisés par les fonctions virtuelles exécutées dessus. Chaque lien est limité par sa bande passante maximale, notée CAP_ℓ. *L'objectif est de minimiser la bande passante utilisée dans le réseau tout en satisfaisant toutes les requêtes et en respectant les capacités des liens et des nœuds.* Chaque requête doit être affectée à un chemin sur lesquels les fonctions du service correspondant sont placées dans le bon ordre.

Graphe en couches. Sur une idée similaire à [DW16], nous modélisons le problème grâce à un graphe à couche. Ce graphe est essentiel pour résoudre efficacement le modèle de génération de colonnes proposé. Le graphe G est transformé en un graphe G^L avec $\max_{c \in C} n_c + 1$ couches. Chaque couche est une copie exacte de G et pour chaque nœud v du réseau, on note v^i le nœud correspondant à la couche i. Les couches $i-1$ et i sont alors connectées par des liens (v^{i-1}, v^i). Trouver un chemin et un placement de fonctions pour la requête (v_s, v_d, c, D^c_{sd}) est alors équivalent à trouver un chemin sur G^L du sommet v_s sur la couche initiale (v^0_s) au sommet v_d sur la n_c-ième couche $(v^{n_c}_d)$. En effet, chaque couche représente la progression de la chaîne, c.à.d., le premier nœud rencontré sur la troisième couche indique le lieu d'exécution de la troisième fonction. L'emplacement d'une fonction est donnée par les arcs entre les couches utilisées par le chemin.

Modèle de génération de colonnes. Nous présentons maintenant un modèle de génération de colonnes, appelé NFV_CG. Ce modèle se base sur le concept de *Chemin de Services* qui représente un chemin potentiel pour une requête. Nous introduisons les notations suivantes pour la formulation du modèle.

- $p \in P^c_{sd}$, un *Chemin de Services* pour la requête (v_s, v_d, c, D^c_{sd}). Il est composé d'un chemin et d'un ensemble de paires $\{(v, f) : f \in c\}$, où (v, f) indique que la fonction f est installée sur le nœud v.
- $a^p_{f^c_i v} \in \{0, 1\}$, où $a^p_{f^c_i v} = 1$ si la $i$éme fonction de c, notée f^i_c est exécutée sur le noeud v pour le *Chemin de Services* $p \in P^c_{sd}$.
- $\delta^p_\ell \in \mathbb{N}^+$, où δ^p_ℓ représente le nombre d'occurrences du lien ℓ dans le chemin p.

Observer que les chemins ne sont pas nécessairement simples à cause des contraintes induites par l'ordre dans lequel les fonctions doivent être exécutées. Puisque la formulation de NFV_CG possède un nombre exponentiel de variables, nous utilisons la génération de colonnes afin de ne considérer qu'un sous-ensemble de *Chemins de Services*. La résolution par génération de colonnes combine un *Problème Maître Réduit (PMR)*, c.à.d le *Problème Maître* avec un sous-ensemble des variables, ainsi qu'un *Problème auxiliaire, appelé "Pricing" (PP)*. Le PMR et le PP sont résolus en alternance jusqu'à ce que le PP ne puisse plus générer de chemins de services qui améliorent la solution optimale de la relaxation continue du PMR. Dans ce cas, la solution courante de la relaxation continue du PMR est optimale et une solution entière peut en être dérivée en introduisant des contraintes d'intégralité. Le *Problème Maître* est formulé de la façon suivante :

Variables : $y^{sd,c}_p \geq 0$, où $y^{sd,c}_p = 1$ si la requête (v_s, v_d, c, D^c_{sd}) est routée sur le chemin de service p, 0 sinon.

$$\textbf{Objectif :} \qquad \min \sum_{(v_s,v_d) \in \mathcal{SD}} \sum_{c \in C_{sd}} \sum_{p \in P^c_{sd}} \left(D^c_{sd} \sum_{\ell \in L} \delta^p_\ell \right) \times y^{sd,c}_p \tag{1}$$

†. Dans la suite, ces nœuds sont appelés nœuds NFV

Service	Chaîne	Débit	% trafic
Web Service	NAT-FW-TM-WOC-IDPS	100 kB s^{-1}	18.2%
VoIP	NAT-FW-TM-FW-NAT	64 kB s^{-1}	11.8%
Video Streaming	NAT-FW-TM-VOC-IDPS	4 MB s^{-1}	69.9%
Online Gaming	NAT-FW-VOC-WOC-IDPS	50 kB s^{-1}	0.1%

TABLE 1: Chaîne de Service [STV15]

