# Uncertainty-Aware Resource Provisioning for Network Slicing

Quang-Trung Luu, Sylvaine Kerboeuf, Michel Kieffer

# Uncertainty-Aware Resource Provisioning for Network Slicing

Quang-Trung Luu, Sylvaine Kerboeuf, and Michel Kieffer

***Abstract*—Network slicing allows Mobile Network Operators to split the physical infrastructure into isolated virtual networks (slices), managed by Service Providers to accommodate customized services. The Service Function Chains (SFCs) belonging to a slice are usually deployed on a best-effort premise: nothing guarantees that network infrastructure resources will be sufficient to support a varying number of users, each with uncertain requirements.**

**Taking the perspective of a network Infrastructure Provider (InP), this paper proposes a resource provisioning approach for slices, robust to a partly unknown number of users with random usage of the slice resources. The provisioning scheme aims to maximize the total earnings of the InP, while providing a probabilistic guarantee that the amount of provisioned network resources will meet the slice requirements. Moreover, the proposed provisioning approach is performed so as to limit its impact on low-priority background services, which may co-exist with slices in the infrastructure network.**

**Taking all these constraints into account leads to an integer programming problem with many nonlinear constraints. These constraints are first relaxed to get an integer linear programming formulation of the slice resource provisioning problem. This problem is then solved considering the slice resource provisioning demands jointly. A suboptimal approach is finally proposed where slice resource provisioning demands are considered sequentially. Both solutions are compared to provisioning schemes that do not account for best-effort services sharing the common infrastructure network, as well as uncertainties in the slice resource demands.**

***Index Terms*—Network slicing, resource provisioning, uncertainty, wireless network virtualization, 5G, linear programming.**

## I. Introduction

Network slicing will play an essential role in 5G communication systems [1–3]. Leveraging network function virtualization, network slicing reduces overall equipment and management costs [4] by increasing flexibility in the way the network is operated [5]. Multiple dedicated end-to-end virtual networks or *slices* can be managed in parallel over a given infrastructure network owned by one or several Infrastructure Providers (InPs). With network slicing, vertical markets can be addressed: Customers can manage their own applications by exploiting built-in network slices tailored to their needs [6].

In the extended survey [3] of the so far research efforts in 5G network slicing, the authors provide a taxonomy of network slicing, architectures and future challenges. One of the significant questions is how to meet the slice requirements of different verticals, where multiple network segments including the radio access, transport, and core networks, have to be considered. Infrastructure networks on which slices are operated must support high-quality services with increasing resource consumption (video streaming, telepresence, augmented reality, remote vehicle operation, gaming, *etc*.). Moreover, the number of users of each slice, their location (usually difficult to predict [7]), and resource demands may fluctuate with time. These uncertainties may impact significantly the resources consumed by each slice and raise the challenging problem of *slice resource provisioning*. Enough infrastructure resources should be dedicated to a given slice to ensure an appropriate Quality of Service (QoS) despite the uncertainties in the number of slice users and their demands. Over-provisioning should also be avoided, to limit the infrastructure leasing costs and leave resources to concurrent slices.

Existing work on network slicing, see, *e.g*., [3, 8–10], is mainly focused on the resource allocation aspect, *i.e*., assigning infrastructure network resources to virtual network components, with the aim to maximize resource utilization and minimize operation costs. The traffic dynamics in individual slices, such as flow arrival/departure, as well as the dynamics of resource availability on the network infrastructure, may lead to slice QoS below the level expected by the Service Provider (SP) managing the slice. Consequently, to fully unleash the power of network slicing in dynamic environments, uncertainties related to the resource demands need to be carefully addressed.

The main contribution of this paper is to propose a method to provision infrastructure resources for network slices, while being robust to a partly unknown number of users with a random usage of the slice resources. This robustness is achieved by providing a probabilistic guarantee that the amount of provisioned network resources for a slice will meet the slice requirements. Since some parts of the infrastructure network on which slices should be deployed are often already employed by low-priority background services, a second contribution of the proposed provisioning approach is to limit its impact on these services. The robustness to uncertainties of demands as well as the limitation of the impact on background services leads to an integer programming problem with many nonlinear constraints. These constraints are relaxed to get an integer linear programming formulation of the slice resource provisioning problem. This problem is then solved considering the slice resource provisioning demands jointly. A suboptimal approach is then proposed where slice resource provisioning demands

Q.-T. Luu is with Nokia Bell Labs and the L2S, CNRS-CentraleSupélec-Univ Paris-Sud-Univ Paris-Saclay, France, e-mail: quang_trung.luu@nokia.com.

S. Kerboeuf is with Nokia Bell Labs, France, e-mail: sylvaine.kerboeuf@nokia-bell-labs.com.

M. Kieffer is with the L2S, CNRS-CentraleSupélec-Univ Paris-Sud-Univ Paris-Saclay, France, e-mail: michel.kieffer@l2s.centralesupelec.fr.

are considered sequentially. Both solutions are compared to provisioning schemes that do not account for uncertainties in the slice resource demands as well as best-effort services sharing the common infrastructure network.

The rest of the paper is structured as follows. Section II analyzes related work, and highlights our main contributions. The model of the infrastructure network and of the slice resource demands are presented in Section III. The robust slice resource provisioning problem with uncertainties in the number of users as well as in the resource demands and accounting for the best-effort background services is then formulated in Section IV. The robust slice provisioning problem is transformed into an integer linear programming (ILP) problem in Section V. Numerical results are presented in Section VI. Finally, Section VII draws some conclusions and perspectives.

## II. Related work

In many conventional approaches enough network resources are allocated to make a service available to all users, all the time.

For that purpose, in [11], flexible service availability levels are defined, leading to cost savings for the infrastructure provider that can offer overbooked resources for users accepting a service with possibly degraded availability. In the context of network slicing, SPs can benefit from such an approach by providing services with reduced availability or degraded quality to some users ready to accept these conditions. Nevertheless, to evaluate the incidence on the QoS of such under-provisioning mechanism, it is necessary to introduce models for the service's number of users and of the resource consumption, which have not been considered in [11].

A worst-case allocation at peak traffic is considered in [8, 10]. Nevertheless, this infrastructure resource overbooking is costly and most of the time unnecessary, as all individual slice resource demands are very unlikely peaking simultaneously. In [12], the virtual network embedding problem is solved considering uncertain traffic demands. A mixed integer linear programming (MILP) formulation is considered, where some of the constraints are required to be satisfied with high probability. In [13], the total deployment costs for cloud computing applications are minimized, while satisfying some QoS constraints. To cope with the uncertain nature of the demands, a stochastic optimization approach is adopted by modeling user demands as random variables obeying normal distributions. Deployment is performed based on the mean demands increased by an integer amount of their standard deviations. This might lead to a conservative solution, requiring more allocated resources than needed. This also reduces somehow the possibility of having service-dependent required confidence levels.

In [14], the necessity to consider diversified services, with various requirements such as bandwidth and latency, is pointed out. To efficiently guarantee a timely data transfer within specified delay bound, and providing a desired throughput, [14] introduces a novel heterogeneous statistical delay-bounded QoS provisioning architecture integrating device-to-device communication, full-duplex, and cognitive radio. The potential benefits of network slicing have, however, not been considered. The heterogeneity of service requirements is also considered in [15] and [16], where resource provisioning frameworks are introduced for network slicing in a virtualized radio access network (V-RAN) context. Provisioning is performed at the resource block (RB) level. The problem of radio resource provisioning and allocation from base stations (BSs) to a slice, and the assignment of users within the slices to BSs are considered. The first problem (provisioning and allocation) is solved in [15] via heuristics, while a deep reinforcement learning technique is considered in [16]. The second problem (user assignment) is cast in the framework of an NP-complete 0-1 multiple knapsack problem.

In [17], a network slicing problem is considered for light fidelity (Li-Fi) attocell access networks. The proposed slicing scheme supports dynamic allocation of Li-Fi uplink and downlink resources among multiple Li-Fi access points to slices with diversified requirements. In that study, slice resource demands consists of spectrum resources (resource blocks) and the buffer space of the access points. In this paper, no specific access network is considered. Moreover, contrary to [17], we have considered a provisioning scheme involving its impact on the background traffic, and the uncertainties of the slice resource demands.

A network slice embedding problem is considered in [18], where available resources and resource demands are assumed to be partly uncertain. They are described by normal distributions built upon the data history on mobile network resource availability as well as slice resource utilization. To control the probability that a slice embedding solution will benefit from enough infrastructure resource, despite the uncertainties, some adjustable safety factor $\gamma$ is introduced. As in [13], enough resources are dedicated to a service so as to satisfy the mean plus $\gamma$ times the standard deviation of the demands. In [18], additionally, a similar approach is considered to account for the uncertainty in the available resources. A *probability of feasibility*, depending on $\gamma$, is then evaluated for the slice embedding to measure the risk of having a degraded service for some users. The proposed solution leads to a slice resource allocation solution robust to uncertainties. Nevertheless, the resource demands of the different components of the slice have been considered as independent. Moreover, the safety factor $\gamma$ is chosen identical for resource demands and available resources. This again may lead to allocating more resources than strictly necessary, and increases the operation cost.

The network slice embedding problem with demand uncertainties is also addressed in [19]. The minimization of deployment costs considering first static resource demands is formulated as an MILP. Two robust network slice design formulations are then proposed, in uncorrelated and correlated demand uncertainties are considered. In both cases, the objective function is unchanged but some constraints become nonlinear due to the addition of inner maximization problems. These problems account for the upper bound of the resource demands, thus making the network slice embedding problem more complex. A linearization technique inspired by [20] is proposed to relax these inner problems. A tuning parameter $\Gamma$

is introduced to control the trade-off between robustness to the demand uncertainties and the deployment costs. Uncertainties related to the background traffics on the infrastructure, which clearly affect the residual infrastructure resources, are not considered.

To reduce the computation effort required to solve the robust network slice embedding problem, [21] proposes to use a genetic algorithm, shown to surpass the performance of state-of-the-art robust MILP solvers used, *e.g.*, in [19]. Uncertainties in infrastructure link bandwidth are also considered in [22], where possible failures of infrastructure nodes or links are taken into account to propose a robust algorithm that minimizes the network resource consumption under uncertain demands, while remapping the network slice in case of infrastructure failures. Since [19], [21], and [22] assume that the distribution of the variable demands and available infrastructure resource are unknown, their optimization are relatively conservative. Furthermore, uncertainties in various types of resources such as computing, memory, or wireless are not addressed.

