

# Une Heuristique de Placement et Chainage de Fonctions de Services composées de Micro-Services

Hichem Magnouche<sup>1</sup>, Caroline Prodhon<sup>1</sup>, Guillaume Doyen<sup>2</sup>

<sup>1</sup>ICD-LIST3N, Université de Technologie de Troyes, Troyes, France

<sup>2</sup>Imt-Atlantique, Rennes, France

{hichem.magnouche, caroline.prodhon}@utt.fr  
guillaume.doyen@imt-atlantique.fr

**Mots-clés :** *Virtualisation des fonctions réseau, Optimisation, Orchestration, Faible latence, Micro-services*

## 1 Introduction

La virtualisation des architectures de réseau présente de nombreux avantages et tend à devenir une préoccupation de plus en plus importante dans le domaine des télécommunications. Elle permet une plus grande flexibilité pour le déploiement des fonctions de réseau (NF), permettant la mise en place de services émergents tels que ceux nécessitant une faible latence. Dans ce domaine, les concepts de micro-services et de programmabilité du réseau constituent des solutions prometteuses mais induisent des problèmes concernant les règles d'orchestration, à savoir le placement et le chaînage. Après avoir passé en revue les problèmes liés à l'optimisation, nous avons proposé un modèle de placement et de chaînage de micro-services ainsi que trois stratégies de cohabitations entre les SFC faible latence et Best Effort nécessitant moins de latence. Faisant face à un temps de résolution extrêmement long nous proposons, dans ce travail, une heuristique permettant le placement et le chaînage de micro-services en un temps relativement court. Cette recherche est menée dans le cadre d'un projet ANR MOSAICO.

## 2 Orchestration de micro-services

Après avoir étudié les solutions de l'état de l'art dans le domaine de l'orchestration de fonction réseaux virtualisées, nous avons proposé une approche qui tire profit des caractéristiques des micro-services afin de déployer des SFC à faible latence [2]. L'approche se compose d'un algorithme de pré-traitement ainsi que d'une modèle mathématique (MILP) de placement et de chaînage. L'algorithme de pré-traitement permet la mutualisation des micro-services mutualisables au sein d'une même SFC, cela a pour effet de réduire leur longueur et par conséquent leur latence. Concernant le MILP, il permet le placement et chaînage des micro-services tout en optimisant leur parallélisation. En effet, une des caractéristiques clés des micro-services, c'est leur aptitude à s'exécuter en parallèle ce qui permet de réduire leur latence d'exécution. Le modèle proposé produit des solutions optimales au bout de quelques heures (en utilisant CPLEX) pour des instances de quelques SFC sur des infrastructures composées entre 5-8 noeuds seulement.

## 3 Heuristique de placement et de chaînage de micro-services

Les évaluations réalisées sur l'approche proposée prouvent un réel gain en terme de latence comparée à une approche monolithique. Néanmoins, éprouver ce modèle sur des instances plus grandes en termes de noeuds et de SFC s'avère impossible car le temps de résolution associé croît de manière exponentielle. Pour pallier cela, nous développons actuellement une

méthode heuristique qui, en s'appuyant sur les caractéristiques de parallélisme des micro-services, produit des solutions proches de l'optimale en un temps polynomial. L'heuristique repose sur l'algorithme d'Eppstein [1] permettant le calcul des k plus courts chemins pour chaque SFC puis déploie les micro-services relatifs à celle-ci sur un des k chemins suivant un ordre croissant. Pour chaque SFC, si le déploiement est réussi, l'algorithme vérifiera le respect de la latence, si celle-ci n'est pas respectée il lancera une phase de parallélisation interne (micro-services déployés sur un même noeud) permettant de réduire sa latence d'exécution. Si une SFC n'est pas déployée en raison de la surcharge des chemins, l'algorithme déploiera la SFC en relâchant la contrainte de capacité sur les noeuds puis lancera une phase de parallélisation externe (micro-services déployés sur des noeuds différents) ce qui permettra de soulager la capacité du chemin surchargé. Dans cette présentation, nous présenterons l'avancée de ce travail à date et les premiers résultats qu'il a produit.

## Références

- [1] D. Eppstein. Finding the k shortest paths. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 154–165, 1994.
- [2] Hichem Magnouche, Guillaume Doyen, and Caroline Prodron. Leveraging micro-services for ultra-low latency : An optimization model for service function chains placement. In *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*, pages 198–206, 2022.