Réseau	# req.	# nœuds NFV	# col. gén.	$z^\star_{\text{LP}}$	$\bar{z}_{\text{ILP}}$	ε	Temps (ms)
internet2	360	5	440	2086.7	2086.7	0	22
		6	416	2075.4	2075.4	0	23
atlanta	840	7	1024	2625.1	2625.1	0	44
		8	944	2596.1	2596.1	0	36
germany50	9800	24	25766	4217.6	4218.0	7.82×10^{-5}	7354
		25	24381	4211.9	4212.3	9.01×10^{-5}	6463
ta2	16640	33	48819	4231.6	4231.8	5.20×10^{-5}	29681
		34	45212	4233.2	4232.8	8.74×10^{-5}	45212

TABLE 2: Résultats obtenus avec NFV_CG

$$\textbf{Contraintes :} \quad \sum_{p\in P^c_{sd}} y^{sd,c}_p = 1 \qquad c\in C_{sd}, (v_s,v_d)\in\mathcal{SD} \qquad (2)$$

$$\sum_{c\in C_{sd}} \sum_{(v_s,v_d)\in\mathcal{SD}} \sum_{p\in P^c_{sd}} \left(D^c_{sd}\,\delta^p_\ell\right) \times y^{sd,c}_p \le \text{CAP}_\ell \qquad \ell\in L \qquad (3)$$

$$\sum_{c\in C_{sd}} \sum_{(v_s,v_d)\in\mathcal{SD}} \sum_{p\in P^c_{sd}} \left(\sum_{i=0}^{n_c} D^c_{sd}\,\Delta_{f^i_c}\,a^{f^i_c}_{vp}\right) \times y^{sd,c}_p \le \text{CAP}_v \qquad v\in V^{\text{NFV}}. \qquad (4)$$

L'objectif est de minimiser la bande passante utilisée sur le réseau, ce qui correspond à minimiser la longueur des chemins utilisés multipliée par le débit de la requête. Les contraintes (2) assurent qu'exactement un chemin de service est choisi pour chaque requête. Les capacités de liens et de nœuds sont respectées grâce aux contraintes (3) et (4), respectivement.

Le Problème auxiliaire consiste à trouver un *Chemin de Services* pour une requête qui minimise le coût réduit de la variable correspondante dans le PMR. Le coût réduit est calculé en utilisant les valeurs duales $u^{(j)}$ du PMR, où $u^{(j)}$ représente le vecteur des valeurs duales correspondant aux contraintes (j) du PMR. Le coût réduit d'un *Chemin de Services* pour la requête (v_s, v_d, c, D^c_{sd}) est donné par

$$-u^{(2)} + \sum_{\ell\in L}\sum_{i=0}^{n_c} D^c_{sd}\,\varphi_{i\ell}\left(1+u^{(3)}_\ell\right) + D^c_{sd}\sum_{v\in V} u^{(4)}_v \sum_{i=0}^{n_c}\Delta_{f^i_c}\,\alpha_{iv} \qquad (5)$$

où $\varphi_{i\ell} = 1$ si la requête est approvisionnée sur le lien ℓ au niveau i et $\alpha_{iv} = 1$ si la $i$éme fonction de c, notée f^i_c, est exécutée sur le noeud v.

Puisque $u^{(2)}$ est une constante pour chaque requête (v_s, v_d, c, D^c_{sd}), trouver un *Chemin de Service* est équivalent à trouver un plus court chemin entre v^0_s et $v^{n_c}_d$ dans G_L. Le poids d'un lien (u^i, v^i) entre deux noeuds d'un même niveau est égale à $1+u^{(3)}_\ell$ et le poids d'un lien (v^i, v^{i+1}) entre deux niveaux est égal à $u^{(4)}_v \times \Delta_{f^i_c}$. Puisque les valeurs duales $u^{(3)}_\ell$ et $u^{(4)}_v$ sont positives, ce plus court chemin peut être trouvé avec l'algorithme de Dijkstra.

3 Résultats

Dans cette section, nous présentons les résultats obtenus avec NFV_CG. Premièrement, nous décrivons le jeu de données utilisé. Nous présentons ensuite les performances de NFV_CG, suivies d'une étude sur le compromis entre le nombre de nœuds NFV et la bande passante requise ainsi que le nombre de sauts.

Jeux de données. Afin d'émuler un trafic réaliste, nous avons extrait du rapport Cisco annuel la distribution du trafic Internet. En utilisant le débit moyen d'un service (voir Table 1), nous pouvons en déduire le nombre de requêtes demandant chaque service. Par exemple, sur une charge totale de 1 TB s^{-1}, le streaming vidéo représente 699 GB s^{-1} ce qui correspond à environ 175k requêtes différentes. Les sources et destinations sont alors choisies de manière uniformément aléatoire et les requêtes possédant la même source, destination et chaînes sont alors agrégées. Nous avons généré nos requêtes sur 4 réseaux différents `internet2`, `atlanta`, `germany50` et `ta2`. Lors de l'étude du compromis entre le nombre de nœuds et l'utilisation de la bande passante, nous sélectionnons les nœuds selon sur leur *betweeness centrality*.