In all the above works, the effect of the best effort background services combined with a approach robust to uncertainties in the demands and in the infrastructure resources has not yet been considered for the slice provisioning problem. As shown in the sequel these are two important aspects that need to be taken into account for efficiently providing slices with guaranteed Service Level Agreement (SLA). Finally, we emphasize that these approaches are solving the problem of resource allocation rather than provisioning, *i.e.*, reserving infrastructure resource for a further allocation.

In this paper, we adopt the point of view of the Infrastructure Provider (InP). We propose a provisioning scheme which aims at maximizing the total earnings of the InP, while providing a probabilistic guarantee that the amount of provisioned network resources will meet the slice requirements. In the provisioning approach, various infrastructure network resources are booked for a slice to satisfy its requirements. Slice resource demands are aggregated. Consequently, resources of several infrastructure nodes may have to be gathered and parallel physical links have to be considered to satisfy these aggregated demands.

The provisioning is performed prior to the resource allocation at the time of deployment described, *e.g.*, in [23, 24], where virtual nodes and links are mapped on the infrastructure network. The idea of provisioning resources in [15] and [16] is relatively similar to ours: resources are provisioned in advance for each slice before the slice deployment and exploitation takes place. Nevertheless, [15] and [16] mainly focus on the V-RAN context, where the slicing is performed at radio resource block level. In our paper, the considered resources are mainly in the core and access network (computing, memory, bandwidth, and wireless resources). We do not care about coverage constraints. This aspect has been considered in our previous paper [25]. Coverage constraints could be taken into account in a way similar to that introduced in [25], by resorting to a two-step provisioning approach: one step focusing on coverage constraints, the second on the core network resource provisioning, as in [26].

While the approach in [16] allows the adjustment of resources allocated to slices after each decision time interval (slicing time), the uncertainties of slice resource demand during each time interval are not considered. Moreover, instead of considering uncertainties in the available network resource, as in [18], here, we consider best-effort background services running in parallel with the network slices on the infrastructure network. The proposed scheme is able to maintain the impact of resource provisioning on those background services at a prescribed level.

## III. Notations and Hypotheses

A typical network slicing system involves several entities. This includes one or many InPs, Mobile Network Operators (MNOs), and SPs, as depicted in Figure 1 [4]. InPs own and manage the wireless and wired infrastructure such as the cell sites, the fronthaul and backhaul networks, and cloud data centers. MNOs lease resources from InPs to setup and manage the slices. SPs then exploit the slices supplied by MNOs, and provide to their customers the required services running within the slices.

Figure 1 illustrates the various steps involved in the proposed provisioning approach. To satisfy expected service demands of users (1), the SP identifies the necessary service characteristics in terms of QoS, satisfaction probability, *etc*. These service characteristics are forwarded to the MNO within an SLA denoted SM-SLA (2). The MNO then translates these characteristics into constraints to be satisfied by the slice dedicated to the required service (3). The slice constraints form the MI-SLA between the MNO and the InP and include the aggregate resource demands of all users, the successful provisioning probability that has to be guaranteed by the InP, *etc*.

Then slice resource provisioning is performed (4-5) by the InP based on the MI-SLA between the MNO and the InP. This step is followed by slice deployment and activation (6): The provisioned resource are leased by the MNO to deploy and activate the target slice. Finally, the slice is exploited (7-8) by an SP who assigns users to the SFCs supplied by the MNO.

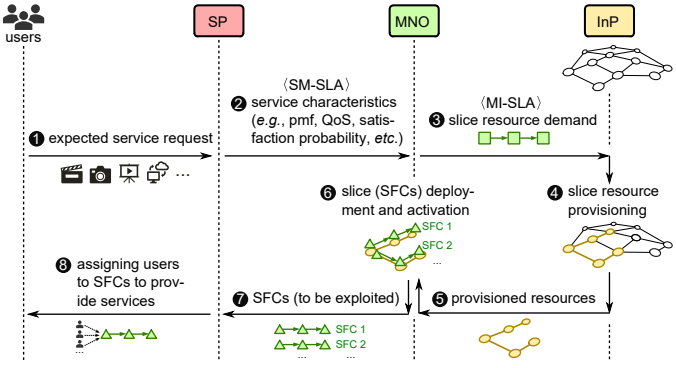



Fig. 1. Network slicing entities and their SLA-based relationships.

The service and slice characteristics within the SM-SLA and the MI-SLA are detailed in what follows.

The SM-SLA describes, at a high level of abstraction, characteristics of the service with the desired QoS. These characteristics may be time-varying due, *e.g.*, to user mobility.

In this paper, one considers SM-SLAs composed of: (*i*) a probability mass function (pmf) describing the target number of users/devices to be supported by the slice, (*ii*) a description of the characteristics of the service and of the way it is employed by a typical user/device, and (*iii*) a target probability of service satisfaction. In addition, several time intervals may be considered in the SM-SLA, intervals over each of which the service characteristics and constraints are assumed constant, but may vary from one interval to the next one. These time intervals translate, *e.g.*, day and night variations of user demands. They last between tens of minutes to hours. Uncertainties in the number of users and in the user demands may account for short-term small variations of the demands, due, *e.g.*, to user mobility. It is of the responsibility of the SP and MNO to properly scale the requirements expressed in the SM-SLA, by considering, for example, similar services deployed in the past.

Taking the InP perspective, our aim, with resource provisioning is to reserve, somewhat in advance, enough infrastructure resources to ensure that the MNO will be able to provide a slice with characteristics as stated in the SM-SLA it has with the SP. The time scale at which provisioning is performed is much larger than that at which slices are deployed and adapted to meet actual time-varying user demands. In what follows, one focuses on a given time interval over which resources will be provisioned so as to be compliant with the variations of user demands within a slice. The duration of this time interval results from a compromise between the need to update the provisioning and the level of conservatism in the amount of provisioned resources required to satisfy fast fluctuating user demands.

Each slice consists of one or multiple Service Function Chains (SFCs) of different types. An SFC consists of an ordered set of interconnected Virtual Network Functions (VNFs) describing the processing applied to data flows related to a given service. The MNO translates the SP high-level demands into SFCs able to fulfill the service requirements. Based on the characteristics of the service and of its usage, the MNO describes the way the slice (SFCs) resources are consumed by a given user/device. To characterize the variability over time and among users of these demands, we assume that the MNO considers a probabilistic description of the consumption of slice resources by a typical user. The MNO then forwards to the InP these characteristics as part of an SLA between them (MI-SLA). Each InP then provisions infrastructure resources needed for the SFCs. Under the MI-SLA, this provisioning has to meet the target probability of service satisfaction. This translates the fact that enough resources of various types have been provisioned to satisfy the resource demands of the users of the service. This probability is evaluated considering the pmf describing the number of users of the service and the probabilistic description of the slice resource consumption by a typical user. When performing the provisioning, each InP has to limit the impact on other best-effort service running on its infrastructure network.

In this paper, one considers an infrastructure owned by a single InP. To perform the provisioning, the InP has to identify the infrastructure nodes which will provide resources for future deployment of VNFs and the links able to transmit data between these nodes, while respecting the structure of SFCs and optimizing a given objective (*e.g.*, minimizing the infrastructure and software fee costs).

When several InPs are present, an MNO may send the same provisioning request to different InPs. The InPs may then run in parallel the proposed provisioning algorithm. If several of these InPs are able to satisfy the request, the MNO may select the InP providing the best service or the cheapest one. When InPs have to collaborate to satisfy the service request of the MNO, the way this collaboration may be performed, the exchange of information between InPs has to be formalized and the proposed algorithm is no more sufficient to address such more difficult situation.

Table I summarizes all parameters involved in the description of the infrastructure network and the graph of SFCs for a slice.

TABLE I
TABLE OF NOTATIONS

| *Symbol* | *Description* |
|---|---|
| $\mathcal{G}$ | Infrastructure network graph, $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ |
| $\mathcal{N}$ | Set of infrastructure nodes |
| $\mathcal{E}$ | Set of infrastructure links |
| $a_n(i)$ | Available resource of type $n$ at node $i$ |
| $a_{\mathrm{b}}(ij)$ | Available bandwidth of link $ij$ |
| $c_n(i)$ | Per-unit cost of resource of type $n$ for node $i$ |
| $c_{\mathrm{b}}(ij)$ | Per-unit cost for link $ij$ |
| $c_{\mathrm{f}}(i)$ | Fixed cost for using node $i$ |
| $\mathcal{S}$ | Set of slices to be deployed |
| $\mathcal{G}_s$ | SFC graph, $\mathcal{G}_s = (\mathcal{N}_s, \mathcal{E}_s)$ |
| $\mathcal{N}_s$ | Set of VNFs $v$ |
| $\mathcal{E}_s$ | Set of interconnections $vw$ between VNF $v$ and $w$ |
| $r_{s,n}(v)$ | Fixed amount of resources of type $n$ required by an instance of VNF $v$ to operate properly |
| $r_{s,\mathrm{b}}(vw)$ | Fixed amount of bandwidth to sustain traffic demand between VNF instances $v$ and $w$ |
| $U_{s,n}(v)$ | Random amount of resources of type $n$ of virtual node $v$ employed by a user |
| $U_{s,\mathrm{b}}(vw)$ | Random amount of bandwidth of virtual link $vw$ employed by a user |
| $R_{s,n}(v)$ | Random amount of resources of type $n$ of virtual node $v$ employed by $N_s$ users |
| $R_{s,\mathrm{b}}(vw)$ | Random amount of bandwidth of virtual link $vw$ employed by $N_s$ users |
| $B_{s,n}(i)$ | Amount of resources of type $n$ on infrastructure node $i$ consumed by background services |
| $B_{s,\mathrm{b}}(ij)$ | Amount of bandwidth on infrastructure link $ij$ consumed by background services |

### A. Infrastructure Network

Consider an infrastructure network managed by a given InP. This network is represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of infrastructure nodes and $\mathcal{E}$ is the set of infrastructure links, which correspond to the wired connections between and within nodes (loopback links) of the infrastructure network.

