Slicing 5G économe en énergie et garantissant les contraintes de performance

Wei Huang, Andrea Araldo, Hind Castel-Taleb, Badii Jouaber

Slicing 5G économe en énergie et garantissant les contraintes de performance

W. Huang[1] [†]et A. Araldo [1] et H. Castel-Taleb[1] et B. Jouaber [1]

[1] *Telecom SudParis, SAMOVAR, Institut Polytechnique de Paris*

Le placement des slices c'est-à-dire des fonctions et liens virtuels sur un même réseau physique a été largement étudié depuis ces dernières années. Cependant, les performances de bout en bout ne dépendent pas que du placement du slice mais aussi de la quantité de ressources allouées et du trafic des utilisateurs. Nous proposons une méthode de dimensionnement optimal des ressources du réseau physique (CPU et bande passante) basée à la fois sur la résolution d'un problème d'optimisation sous contraintes et les réseaux de Jackson pour la modélisation des slices. Chaque slice est modélisé par un réseau de Jackson, offrant une forme close pour la distribution stationnaire permettant de calculer exactement le temps de réponse de bout en bout et la consommation énergétique. Sur le réseau physique, un ensemble de slices avec des contraintes différentes (mean latency constraints pour le "Service Level Agreement") cohabitent, l'intérêt de notre travail est de générer la quantité optimale de ressources allouées à chaque slice afin de minimiser la consommation énergétique tout en respectant les contraintes, sous la forme de mean latency constraints.

Nous montrons numériquement l'efficacité de notre solution par rapport à d'autres approches, c'est-à-dire sa capacité à allouer exactement la quantité nécessaire de ressources pour minimiser l'énergie consommée et garantir les contraintes de délais.

Mots-clefs : Optimisation, Réseaux de Jackson, Énergie , Slicing 5G, Délais

1 Introduction

The fifth-generation (5G) communications system is envisioned to serve a variety of novel services and industries, such as : autonomous vehicles, Virtual Reality (VR), Augmented Reality (AR) and remote healthcare, each requiring different Quality-of-Service (QoS) constraints. In this context, network slicing is a way to dynamically virtualize resources present in physical network, and partition them into multiple slices. Slices can thus be given to third party Service Providers (SPs) and dimensioned so as to support the different requirements of the different SPs. Different problems arise in the context of the slicing. One important question is how to place slices efficiently on a physical network while guaranteeing both the quality of service and saving the resources used. This problem is known as Virtual Network Embedding (*VNE*) [F$^+$13], which consists in deciding in which physical node we should place virtual component and in which physical path we should place virtual links. Various methods have been studied for this problem, among which many are heuristic algorithms based on Linear programming or Reinforcement Learning (RL) methods [H$^+$17, E$^+$22]. However, many papers in *VNE* typically assume that each component and virtual link "needs" a pre-defined amount of resources. We believe that this assumption is too strong. Indeed, a component can work under different amount of allocated computation capacity : if such a capacity is large, computation will be faster, but we consume more physical resources and therefore power. We suppose in our study that a slice is embedded with a certain amount of resource on each node, which may be ajusted after according to the trafic and the Service Level Agreement (*SLA*). We aim to dimension bandwidth and computation resources while minimizing the dynamic power consumption and guaranteeing the *mean latency constraints*. Our approach consists in modeling network slices as Jackson networks and solving an optimisation problem to minimize dynamic power. The main result is that we are able to compute the amount of resource used

[†]This work was partially funded by Beyond5G, a project of the French Government recovery plan "France Relance".

by several slices hosted on a physical network, so as we minimize the dynamic power consumption of the physical network and we guarantee the constraints on the mean response time for different classes of slices. For more details, we refer to [H+22] where this work has been published.

2 System model

The substrate network (or physical network) is represented by a set of computational nodes, (e.g., servers) interconnected by network links. We model the physical network as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ the set of edges.