Performance de NFV_CG. La Table 2 montre les résultats obtenus avec NFV_CG sur les quatres réseaux lorsque le réseau possède aux alentours de 50% de nœuds NFV. Le trafic total de chaque instance est 1 Tbps.

Les quatre dernières colonnes représentent la valeur optimale de la relaxation continue du MPR ($z^{\star}_{\text{LP}}$), la valeur optimale de la solution entière ($\tilde{z}_{\text{ILP}}$), le ratio entre les deux, ε et le temps de calcul. Nous observons que $\varepsilon = 0$ dans la plupart des instances, ce qui signifie que la solution entière est optimale. Lorsque $\varepsilon > 0$, le ratio reste petit : $\tilde{z}_{\text{ILP}}$ est très proche de la valeur optimale entière. De plus, les temps de calcul restent très faibles, même pour les plus gros jeux de données (environ 45 s).

Compromis entre bande passante requise et nombre de nœuds NFV. Nous étudions ici le compromis entre le nombre de nœuds NFV présents sur le réseau et la bande passante totale requise. Par manque de place, nous ne présentons que les résultats sur `ta2` pour un trafic total de 1 Tbps. Les mêmes phénomènes peuvent être observés pour `internet2`, `atlanta` et `germany50` [HJG17].

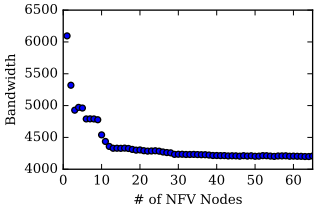

FIGURE 1: Bande passante

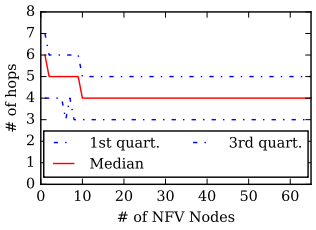

FIGURE 2: Longueurs des chemins en nombre de sauts

La Figure 1 montre la bande passante en fonction du nombre de noeuds NFV. Sans surprise, la bande passante diminue lorsque le nombre de nœud NFV augmente. Comme chaque requête nécessite une chaîne, le chemin doit alors passer par les nœuds NFV dans le bon ordre, ce qui peut allonger les chemins de façon significative. Cependant, à partir de 50% de noeuds NVF, le gain de bande passante devient beaucoup plus faible.

Dans la Figure 2, nous montrons la distribution du nombre de sauts en fonction du nombre de nœuds NFV. Nous observons que la valeur médiane du nombre de sauts se stabilise à partir de 9 nœuds NVF dans le réseau. Bien que le gain en bande passante se stabilise plus tard, nous pouvons observer qu'une faible portion des requêtes est affectée lorsque le nombre de nœuds est supérieur à 10.

4 Conclusion

Dans cet article, nous proposons un modèle de génération de colonnes pour le problème de Placement de Chaînes de Services Réseau. Nous montrons que notre modèle peut résoudre de façon exacte le problème sur des jeux de données allant jusqu'à 65 nœuds et 16 000 en un temps de calcul acceptable. Nous avons aussi étudié le compromis entre le nombre de nœuds NFV et la bande passante requise. Nous avons montré que les gains de bande passante devenaient faibles lorsque le réseau possédait plus de 50% de nœuds NFV.

Références

[DW16] A. Dwaraki and T. Wolf. Adaptive service-chain routing for virtual network functions in software-defined networks. In *Workshop on Hot topics in Middleboxes and Network Function Virtualization (HotMIddlebox)*, pages 32–37, 2016.

[GHC+15] A. Gupta, M.F. Habib, P. Chowdhury, M. Tornatore, and B. Mukherjee. Joint virtual network function placement and routing of traffic in operator networks. Technical report, UC Davis, Davis, CA, USA, 2015.

[GJTM18] A. Gupta, B. Jaumard, M. Tornatore, and B. Mukherjee. A scalable approach for service chain (SC) mapping with multiple sc instances in a wide-area network. *JSAC*, 2018.

[HJG17] N. Huin, B. Jaumard, and F. Giroire. Optimization of Network Service Chain Provisioning. In *IEEE International Conference on Communications 2017*, Paris, France, May 2017.

[LBB+15] M.C. Luizelli, L.R. Bays, L.S. Buriol, M.P. Barcellos, and L.P. Gaspary. Piecing together the NFV provisioning puzzle : Efficient placement and chaining of virtual network functions. In *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 98–106, 2015.

[RBR+15] R. Riggio, A. Bradai, T. Rasheed, J. Schulz-Zander, S. Kuklinski, and T. Ahmed. Virtual network functions orchestration in wireless networks. In *International Conference on Network and Service Management (CNSM)*, pages 108–116, 2015.

[STV15] M. Savi, M. Tornatore, and G. Verticale. Impact of processing costs on service chain placement in network functions virtualization. In *IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, pages 191–197, 2015.