Each infrastructure node $i \in \mathcal{N}$ is characterized by a given amount of available computing, memory, and wireless resources, denoted as $a_{\mathrm{c}}(i)$, $a_{\mathrm{m}}(i)$, and $a_{\mathrm{w}}(i)$, which may be *allocated* to new network slices. These amounts correspond to the total available resources reduced by the amount of

resources previously provisioned to concurrent slices. An operation cost paid by the InP is attributed to each unit of node resource. The per-unit node resource cost associated to a given node $i$ consists of a fixed part $c_{\mathrm{f}}(i)$ for node disposal (paid for each slice using node $i$), and variable parts $c_{\mathrm{c}}(i)$, $c_{\mathrm{m}}(i)$, and $c_{\mathrm{w}}(i)$, which depend linearly on the amount of resources provided by that node.

Similarly, each infrastructure link $ij \in \mathcal{E}$ connecting node $i$ to $j$ has an available bandwidth $a_{\mathrm{b}}(ij)$, and an associated per-unit bandwidth cost $c_{\mathrm{b}}(ij)$. Several distinct VNFs of the same slice may be deployed on a given infrastructure node. When communication between these VNFs is required, an internal (loopback) infrastructure link $ii \in \mathcal{E}$ can be used at each node $i \in \mathcal{N}$, as in [27], in the case of interconnected virtual machines (VMs) deployed on the same host. The associated per-unit bandwidth cost, in that case, is $c_{\mathrm{b}}(ii)$.

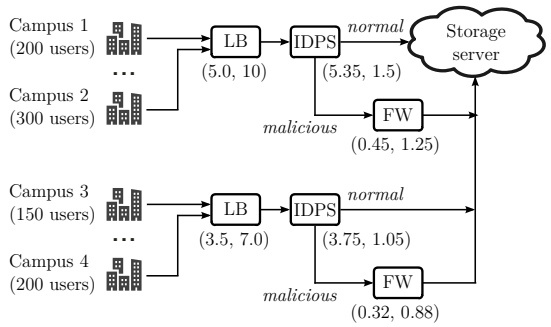
### B. Graphs of Resource Demands

A demand of resources is defined on the basis of an SLA between an SP and the MNO. As in [25], we consider that a slice is devoted to a single type of service supplied by a given type of SFC. Several instances of that SFC may have to be deployed so as to satisfy the user demand. The topology of each SFC of slice $s$ is represented by a graph $\mathcal{G}_s = (\mathcal{N}_s, \mathcal{E}_s)$ representing the VNFs and their interconnections. Each virtual node $v \in \mathcal{N}_s$ represents an instance of a VNF, and each virtual link $vw \in \mathcal{E}_s$ represents the connection between virtual nodes $v$ and $w$.

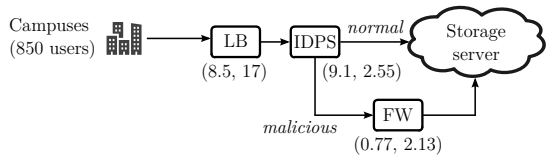
The following *weighted* graphs are build upon $\mathcal{G}_s$.

- $\mathcal{G}_s^{\mathrm{r}} = (\mathcal{N}_s^{\mathrm{r}}, \mathcal{E}_s^{\mathrm{r}})$ is the graph of Resource Demands of an SFC (SFC-RD) of slice $s$. Each node $v \in \mathcal{N}_s^{\mathrm{r}}$ is characterized by a fixed amount of computing $r_{s,\mathrm{c}}(v)$ and memory $r_{s,\mathrm{m}}(v)$ resources allocated by the infrastructure node on which the VNF instance $v$ is deployed to operate properly. Each link $vw \in \mathcal{E}_s^{\mathrm{r}}$ is characterized by a given amount of bandwidth $r_{s,\mathrm{b}}(vw)$ that has to be allocated by the infrastructure network to sustain the traffic demand between VNF instances $v$ and $w$.
- $\mathcal{G}_s^{\mathrm{U}} = \left(\mathcal{N}_s^{\mathrm{U}}, \mathcal{E}_s^{\mathrm{U}}\right)$ is the graph of Resource Demands a typical User (U-RD) of slice $s$. Each user of slice $s$ is assumed to consume a random proportion of the resources of an SFC of that slice. In addition, the consumed resources by various users are represented by independently and identically distributed random vectors. For a typical user, let $U_{s,\mathrm{c}}(v)$, $U_{s,\mathrm{m}}(v)$, $U_{s.\mathrm{w}}(v)$, and $U_{s,\mathrm{b}}(vw)$ be the random amount of employed resources of VNF instance $v \in \mathcal{N}_s^{\mathrm{r}}$ and of virtual link $vw \in \mathcal{E}_s^{\mathrm{r}}$ of the SFC-RD $\mathcal{G}_s^{\mathrm{r}}$.
- $\mathcal{G}_s^{\mathrm{R}} = \left(\mathcal{N}_s^{\mathrm{R}}, \mathcal{E}_s^{\mathrm{R}}\right)$ is the graph of Resource Demands of Slice $s$ (S-RD). The weight of each node $v \in \mathcal{N}_s^{\mathrm{R}}$ and of each link $vw \in \mathcal{E}_s^{\mathrm{R}}$ represents the aggregate amount of resources employed by a random number $N^s$ of independent users of slice $s$. These amounts are described by random variables denoted as $R_{s,\mathrm{c}}(v)$, $R_{s,\mathrm{m}}(v)$, $R_{s,\mathrm{w}}(v)$, and $R_{s,\mathrm{b}}(vw)$, for computing, memory, wireless, and bandwidth demand, respectively.

Figure 2 illustrates the SFCs required for the deployment of an HTTP traffic monitoring service inspired by [28] and its associated S-RD graph. Figure 2a describes the three VNFs to be deployed, including: a Load Balancer (LB), an Intrusion Detection Prevention System (IDPS), and a Firewall (FW) function. Each of these VNFs is characterized by computing and memory requirements, *i.e.*, $r_{s,\mathrm{c}}$ and $r_{s,\mathrm{m}}$, shown below the function. Figure 2b shows the corresponding SFC-RD graph. All identical instances of SFCs deployed within the slice are represented by a single graph whose structure is identical to the SFC-RD graph. The requirements in terms of storage and memory of each component of this S-RD graph aggregate the corresponding requirements of the components of the SFC-RD graphs in Figure 2a.




(a) SFC-RD graph.




(b) Corresponding S-RD graph.

Fig. 2. SFCs and their required computing (in CPUs) and memory (in GBytes) resources, denoted as $(r_{s,\mathrm{c}}, r_{s,\mathrm{m}})$ below each function in the figure, for the deployment of an HTTP traffic monitoring service (top) and their associated S-RD graph (bottom).

Considering the analysis of co-allocated online services of large scale data centers reported in [29], the utilization of CPU and memory of virtual machines (VMs) have a positive correlation in the majority of cases. Moreover, this correlation is particularly strong at the VMs that execute the same jobs, showing correlation coefficients larger than $0.85$. Based on this observation, for a typical user, the resource demands of different types for a given node $v \in \mathcal{N}_s^{\mathrm{U}}$ are considered to be correlated. The demands for resources of the same type among virtual nodes are also correlated. Finally, the resulting traffic demands between nodes is usually also correlated with the resource demands for a given virtual node. To represent this correlation, consider the vector of joint resource demands for a typical user of an SFC of slice $s$

$$\mathbf{U}_s = \left(U_{s,\mathrm{c}}(v), U_{s,\mathrm{m}}(v), U_{s,\mathrm{w}}(v), U_{s,\mathrm{b}}(vw)\right)^{\top}_{(v,vw)\in\mathcal{G}_s^{\mathrm{U}}}.$$

Assuming that $U_{s,\mathrm{c}}(v)$, $U_{s,\mathrm{m}}(v)$, $U_{s.\mathrm{w}}(v)$, and $U_{s,\mathrm{b}}(vw)$ are normally distributed, $\mathbf{U}_s$ follows a multivariate normal distribution with probability density

$$f(\mathbf{x}; \boldsymbol{\mu}_s, \boldsymbol{\Gamma}_s) = \frac{1}{\sqrt{(2\pi)^{\mathrm{card}(\mathbf{U}_s)}\,|\boldsymbol{\Gamma}_s|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_s)^{\top}(\boldsymbol{\Gamma}_s)^{-1}(\mathbf{x}-\boldsymbol{\mu}_s)}, \tag{1}$$

with mean

$$\boldsymbol{\mu}_s = \left(\mu_{s,\mathrm{c}}(v), \mu_{s,\mathrm{m}}(v), \mu_{s,\mathrm{w}}(v), \mu_{s,\mathrm{b}}(vw)\right)^\top_{(v,vw)\in\mathcal{G}_s^{\mathrm{U}}},$$

and covariance matrix $\boldsymbol{\Gamma}_s$ such that

$$\mathrm{diag}\left(\boldsymbol{\Gamma}_s\right) = \left(\sigma^2_{s,\mathrm{c}}(v), \sigma^2_{s,\mathrm{m}}(v), \sigma^2_{s,\mathrm{w}}(v), \sigma^2_{s,\mathrm{b}}(vw)\right)^\top_{(v,vw)\in\mathcal{G}_s^{\mathrm{U}}},$$

the off-diagonal elements of $\boldsymbol{\Gamma}_s$ representing the correlation between different types of resource demands. In (1), $\mathrm{card}\left(\mathbf{U}_s\right)$ is the number of elements of $\mathbf{U}_s$. One has thus $U_{s,n}(v) \sim \mathcal{N}\left(\mu_{s,n}(v), \sigma^2_{s,n}(v)\right)$, with $n \in \{\mathrm{c},\mathrm{m},\mathrm{w}\}$ and $U_{s,\mathrm{b}}(vw) \sim \mathcal{N}\left(\mu_{s,\mathrm{b}}(vw), \sigma^2_{s,\mathrm{b}}(vw)\right)$.