— Each node $v \in \mathcal{V}$ has resource capacity R_v, which in our case is CPU cycles.
— We denote with $(v, v') \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ a link and with $R_{v,v'} \geq 0$ its bandwidth capacity (in bits/sec).

Each Service Provider (SP), is represented by a *chain* of n^s *software components* $\{c_i^s\}_{i=1,\dots,n^s}$. Each component can run in a container (or a virtual machine) and are interconnected by virtual links. A virtual link is mapped to a path $\mathcal{P}$ on the physical network. Such a path is a sequence of physical links. When allocating bandwidth to a virtual link, such a bandwidth must be reserved on each link of $\mathcal{P}$. We call "virtual channel" the bandwidth reserved to a slice in each of the links of $\mathcal{P}$. Some nodes are considered as "ingress" because they receive trafic from users. We suppose that a SP denoted by s is characterized by :

— Poissonian user-requests arriving at rate $\lambda_{v,in}^s \geq 0$. The nodes $v \in \mathcal{V}$ for which $\lambda_{v,in}^s > 0$ are *ingress nodes* for s.
— The *Computation complexity* $A_{c_i^s}$ of component c_i^s is the amount of machine-level instructions to be executed by that component at every request. We assume $A_{c_i^s}$ is an exponentially distributed random variable with mean $\alpha_{c_i^s}$. Component c_i^s sends data to c_{i+1}^s. Each component $\{c_i^s\}$ could be replicated on several physical nodes. We denote by $\{c_{i,v}^s\}$ its replica on physical node v.
— The *Communication complexity* $B_{c_i^s, c_{i+1}^s}$ is the amount of data (in bits) sent from c_i^s to c_{i+1}^s at each request. It is also an exponentially distributed random variable with mean $\beta_{c_i^s, c_{i+1}^s}$.

3 Resource dimensioning and mean latency constraints guarantee

We suppose that components and virtual links of all SPs are already placed in the physical networks. This placement is an input to our problem. For any SP s, we define :

— $r_{i,v}^s$ as the computational resources allocated by the Network Operator (NO) to component replica $c_{i,v}^s$.
— $r_{i,v,i+1,v'}^s$ as the bandwidth allocated by the NO for the communication between component replicas $c_{i,v}^s$ and $c_{i+1,v'}^s$. Such a bandwidth is allocated on all the physical links to which virtual link between $c_{i,v}^s$ and $c_{i+1,v'}^s$ is mapped.

Jackson Network Each SP s can be modeled as a feed-forward network of queues $\mathcal{Q}^s$: a request enters from an ingress node and circulates into the network of queues until the execution of the request ends, in the last component. We model each slice as a network of queues. In general, for a queue q we can write the service rate as $\mu_q(r_q) = \frac{r_q}{\alpha_q}$ where :

— $r_q = r_{i,v}^s$ and $\alpha_q = \alpha_{c_i,v}^s$ if q represents component replica $c_{i,v}^s$
— $r_q = r_{i,v,i+1,v'}^s$ and $\alpha_q = \beta_{i,v,i+1,v'}^s$ if q represents the virtual channel reserved for the communication between $c_{i,v}^s$ and $c_{i+1,v'}^s$ in any link of the path between (v, v').

Observe that several networks of queues co-exist in the same physical network. The networks of queues corresponding to the two SPs are represented in Fig. 1. Then we can derive the closed formula of requests' mean journey time T^s from Jackson network models as $T^s\left(\{r_q\}_{q\in\mathcal{Q}^s}\right) = \frac{1}{\sum_{v\in\mathcal{V}} \lambda_{v,in}^s} \cdot \sum_{q\in\mathcal{Q}^s} \frac{\lambda_q}{\mu_q(r_q)-\lambda_q}$.