Assume that the number of users $N_s$ to be supported by slice $s$ is described by the pmf

$$p_k = \Pr\left(N_s = k\right). \tag{2}$$

Since the amount of resources of VNF $v$ and of virtual link $vw$ consumed by different users is represented by independently and identically distributed copies of $\mathbf{U}_s$, the joint distribution of the aggregate amount $\mathbf{U}_{s,k}$ of resources consumed by $k$ independent users is $f\left(\mathbf{x}, k\boldsymbol{\mu}_s, k^2\boldsymbol{\Gamma}_s\right)$. The total amount of resources employed by a random number $N_s$ of independent users, $\mathbf{R}_s = \mathbf{U}_{s,N_s} = \left(R_{s,\mathrm{c}}(v), R_{s,\mathrm{m}}(v), R_{s,\mathrm{w}}(v), R_{s,\mathrm{b}}(vw)\right)^\top_{(v,vw)\in\mathcal{G}_s^{\mathrm{R}}}$, is distributed according to

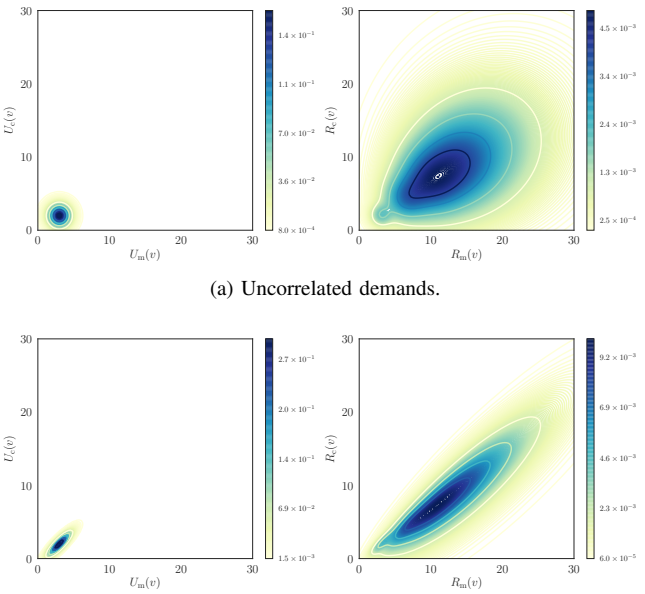
$$g\left(\mathbf{x}, \boldsymbol{\mu}_s, \boldsymbol{\Gamma}_s\right) = \sum_{k=0}^{\infty} p_k f\left(\mathbf{x}, k\boldsymbol{\mu}_s, k^2\boldsymbol{\Gamma}_s\right). \tag{3}$$

The typical joint distribution of two components of $\mathbf{U}_s$ and $\mathbf{R}_s$ is illustrated in Figure 3. Considering a virtual node $v$ of a given slice $s$, Figure 3 represents the joint distribution $f\left(\mathbf{x}; \boldsymbol{\mu}_s, \boldsymbol{\Gamma}_s\right)$ of $U_{s,\mathrm{c}}(v)$ and $U_{s,\mathrm{m}}(v)$ and the resulting joint distribution $g\left(\mathbf{x}, \boldsymbol{\mu}_s, \boldsymbol{\Gamma}_s\right)$ of $R_{s,\mathrm{c}}(v)$ and $R_{s,\mathrm{m}}(v)$. Here $N_s$ follows the binomial distribution $N_s \sim \mathcal{B}(10, 0.5)$, $\boldsymbol{\mu}_s = [2,3]^\top$. In Figure 3a, $\boldsymbol{\Gamma}_s = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is diagonal. Even if the level sets of $f\left(\mathbf{x}; \boldsymbol{\mu}_s, \boldsymbol{\Gamma}_s\right)$ are circles, the level sets of the resulting $g\left(\mathbf{x}, \boldsymbol{\mu}_s, \boldsymbol{\Gamma}_s\right)$ illustrate the correlation between $R_{s,\mathrm{c}}(v)$ and $R_{s,\mathrm{m}}(v)$. In Figure 3b, $\boldsymbol{\Gamma}_s = \begin{bmatrix} 1 & 0.85 \\ 0.85 & 1 \end{bmatrix}$ is non-diagonal, *i.e.*, $U_{s,\mathrm{c}}(v)$ and $U_{s,\mathrm{m}}(v)$ are correlated, the correlation between $R_{s,\mathrm{c}}(v)$ and $R_{s,\mathrm{m}}(v)$ increases significantly.

### C. Resource Consumption of Best-Effort Background Services

In the considered time interval, a given part of the available resources is consumed by other best-effort background services for which no resource provisioning has been performed. The aggregate amount of resources consumed by these best-effort services is represented by random variables $B_{\mathrm{c}}(i)$, $B_{\mathrm{m}}(i)$ and $B_{\mathrm{w}}(i)$, $\forall i \in \mathcal{N}$, and $B_{\mathrm{b}}(ij)$, $\forall ij \in \mathcal{E}$. Each of those variables is assumed to be uncorrelated and Gaussian distributed, $B_n(i) \sim \mathcal{N}\left(\mu_{\mathrm{B},n}(i), \sigma^2_{\mathrm{B},n}(i)\right)$, $\forall i \in \mathcal{N}$, $\forall n \in \{\mathrm{c},\mathrm{m},\mathrm{w}\}$, and $B_n(i) \sim \mathcal{N}\left(\mu_{\mathrm{B,b}}(ij), \sigma^2_{\mathrm{B,b}}(ij)\right)$, $\forall ij \in \mathcal{E}$. Finally, denote $\mathbf{B} = \left(B_{\mathrm{c}}(i), B_{\mathrm{m}}(i), B_{\mathrm{b}}(ij)\right)^\top_{(i,ij)\in\mathcal{G}}$ as the vector gathering all resource consumption of the background services. $\mathbf{B}$ is distributed according to $f\left(\mathbf{x}; \boldsymbol{\mu}_{\mathrm{B}}, \boldsymbol{\Gamma}_{\mathrm{B}}\right)$, with

$$\boldsymbol{\mu}_{\mathrm{B}} = \left(\mu_{\mathrm{B,c}}(i), \mu_{\mathrm{B,m}}(i), \mu_{\mathrm{B,w}}(i), \mu_{\mathrm{B,b}}(ij)\right)^\top_{(i,ij)\in\mathcal{G}}$$




(a) Uncorrelated demands.

(b) Correlated demands.

Fig. 3. Joint distribution $f\left(\mathbf{x}; \boldsymbol{\mu}_s, \boldsymbol{\Gamma}_s\right)$ (top left and bottom left) and $g\left(\mathbf{x}, \boldsymbol{\mu}_s, \boldsymbol{\Gamma}_s\right)$ (top right and bottom right), when $U_{s,\mathrm{c}}(v)$ and $U_{s,\mathrm{m}}(v)$ are (a) uncorrelated, and (b) correlated.

and

$$\boldsymbol{\Gamma}_{\mathrm{B}} = \mathrm{diag}\left(\sigma^2_{\mathrm{B,c}}(i), \sigma^2_{\mathrm{B,m}}(i), \sigma^2_{\mathrm{B,w}}(i), \sigma^2_{\mathrm{B,b}}(vw)\right)^\top_{(i,ij)\in\mathcal{G}},$$

since the elements of $\mathbf{B}$ are assumed to be uncorrelated.

## IV. Optimal Slice Resource Provisioning

Consider a set of slices $\mathcal{S}$ for which infrastructure resources have to be provisioned. To provision resource for a given slice $s \in \mathcal{S}$, the InP has to determine the amount of resources each of its infrastructure nodes and links has to reserve to satisfy the slice resource demands with a given probability. Moreover, the InP has to preserve enough resource for background services. This will be done by evaluating and bounding the probability that the provisioning impacts (reduces) the resources and traffic involved by best effort services.

The slice resource provisioning is represented by a mapping between the infrastructure graph $\mathcal{G}$ and the S-RD graph $\mathcal{G}_s^{\mathrm{R}}$, as depicted in Figure 4. In this example, slice $s$ consists of several linear SFCs of the same type. The mapping has to be performed so as to minimize the provisioning costs, while being able to satisfy the uncertain slice demands with a high probability. The constraints that have to be satisfied by this mapping are detailed in the following sections.

Let $\kappa_s(i,v)\, r_{s,n}(v)$ be the amount of resource of type $n \in \{\mathrm{c},\mathrm{m},\mathrm{w}\}$ provisioned by node $i$ for a VNF of type $v$, with $\kappa_s(i,v) \in \mathbb{N}_0$. Consequently $\kappa_s(i,v)$ represents the number of VNF instances of type $v \in \mathcal{N}_s$ that node $i$ will be able to host. Similarly, let $\kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw)$ be the bandwidth provisioned by link $ij$ to support the traffic between virtual nodes of type $v$ and $w$.

A solution of the provisioning problem for slice $s$ is thus defined by a given assignment of the variables $\boldsymbol{\kappa}_s = \left\{\kappa_s(i,v), \kappa_s(ij,vw)\right\}_{(i,ij)\in\mathcal{G},(v,vw)\in\mathcal{G}_s^{\mathrm{R}}}$. This assignment has

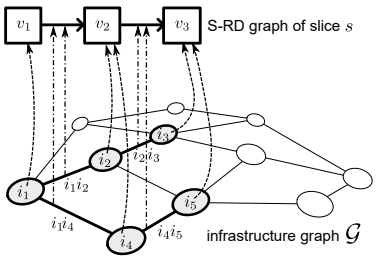



Fig. 4. Provisioning of infrastructure resources for slice $s$: Resources from the infrastructure node $i_1$ and the aggregate resources from the infrastructure node pairs $(i_2, i_4)$ and $(i_3, i_5)$ are provisioned for the virtual nodes $v_1$, $v_2$, and $v_3$; The infrastructure links $(i_1i_2, i_1i_4)$ and $(i_2i_3, i_4i_5)$ (in bold lines) are used to provision resource for the virtual links $v_1v_2$ and $v_2v_3$.

to satisfy some constraints to ensure a satisfying behavior of the SFC and the satisfaction of the MI-SLA for slice $s$ defined in terms of probability of satisfaction of the aggregate user demands $\underline{p}_s$, see Section IV-A. In addition, from the perspective of the InP, this assignment has also to have a limited impact on the operation of background best-effort services.