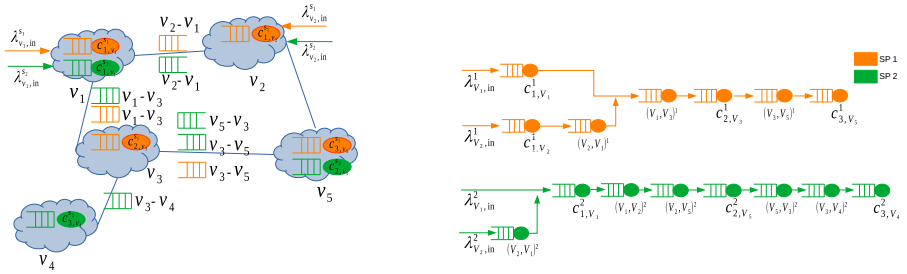


FIGURE 1: SP 1 and 2 modeled as Jackson networks

Power optimization We propose to optimize the dynamic power consumption of the physical network, while guaranteeing the mean latency constraints of SPs :

$$\text{OptRes} : \min \sum_{v\in\mathcal{V}} \sum_{q\in\mathcal{Q}_v} P_{q,v}(r_q) + \sum_{(u,u')\in\mathcal{E}} P_{u,u'}(r_{u,u'})$$

$$\text{s.t.} \quad \sum_{q\in\mathcal{Q}_v} r_q \leq R_v, \forall v \in \mathcal{V} \qquad \sum_{q\in\mathcal{Q}_{u,u'}} r_q \leq R_{u,u'}, \forall (u,u') \in \mathcal{E} \qquad T^s\left(\{r_q\}_{q\in\mathcal{Q}^s}\right) \leq D^s, \forall \text{SP } s$$

$$\mu_q(r_q) > \lambda_q, \forall q \in \mathcal{Q}^s \qquad r_{u,u'} = \sum_{q\in\mathcal{Q}_{u,u'}} r_q, \forall (u,u') \in \mathcal{E}$$

Where $P_{q,v}(r_q) = \frac{42.29 \cdot r_q}{R_v}$ is the dynamic power consumption at node v for queue q and allocated resource r_q and $P_{u,u'}(r_{u,u'}) = \frac{14.555}{550} \cdot r_{u,u'} + 4.5$ if $r_{u,u'} < 550$Mb/s. Otherwise, $P_{u,u'}(r_{u,u'}) = 19.055 + \frac{r_{u,u'}-550}{10^4}$. This is the dynamic power consumption on physical link (u,u') (where $r_{u,u'}$ is the total allocated bandwidth in the corresponding link).

Our goal is to minimize the dynamic power of the physical network while ensuring mean journey time T^s for each SP does not exceed a certain tolerated time D^s (which is an input to the problem and represents the requirement of SPs). Note that we only focus on minimizing CPU dynamic power consumption, for the reasons that CPU consume highest dynamic power compared to memory and disk in one physical node and CPU static power can be considered a constant which will not change our optimization decisions.

4 Numerical results

We consider the network in Fig. 1 and two SPs, with component replicas distributed as in the figure. We use the scenario parameters of Table 1. Our model is implemented in the Matlab solver and solved with Sequential Quadratic Programming. Each solution is obtained in less than 5 minutes. We compare

Node computational capacity	$R_v = 1285.2 \cdot 10^9$ Instr. per sec., $\forall v \in \mathcal{V}$
Physical link capacity	$R_{u,u'} = 10$ Gb/s, $\forall (u,u') \in \mathcal{E}$
Comput. complexity of SP 2	$\alpha^2_{c_i} = 3 \cdot 10^9$ instr/req, $\forall c_i$
Comm. complexity of SP 2	$\beta^2_{c_i,c_{i+1}} = 85$ Kb/req, $\forall c_i, c_{i+1}$
Complexities of SP 1	$\alpha^1_{c_i} = \frac{1}{2}\alpha^2_{c_i}$ $\beta^1_{c_i} = \frac{1}{2}\beta^2_{c_i}$
Ingress req rate	$\lambda^1_{v,\text{in}} = \lambda^2_{v,\text{in}} = 20$ req/sec, $\forall v \in \mathcal{V}$
Delay constraints of SP 1	$D^1 = 0.1$ sec
Delay constraints of SP 2	$D^2 = 1$ sec