### A. Constraints

Consider slice $s$ and a given assignment of the variables $\boldsymbol{\kappa}_s$. For a given node $v \in \mathcal{N}_s^{\mathrm{R}}$, the probability that enough resources are provisioned in the infrastructure network to satisfy the resource demand $R_{s,n}(v)$ of type $n \in \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$ is

$$p_{s,n}(v) = \Pr\Big\{ \sum_i \kappa_s(i,v)\, r_{s,n}(v) \geqslant R_{s,n}(v) \Big\}. \quad (4)$$

Similarly, for a given virtual link $vw \in \mathcal{E}_s^{\mathrm{R}}$, the probability that enough bandwidth is provisioned in the infrastructure network to satisfy the demand $R_{s,\mathrm{b}}(vw)$ is

$$p_{s,\mathrm{b}}(vw) = \Pr\Big\{ \sum_{ij} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant R_{s,\mathrm{b}}(vw) \Big\}. \quad (5)$$

In both cases, the assignment has to be such that, for each infrastructure node $i \in \mathcal{N}$ and link $ij \in \mathcal{E}$, the total amount of provisioned resources for all slices $s \in \mathcal{S}$ is less or equal than the amount of available resources

$$\sum_{s,v} \kappa_s(i,v)\, r_{s,n}(v) \leqslant a_n(i), \quad (6)$$

$$\sum_{s,vw} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij). \quad (7)$$

The constraints (6)-(7) may leave no resources for the background best-effort services. The probability that the background best-effort services are impacted at a node $i$ or on the link $ij$ by the provisioning for all slices $s \in \mathcal{S}$ are, $\forall n \in \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$,

$$p_n^{\mathrm{im}}(i) = \Pr\Big\{ \sum_{s,v} \kappa_s(i,v)\, r_{s,n}(v) \geqslant a_n(i) - B_n(i) \Big\} \quad (8)$$

and

$$p_{\mathrm{b}}^{\mathrm{im}}(ij) = \Pr\Big\{ \sum_{s,vw} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant a_{\mathrm{b}}(ij) - B_{\mathrm{b}}(ij) \Big\}. \quad (9)$$

The impact probabilities (IPs) of the provisioning for all slice $s \in \mathcal{S}$ on the nodes and links resources employed by best-effort service has to be such that, $\forall (i,ij) \in \mathcal{G}$, $\forall n \in \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$

$$p_n^{\mathrm{im}}(i) \leqslant \overline{p}^{\mathrm{im}}, \quad (10)$$

$$p_{\mathrm{b}}^{\mathrm{im}}(ij) \leqslant \overline{p}^{\mathrm{im}}, \quad (11)$$

where $\overline{p}^{\mathrm{im}}$ is the maximum tolerated impact probability. The value of $\overline{p}^{\mathrm{im}}$ is chosen by the InP to provide sufficient resources for the background services at every infrastructure nodes and links. A small value of $\overline{p}^{\mathrm{im}}$ leads to a small impact of slice resource provisioning on background services, but makes the provisioning problem more difficult to solve compared to a value of $\overline{p}^{\mathrm{im}}$ close to one.

The considered assignment has to satisfy additional constraints to ensure that the data can be correctly carried between VNFs. For each virtual link $vw \in \mathcal{E}_s^{\mathrm{R}}$, resources on a sequence of infrastructure links must be provisioned between *each* pair of infrastructure nodes that have provisioned resources to the virtual nodes $v$ and $w$. One obtains a flow conservation constraint similar to that introduced in [25]. One should have $\forall s \in \mathcal{S}$, $\forall i \in \mathcal{N}$, $\forall vw \in \mathcal{E}_s$,

$$\sum_{j \in \mathcal{N}} \left[\kappa_s(ij,vw) - \kappa_s(ji,vw)\right] = \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{vu} r_{s,\mathrm{b}}(vu)}\right) \kappa_s(i,v) - \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{uw} r_{s,\mathrm{b}}(uw)}\right) \kappa_s(i,w). \quad (12)$$

Finally, considering an assignment $\boldsymbol{\kappa} = \{\boldsymbol{\kappa}_s\}_{s \in \mathcal{S}}$, satisfying (6)–(12), the probability that this assignment is compliant with the constraints imposed for slice $s$ and by the infrastructure, *i.e.*, the Probability of Successful Provisioning (PSP) for slice $s$ is

$$p_s(\boldsymbol{\kappa}_s) = \Pr\Big\{ \sum_i \kappa_s(i,v)\, r_{s,n}(v) \geqslant R_{s,n}(v), \forall v, n, \sum_{ij} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant R_{s,\mathrm{b}}(vw), \forall vw \Big\}, \quad (13)$$

and, as stated in the MI-SLA, the InP has to ensure a minimum PSP of $\underline{p}_s$ for every slice $s \in \mathcal{S}$, *i.e.*,

$$p_s(\boldsymbol{\kappa}_s) \geqslant \underline{p}_s. \quad (14)$$

### B. Costs, Incomes, and Earnings

Considering the perspective of the InP, this section presents the cost, income, and earnings model for the slice resource provisioning problem.

Consider a given slice $s \in \mathcal{S}$ and its related assignment of the variables $\boldsymbol{\kappa}_s$. Let

$$x_s(\boldsymbol{\kappa}_s) = \begin{cases} 1 & \text{if } p_s(\boldsymbol{\kappa}_s) \geqslant \underline{p}_s \\ 0 & \text{else} \end{cases} \quad (15)$$

indicate whether the MI-SLA for slice $s$ is satisfied.

Define $I_s$ as the income obtained for a slice $s$ whose MI-SLA is satisfied. The income awarded to the InP from the MNO is then $I_s x_s(\boldsymbol{\kappa}_s)$.

The total provisioning cost $C_s(\boldsymbol{\kappa}_s)$ of a given slice $s$ for the InP is

$$C_s(\boldsymbol{\kappa}_s) = \sum_i \widetilde{\kappa}_s(i)\, c_{\mathrm{f}}(i) + \sum_{i,v,n} \kappa_s(i,v)\, r_{s,n}(v)\, c_n(i) + \sum_{ij,vw} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw)\, c_{\mathrm{b}}(ij), \quad (16)$$

where

$$\widetilde{\kappa}_s(i) = \begin{cases} 1 & \text{if } \sum_v \kappa_s(i,v) > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

The first term of $C_s(\boldsymbol{\kappa}_s)$ represents the fixed costs associated to the use of infrastructure nodes by slice $s$, whereas the second and the third terms indicate the cost of reserved resources from infrastructure nodes and links. The variable $\widetilde{\kappa}_s(i)$ indicates whether the infrastructure node $i$ is used by slice $s$.

Finally, the total earnings $E_s(\boldsymbol{\kappa}_s)$ obtained by the InP for the successful provisioning of slice $s$ is

$$E_s(\boldsymbol{\kappa}_s) = I_s x_s(\boldsymbol{\kappa}_s) - C_s(\boldsymbol{\kappa}_s). \quad (18)$$

### C. Nonlinear Constrained Optimization Problem

Consider a set of slices $\mathcal{S}$, the resource provisioning problem for all slices $s \in \mathcal{S}$, which accounts for uncertain slice user demands and tries to limit the impact on background services, can be formulated as

**Problem 1: Nonlinear Constrained Optimization**

$$\underset{\boldsymbol{\kappa}=\{\boldsymbol{\kappa}_s\}_{s\in\mathcal{S}}}{\text{maximize}} \sum_{s\in\mathcal{S}} E_s(\boldsymbol{\kappa}_s) = \sum_{s\in\mathcal{S}} \left(I_s x_s(\boldsymbol{\kappa}_s) - C_s(\boldsymbol{\kappa}_s)\right),$$

subject to $(6, 7, 10\text{–}12, 14, 15)$.

Solving Problem 1 is complex due to the need to evaluate $p_s(\boldsymbol{\kappa}_s)$ using (13) in the verification of the constraint (14). Section V introduces a simpler method to solve Problem 1.

## V. Reduced-Complexity Slice Resource Provisioning

In this section, a parameterized ILP formulation of Problem 1 is introduced. The main idea is to replace the constraints (10, 11, 14) involving probabilities related to random variables describing the aggregate user demands and best-effort services by linear deterministic constraints.

### A. Linear Inequality Constraints for the PSP

For a given slice $s \in \mathcal{S}$ and for each $v \in \mathcal{N}_s$, $vw \in \mathcal{E}_s$, and $n \in \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$, let

$$\overline{R}_{s,n}(v, \gamma_s) = \mu_{s,n}(v) + \gamma_s \sigma_{s,n}(v), \quad (19)$$

$$\overline{R}_{s,\mathrm{b}}(vw, \gamma_s) = \mu_{s,\mathrm{b}}(vw) + \gamma_s \sigma_{s,\mathrm{b}}(vw), \quad (20)$$

be the target aggregate user demand, depending on some parameter $\gamma_s > 0$. For an assignment $\boldsymbol{\kappa}_s$ that satisfies

$$\sum_i \kappa_s(i,v)\, r_{s,n}(v) \geqslant \overline{R}_{s,n}(v, \gamma_s), \forall n, v, \quad (21)$$

$$\sum_{ij} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant \overline{R}_{s,\mathrm{b}}(vw, \gamma_s), \forall vw, \quad (22)$$

and (6, 7, 12), the PSP defined in (13) can be evaluated as

$$p_s(\gamma_s) = \Pr\Big\{\overline{R}_{s,n}(v,\gamma_s) \geqslant R_{s,n}(v), \forall v, n, \; \overline{R}_{s,\mathrm{b}}(vw,\gamma_s) \geqslant R_{s,\mathrm{b}}(vw), \forall vw\Big\}, \quad (23)$$

which is independent of $\boldsymbol{\kappa}_s$. If $p_s(\gamma_s) \geqslant \underline{p}_s$, the MI-SLA relative to the PSP is satisfied. The main difficulty is now to determine the smallest value of $\gamma_s$ such that $p_s(\gamma_s) \geqslant \underline{p}_s$, since the larger $\gamma_s$, the more difficult the satisfaction of (21) and (22).

Using (3), one has

$$p_s(\gamma_s) = \sum_{k=1}^{m} p_k \int_{\overline{\mathcal{R}}(\gamma_s)} f\left(\mathbf{x}, k\boldsymbol{\mu}, k^2\boldsymbol{\Gamma}\right) \mathrm{d}\mathbf{x}, \quad (24)$$

where $\overline{\mathcal{R}}(\gamma_s) = \left\{\mathbf{x} \in \mathbb{R}^{n_{\mathrm{R}}} \,|\, \mathbf{x} \leqslant \overline{\mathbf{R}}(\gamma_s)\right\}$ and

$$\overline{\mathbf{R}}(\gamma_s) = \Big(\overline{R}_{s,\mathrm{c}}(v_1,\gamma_s), \overline{R}_{s,\mathrm{m}}(v_1,\gamma_s), \ldots \; \overline{R}_{s,\mathrm{b}}(v_1v_2,\gamma_s), \ldots\Big)^{\top}$$

of size $n_{\mathrm{R}}$. Since the pmf of the number of users $p_k$, $k = 1, \ldots, m$ has been assumed to be known, the value of $\gamma_s$ such that $p_s(\gamma_s) = \underline{p}_s$ may be obtained by the bisection search methods, see, *e.g.*, [30]. The multidimensional integral in (24) can be evaluated using a quasi-Monte Carlo integration algorithm presented in [31]. An example of the evolution of $p_s(\gamma_s)$ as function of $\gamma_s$ for a given slice $s$ of Type 1 is depicted in Figure 5, using the simulation setting described in Section VI-A.

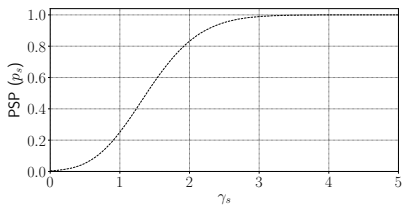



Fig. 5. Evolution of $p_s$ as function of $\gamma_s$.

### B. Linear Inequality Constraints for the IP

For each $i \in \mathcal{N}$, $ij \in \mathcal{E}$, and $n \in \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$, consider the following target level of background service demands

$$\overline{B}_n(i, \gamma_{\mathrm{B}}) = \mu_{\mathrm{B},n}(i) + \gamma_{\mathrm{B}} \sigma_{\mathrm{B},n}(i), \quad (25)$$

$$\overline{B}_{\mathrm{b}}(ij, \gamma_{\mathrm{B}}) = \mu_{\mathrm{B},\mathrm{b}}(ij) + \gamma_{\mathrm{B}} \sigma_{\mathrm{B},\mathrm{b}}(ij), \quad (26)$$

where $\gamma_{\mathrm{B}} > 0$ is some tuning parameter. For an assignment $\boldsymbol{\kappa} = \{\boldsymbol{\kappa}_s\}_{s\in\mathcal{S}}$ that satisfies

$$\sum_{s,v} \kappa_s(i,v)\, r_{s,n}(v) \leqslant a_n(i) - \overline{B}_n(i,\gamma_{\mathrm{B}}), \forall n, i, \quad (27)$$

$$\sum_{s,vw} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij) - \overline{B}_{\mathrm{b}}(ij,\gamma_{\mathrm{B}}), \forall ij, \quad (28)$$

and (6, 7, 12), the IP defined in (8) can be evaluated as follows

$$\begin{aligned} p_n^{\mathrm{im}}(i) &= \Pr\left\{B_n(i) \geqslant \overline{B}_n(i,\gamma_{\mathrm{B}})\right\} \\ &= \int_{\overline{B}_n(i,\gamma_{\mathrm{B}})}^{+\infty} f\left(x;\mu_{\mathrm{B},n}(i),\sigma^2_{\mathrm{B},n}(i)\right)\mathrm{d}x \\ &= 1 - \int_{-\infty}^{\overline{B}_n(i,\gamma_{\mathrm{B}})} f\left(x;\mu_{\mathrm{B},n}(i),\sigma^2_{\mathrm{B},n}(i)\right)\mathrm{d}x \\ &= 1 - \Phi(\gamma_{\mathrm{B}}), \end{aligned} \quad (29)$$

where $\Phi$ is the cumulative distribution function (CDF) of the zero-mean, unit-variance normal distribution. Similarly, the IP defined in (9) can also be evaluated as

$$\begin{aligned} p_{s,\mathrm{b}}^{\mathrm{im}}(ij) &= \Pr\left\{B_{\mathrm{b}}(ij) \geqslant \overline{B}_{\mathrm{b}}(ij,\gamma_{\mathrm{B}})\right\} \\ &= 1 - \Phi(\gamma_{\mathrm{B}}). \end{aligned} \quad (30)$$

Both (29) and (30) are independent of $\boldsymbol{\kappa}_s$, $\forall s \in \mathcal{S}$. To satisfy the impact constraints imposed by (8, 9), $\gamma_{\mathrm{B}}$ has to be chosen such that

$$1 - \Phi(\gamma_{\mathrm{B}}) \leqslant \overline{p}^{\mathrm{im}} \Leftrightarrow \gamma_{\mathrm{B}} \geqslant \Phi^{-1}\left(1-\overline{p}^{\mathrm{im}}\right). \quad (31)$$

Since the larger $\gamma_{\mathrm{B}}$, the more difficult the satisfaction of (27) and (28), the optimal $\gamma_{\mathrm{B}}$ would be $\gamma_{\mathrm{B}} = \Phi^{-1}\left(1-\overline{p}^{\mathrm{im}}\right)$.

### C. ILP Formulation for Multiple Slice Provisioning

Considering the linear inequality constraints introduced in Sections V-A and V-B, instead of the inequality constraints involving probabilities in Problem 1, one may introduce the following relaxed parameterized formulation of Problem 1.

**Problem 2: ILP for Multiple Slice Resource Provisioning**

$$\underset{\{\boldsymbol{d},\boldsymbol{\kappa}\}=\{d_s,\boldsymbol{\kappa}_s\}_{s\in\mathcal{S}}}{\text{maximize}} \quad \sum_{s\in\mathcal{S}} \left(I_s d_s - C_s(\boldsymbol{\kappa}_s)\right),$$

subject to (12) and

$$\sum_i \kappa_s(i,v)\, r_{s,n}(v) \geqslant \overline{R}_{s,n}(v,\gamma_s)\, d_s, \forall s, n, v, \quad (32)$$

$$\sum_{ij} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant \overline{R}_{s,\mathrm{b}}(vw,\gamma_s)\, d_s, \forall s, vw, \quad (33)$$

$$\sum_{s,v} \kappa_s(i,v)\, r_{s,n}(v) \leqslant a_n(i) - \overline{B}_n(i,\gamma_{\mathrm{B}}), \forall n, i, \quad (34)$$

$$\sum_{s,vw} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij) - \overline{B}_{\mathrm{b}}(ij,\gamma_{\mathrm{B}}), \forall ij. \quad (35)$$

Problem 2 is now an ILP. The binary variables $d_s$, $s \in \mathcal{S}$ indicate whether resources are actually provisioned for slice $s$. When $d_s = 0$, the minimization of the provisioning cost $C_s(\boldsymbol{\kappa}_s)$ imposed by the objective function of Problem 2 will enforce $\kappa_s = 0$ in (32) and (33). Remind that $\gamma_s$ and $\gamma_{\mathrm{B}}$ are evaluated by bisection search, as discussed in Sections V-A and V-B, before solving Problem 2.

### D. ILP Formulation for Slice-by-Slice Provisioning

The number of variables involved in the solution of Problem 2 introduced in Section V-C may be relatively large when several slices have to be considered jointly. This section introduces a reduced-complexity formulation where provisioning is performed slice-by-slice.

Consider the set of $n_{\mathrm{s}}$ slices $\mathcal{S} = \{s_1, \ldots, s_{n_{\mathrm{s}}}\}$ for which resources have to be provisioned. Assume that the the slice-by-slice resource provisioning has been performed up to slice $s_{\ell-1}$, $1 \leqslant \ell - 1 < n_{\mathrm{s}}$. A successful provisioning is indicated by $d_s = 1$, whereas $d_s = 0$ indicates that resources cannot be provisioned for slice $s$, due, *e.g.*, to the non-satisfaction of the PSP or IP constraints, or to the lack of infrastructure resources. The corresponding assignment is represented by $\boldsymbol{\kappa}_s$, $s \in \{s_1, \ldots, s_{\ell-1}\}$.

Slice $s_\ell$ is now considered. In the provisioning for slice $s_\ell$, one has simply to account for the amount of infrastructure resources left after the provisioning of all slices $s \in \{s_1, \ldots, s_{\ell-1}\}$. Consequently, only (34) and (35) have to be updated to get the following new ILP formulation for slice-by-slice resource provisioning.

**Problem 3: ILP for Slice-by-Slice Resource Provisioning**

$$\underset{d_{s_\ell},\boldsymbol{\kappa}_{s_\ell}}{\text{maximize}} \quad I_{s_\ell} d_{s_\ell} - C_{s_\ell}(\boldsymbol{\kappa}_{s_\ell}),$$

subject to (12) and

$$\sum_i \kappa_{s_\ell}(i,v)\, r_{s,n}(v) \geqslant \overline{R}_{s_\ell,n}(v,\gamma_{s_\ell})\, d_{s_\ell}, \forall n, v, \quad (36)$$

$$\sum_{ij} \kappa_{s_\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant \overline{R}_{s_\ell,\mathrm{b}}(vw,\gamma_{s_\ell})\, d_{s_\ell}, \forall vw, \quad (37)$$

$$\sum_v \kappa_{s_\ell}(i,v)\, r_{s,n}(v) \leqslant a_n(i) - \overline{B}_n(i,\gamma_{\mathrm{B}}) - \sum_{s\in\{s_1,\ldots,s_{\ell-1}\}} \kappa_s(i,v)\, r_{s,n}(v)\, d_s, \forall n, i, \quad (38)$$

$$\sum_{vw} \kappa_{s_\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij) - \overline{B}_{\mathrm{b}}(ij,\gamma_{\mathrm{B}}) - \sum_{s\in\{s_1,\ldots,s_{\ell-1}\}} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw)\, d_s, \forall ij. \quad (39)$$

The order in which the provisioning is performed is important. One may choose to provision the slices by decreasing income $I_s$. An other possibility is to perform a greedy search, starting with the slice $s^1 \in \mathcal{S}$ for which $I_s d_s - C_s(\boldsymbol{\kappa}_s)$ is maximized, when deployed alone. Then, assuming that resources have been provisioned for $s^1$, one may search $s^2 \in \mathcal{S} \setminus \{s^1\}$ maximizing $I_s d_s - C_s(\boldsymbol{\kappa}_s)$ with the remaining infrastructure resources, and so forth.