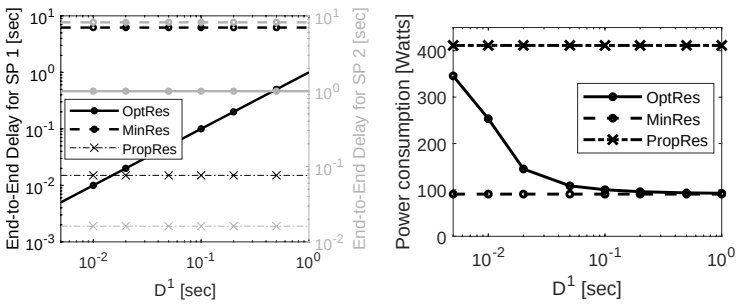
TABLE 1: Scenario parameters.

the performance of the allocation of our approach called *OptRes* with two others, which do not adapt the resource allocation, as common in virtual network embedding (*VNE*) literature :

— *MinRes* : only the minimum resources needed to satisfy stability conditions are allocated ;

— *PropRes* : all physical link and node resources are used, and partitioned among SPs proportional to the computational complexity of the replicas.

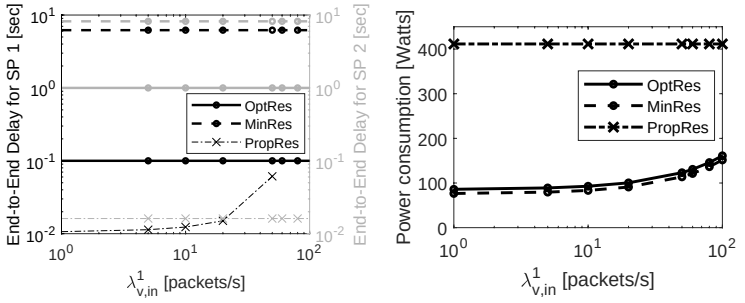
In Fig. 2, we depict the performance of resource allocation with different latency requirements of SP 1 (while SP 2 requirements remain the same). As expected, MinRes consumes the least energy but fails to



(a) Delay (b) Power consumption

FIGURE 2: Delay (SP1 in black and SP2 in grey) and system power used for different latency requirements D^1 of SP 1

meet the latency requirements of both SPs. On the other hand, PropRes is very energy inefficient. OptRes manages instead to satisfy the latency requirements of both SPs, while being energy efficient. Note that, when latency requirements of SP 1 are not stringent, OptRes is as energy efficient as MinRes. Fig. 3 shows with different values of ingress request rate. While OptRes adapts well to different loads, getting practically the same dynamic power consumption of MinRes, and satisfying all latency requirements, also with high load.



(a) Delay (b) Power consumption

FIGURE 3: Delay (SP1 in black and SP2 in grey) and power consumption with different values of load $\lambda^1_{v,in}$ of SP 1

5 Conclusion

We propose an efficient modelling approach for optimal slice resource dimensioning, in order to minimize dynamic power under latency constraints. We are currently working on modeling latency constraints in probabilistic terms (Reliability), and to analyze different slice requirements typical in 5G networks (eMBB, mMTC, URLLC) on larger and realitic network topologies.

Références

[E+22] M. Elkael et al. Monkey business : Reinforcement learning meets neighborhood search for virtual network embedding. *Computer Networks*, 2022.

[F+13] A. Fischer et al. Virtual network embedding : A survey. *IEEE Commun. Surv. Tutor.*, 2013.

[H+17] S. Haeri et al. Virtual network embedding via monte carlo tree search. *IEEE Trans. Cybern.*, 2017.

[H+22] W. Huang et al. Dimensioning resources of network slices for energy-performance trade-off. In *IEEE ISCC*, 2022.

[I+18] I.Afolabi et al. Network slicing and softwarization : A survey on principles, enabling technologies, and solutions. *IEEE Commun. Surv. Tutor.*, 2018.