### E. Slice Resource Provisioning Algorithms

From the suboptimal methods introduced in Sections V-C and V-D, we propose four uncertainty-aware slice resource provisioning variants, JP-B and JP considering the joint approach introduced in Problem 2; SP-B and SP considering the sequential approach introduced in Problem 3.

The JP-B and SP-B approaches account for the impact of provisioning on background services, whereas the JP and SP approaches do not take those services into account. This is obtained by setting $\overline{B}_n(i,\gamma_{\mathrm{B}}) = 0, \forall n, i$ and $\overline{B}_{\mathrm{b}}(ij,\gamma_{\mathrm{B}}) = 0, \forall ij$ in Problems 2 and 3. The SP and JP algorithms have been borrowed from [26], where slice resource demands are considered to be deterministic. Compared to the original approach in [26], the SP and JP approaches in this paper account additionally for the uncertainties of slice resource demand. Moreover, while the main decision variables in the original SP and JP approaches in [26] are the proportion of available resources in the infrastructure, here, the main decision variables are the number of SFC instances for which resources have to be provisioned for a future deployment.

These four provisioning variants are summarized in Algorithms 1 and 2. Each variant requires the solution of one or several ILPs, whose complexity is exponential in the number of variables in the worst case. The JP-B and JP approaches (Algorithm 1) require the solution of a single ILP, with $|\mathcal{N}_s| + |\mathcal{S}|(1 + |\mathcal{N}||\mathcal{N}_s| + |\mathcal{E}||\mathcal{E}_s|)$ variables and $|\mathcal{S}|(|\mathcal{N}||\mathcal{E}_s| + 3|\mathcal{N}_s| + |\mathcal{E}_s|) + 3|\mathcal{N}| + |\mathcal{E}|$ constraints. The SP-B and SP approaches (Algorithm 2) split the work into $|\mathcal{S}|$ subproblems, each of which involves $|\mathcal{N}_s| + |\mathcal{N}||\mathcal{N}_s| + |\mathcal{E}||\mathcal{E}_s| + 1$ variables and $|\mathcal{S}||\mathcal{N}||\mathcal{E}_s| + 3|\mathcal{N}_s| + |\mathcal{E}_s| + 3|\mathcal{N}| + |\mathcal{E}|$ constraints. Therefore, each subproblem in the sequential approaches (SP-B and SP) implies $|\mathcal{S}|$ times less variables than the joint variants (JP-B and JP). Due to the exponential complexity of each problem, solutions for the sequential variants may be obtained faster than with the joint variants. Section VI presents a more detailed performance comparison of these variants.

**Algorithm 1:** Joint Approaches (JP-B and JP)

**Input:** $\mathcal{G} = (\mathcal{N}, \mathcal{E}), \mathcal{S}, \{\mathcal{G}_s, s \in \mathcal{S}\}$
**Output:** $\widehat{\boldsymbol{\kappa}} = \{\widehat{\boldsymbol{\kappa}}_s\}_{s\in\mathcal{S}}$

**switch** provisioning_variant **do**
  **case** JP-B *(background traffics taken into account)* **do**
    Evaluate $\widehat{\boldsymbol{\kappa}} = \arg\max_{\boldsymbol{\kappa}} \sum_{s\in\mathcal{S}} (I_s d_s - C_s(\boldsymbol{\kappa}_s))$, subject to: (12), (32)–(35);
  **case** JP *(background traffic ignored)* **do**
    Evaluate $\widehat{\boldsymbol{\kappa}} = \arg\max_{\boldsymbol{\kappa}} \sum_{s\in\mathcal{S}} (I_s d_s - C_s(\boldsymbol{\kappa}_s))$, subject to: (12), (32)–(33), and

$$\sum_{s,v} \kappa_s(i,v)\, r_{s,n}(v) \leqslant a_n(i), \forall n, i,$$

$$\sum_{s,vw} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij), \forall ij;$$

**Algorithm 2:** Sequential Approaches (SP-B and SP)

**Input:** $\mathcal{G} = (\mathcal{N}, \mathcal{E}), \mathcal{S}, \{\mathcal{G}_s, s \in \mathcal{S}\}$
**Output:** $\widehat{\boldsymbol{\kappa}} = \{\widehat{\boldsymbol{\kappa}}_s\}_{s\in\mathcal{S}}$

**switch** provisioning_variant **do**
  **case** SP-B *(background traffic taken into account)* **do**
    **for** $\ell = 1, \ldots, |\mathcal{S}|$ **do**
      Evaluate $\widehat{\boldsymbol{\kappa}}_{s_\ell} = \arg\max_{d_{s_\ell}, \boldsymbol{\kappa}_{s_\ell}} (I_{s_\ell} d_{s_\ell} - C_{s_\ell}(\boldsymbol{\kappa}_{s_\ell}))$, subject to: (12), (36)–(39);
  **case** SP *(background traffics ignored)* **do**
    **for** $\ell = 1, \ldots, |\mathcal{S}|$ **do**
      Evaluate $\widehat{\boldsymbol{\kappa}}_{s_\ell} = \arg\max_{d_{s_\ell}, \boldsymbol{\kappa}_{s_\ell}} (I_{s_\ell} d_{s_\ell} - C_{s_\ell}(\boldsymbol{\kappa}_{s_\ell}))$, subject to: (12), (36)–(37), and

$$\sum_{v} \kappa_{s_\ell}(i,v)\, r_{s,n}(v) \leqslant a_n(i) - \sum_{s\in\{s_1,\ldots,s_{\ell-1}\}} \kappa_s(i,v)\, r_{s,n}(v)\, d_s, \forall n, i,$$

$$\sum_{vw} \kappa_{s_\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij) - \sum_{s\in\{s_1,\ldots,s_{\ell-1}\}} \kappa_s(ij,vw)\, r_{s,\mathrm{b}}(vw)\, d_s, \forall ij;$$

## VI. Evaluation

In this section, one evaluates via simulations the performance of the four variants (JP-B, SP-B, JP, and SP) of the provisioning algorithms described in Section V. The simulation setup is described in Section VI-A. All numerical results presented in Section VI-B have been performed with the CPLEX MILP solver interfaced with MATLAB.

### A. Simulation Conditions

*1) Infrastructure Topology:* The infrastructure network is generated from a $k$-ary fat tree topology, as in [23, 32]. A typical fat-tree topology is depicted in Figure 6 when $k = 2$. The leaf nodes represent the Remote Radio Heads (RRHs). The other nodes represent the edge, regional, and central data centers. Infrastructure nodes and links provide a given amount of computing, storage, and possibly wireless resources $(a_{\mathrm{c}}, a_{\mathrm{m}}, a_{\mathrm{w}})$, expressed in number of CPUs, Gbytes, and Gbps, depending on the layer they are located. The cost of using each resource of the infrastructure network is $c_n(i) = 1$, $\forall n \in \{\mathrm{c}, \mathrm{m}, \mathrm{w}\}$, $c_{\mathrm{f}}(i) = 65, 60, 55, 50$ for respectively central, regional, edge, RRH nodes, and $c_{\mathrm{b}}(ij) = 1$, $\forall ij \in \mathcal{E}$.

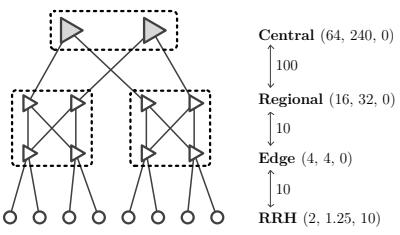



Fig. 6. Description of a $k$-ary fat-tree infrastructure network with $k = 2$; Nodes provide a given amount of computing $a_{\mathrm{c}}$, memory $a_{\mathrm{m}}$, and wireless $a_{\mathrm{w}}$ resources expressed in number of used CPUs, Gbytes, and Gbps; Links are able to transmit data at a rate $a_{\mathrm{b}}$ expressed in Gbps.

*2) Background Services:* At each infrastructure node $i \in \mathcal{N}$ and link $ij \in \mathcal{E}$, the resources consumed by best-effort

background services follow a normal distribution with mean and standard deviation equal to respectively $20\,\%$ and $5\,\%$ percent of the available resource at that node and link, *i.e.*, $\mu_{\mathrm{B},n}(i) = 0.2a_n(i)$, $\sigma_{\mathrm{B},n}(i) = 0.05a_n(i)$, $\forall i \in \mathcal{N}$, $\forall n \in \{\mathrm{c},\mathrm{m},\mathrm{w}\}$, and $\mu_{\mathrm{B,b}}(ij) = 0.2a_{\mathrm{b}}(ij)$, $\sigma_{\mathrm{B,b}}(ij) = 0.05a_{\mathrm{b}}(ij)$, $\forall ij \in \mathcal{E}$.

*3) Slice Resource Demand (S-RD):* Three types of slices are considered.

- Slices of type 1 aim to provide an HD video streaming service at average rate of4 Mbps for VIP users, *e.g.*, in a stadium. The number of users follows a binomial distribution $\mathcal{B}(300, 0.9)$;
- Slices of type 2 are dedicated to provide an SD video streaming service at average rate of 2 Mbps. The number of users follows a binomial distribution $\mathcal{B}(1000, 0.8)$;
- Slices of type 3 aim to provide a video surveillance and traffic monitoring service at average rate of 1 Mbps for 100 cameras, *e.g.*, installed along a highway.

The first two slice types address a video streaming service, and thus have the same function architecture with 3 virtual functions: a virtual Video Optimization Controller (vVOC), a virtual Gateway (vGW), and a virtual Base Band Unit (vBBU). The third slice type consists of five virtual functions: a vBBU, a vGW, a virtual Traffic Monitor (vTM), a vVOC, and a virtual Intrusion Detection Prevention System (vIDPS).

As detailed in Section III-B, the resource requirements for the various SFCs that will have to be deployed within a slice are aggregated within an S-RD graph that mimics the SFC-RD graph. S-RD nodes and links are characterized by the aggregated resource needed to support the targeted number of users. Details of each resource type as well as the associated U-RD, SFC-RD, and S-RD graph are given in Table IV. Numerical values in Table IV have been adapted from [33].

### B. Results

This section illustrates the performance of the various resource provisioning variants, in terms of: utilization of infrastructure nodes and links, maximal probability of impact $p^{\mathrm{im}}$ on the background services at every infrastructure node and link, provisioning cost, total earnings of the InP, and number of impacted nodes and links, *i.e.*, the number of nodes $i \in \mathcal{N}$ such that $\exists n \in \{\mathrm{c},\mathrm{m},\mathrm{w}\}$ $p_n^{\mathrm{im}}(i) > \overline{p}^{\mathrm{im}}$ and links $ij \in \mathcal{E}$ such that $p_{\mathrm{b}}^{\mathrm{im}}(ij) > \overline{p}^{\mathrm{im}}$.

We first evaluate the effect on the slice resource provisioning of the constraint related to the impact on background services. This is done by comparing the two variants SP-B and SP in Section VI-B1 and VI-B2, considering (*i*) a single and (*ii*) a multiple slice provisioning problem. In Section VI-B3, the performance of the four proposed resource provisioning variants (JP, SP, JP-B, and SP-B) are compared. Finally, the benefits of the uncertainty-aware slice resource provisioning approach in terms of improved probability of successful provisioning are illustrated in Section VI-B4.

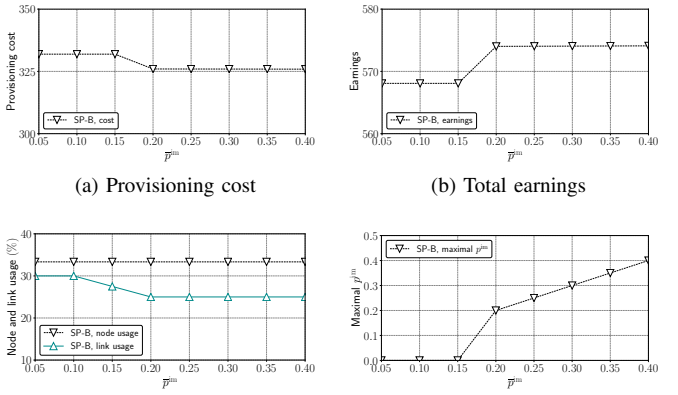
*1) Provisioning of a Single Slice:* Table II shows the performance of two variants SP-B and SP for the provisioning of a single slice of Type 1, where $\underline{p}_s = 0.99$ and $\overline{p}^{\mathrm{im}} = 0.1$. Recall that these two variants differ from each other in whether the impact on background service is considered or not.

It is observed that the SP variant, which does not account for impact on background services, has a lower link usage and provisioning cost, and yields a higher earning for the InP than that of the SP-B variant. Nevertheless, as expected, the SP variant has a higher impact on background services, with maximal impact probability of $0.58$ exceeding the maximum tolerated impact probability $\overline{p}^{\mathrm{im}}$ at one infrastructure node, as summarized in Table II.

TABLE II
PERFORMANCE OF SP-B AND SP ON SINGLE SLICE PROVISIONING

| Criteria | SP-B | SP |
|---|---|---|
| Node usage | 33% | 33% |
| Link usage | 28% | 25% |
| Maximal $p^{\mathrm{im}}$ | 1.26e-4 | 0.58 |
| Provisioning cost | 332 | 326 |
| Total earnings | 568 | 574 |
| #impacted nodes | 0 | 1 |
| #impacted links | 0 | 0 |

The way $\overline{p}^{\mathrm{im}}$ affects the performance of SP-B is shown in Figures 7a-7d, when $\underline{p}_s = 0.99$ and $\overline{p}^{\mathrm{im}}$ ranges from $0.05$ to $0.4$. One observes that, the higher $\overline{p}^{\mathrm{im}}$, the lower the provisioning cost and the higher earnings for the InP. This is due to the fact that, with higher $\overline{p}^{\mathrm{im}}$, it is easier to provision slices with limited resources. This can be observed in the decrease of link usage in Figure 7c. On the other hand, the impact probability $p^{\mathrm{im}}$ is always kept under the threshold $\overline{p}^{\mathrm{im}}$ imposed by the InP, as shown in Figure 7d.




(a) Provisioning cost (b) Total earnings

(c) Node and link usage (d) Evolution of $p^{\mathrm{im}}$ as a function of $\overline{p}^{\mathrm{im}}$

Fig. 7. Performance of the SP-B approach on single slice provisioning problem with different values of $\overline{p}^{\mathrm{im}}$, in terms of (a) provisioning cost, (b) total earnings, (c) node and link utilization, and (d) maximal impact probability $p^{\mathrm{im}}$.

*2) Provisioning Several Slices of a Same Type:* Now, considering 10 slices of type 1, the SP-B and SP variants are compared in terms of acceptance rate, *i.e.*, percentage of slices that have been successfully provisioned (given by $\sum_{s\in\mathcal{S}} \frac{x_s}{|\mathcal{S}|}$) and number of impacted nodes and links (for which the impact probability is larger than $\overline{p}^{\mathrm{im}}$), for different value of $\underline{p}_s$, see Figure 8a. The tolerated impact probability $\overline{p}^{\mathrm{im}}$ is set to $0.1$. As expected, when $\underline{p}_s$ increases, the acceptance rate decreases

for both approaches. The SP approach, which does not account for background services, always has a higher acceptance rate and earnings compared to the SP-B approach, but its impact on the background services is significantly larger. Using the SP approach, provisioned resources are concentrated on a fewer amount of nodes and links. Consequently, the background services running on such nodes and links may then be affected.

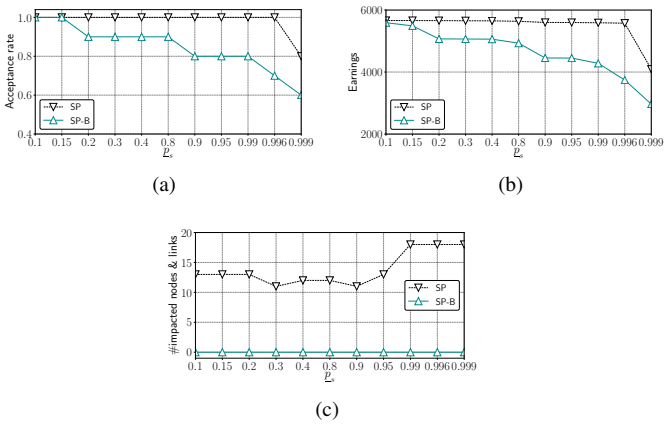



Fig. 8. Performance of the SP-B and SP approaches on the provisioning of multiple slices of one type, with different values of $\underline{p}_s$, in terms of (a) acceptance rate, (b) total earnings, and (c) number of impacted nodes and links.

*3) Provisioning of Several Slices of Different Types:* The performance of the four variants is illustrated in this section, when resources for 2 to 8 slices of three different types have to be provisioned. The number of slices of each type and their associated $\underline{p}_s$ are detailed in Table III. The impact probability threshold $\overline{p}^{\text{im}}$ is set to 0.1 in all scenarios.

TABLE III
NUMBER OF SLICES OF EACH TYPE AS A FUNCTION OF $|\mathcal{S}|$

| Case | #Type 1 | #Type 2 | #Type 3 |
|---|---|---|---|
| $\|\mathcal{S}\| = 2$ | 1 | 1 | 0 |
| $\|\mathcal{S}\| = 4$ | 2 | 1 | 1 |
| $\|\mathcal{S}\| = 6$ | 2 | 2 | 2 |
| $\|\mathcal{S}\| = 8$ | 3 | 2 | 3 |

The use of infrastructure nodes and links is shown in Figures 9a and 9b. The joint provisioning approaches (JP and JP-B) require a reduced amount of nodes and links compared to the sequential schemes (SP and SP-B). Moreover, accounting for the impact on background services requires, again, provisioning resources on more nodes and links.

Figure 9c shows the provisioning costs obtained with the various approaches. One observes that the JP variant yields the smallest cost among all variants, as it aims at finding an optimal solution for all slices, without considering the impact probability, contrary to the JP-B variant. This leads to the highest earnings for the InP, as shown in Figure 9d.

The total number of impacted nodes and links is shown in Figure 9e. The JP-B and SP-B variant have no impacted nodes or links, whereas the provisioning performed by the JP and SP approaches significantly impact the background services. The SP variant has a higher impact on the background services, due to the higher utilization of infrastructure nodes and links, as shown in Figures 9a and 9b. Consequently, provisioning with the JP and SP approaches can significantly deteriorate the performance of background services for which no provisioning is performed. This impact increases with the number of considered slices.

From the InP perspective, the use of impact-unaware variants (JP and SP) maximizes its earning but violates background services at a significant number of infrastructure nodes and links. This may necessitate to reconfigure those background services. On contrary, by using the impact-aware variants (JP-B and SP-B), the InP can provision slices and preserve a tolerable impact on the background services. The price to be paid is somewhat degraded node and link utilization efficiency and a higher provisioning cost compare to the impact-aware variants, leading to a lower earnings for the InP. For instance, when provisioning for 4 slices, the JP-B variant uses around 72% of the total infrastructure nodes to aggregate resources needed to support the slices, while only 66.7% of the nodes are employed by the JP method, leading to a reduction of 3.5% of total earnings, as depicted in Figures 9a and 9d.

As expected, the sequential provisioning methods (SP-B and SP) perform better in terms of computing time than the joint approaches (JP-B and JP). Increasing the number of slices leads to an increase of the cardinality of the sets of variables $\boldsymbol{d}$ and $\boldsymbol{\kappa}$, and therefore increases the computing time. In sequential provisioning, slices are considered successively. There is only a very small difference (usually less than 5%) in computing time between the SP-B and SP approaches and between the JP-B and JP approaches.

*4) Benefits of the Uncertainty-Aware Slice Resource Provisioning:* In this section, we show the benefits of the proposed uncertainty-aware slice resource provisioning method, in terms of deployment efficiency, when considering the SFC embedding. Slice resource provisioning is performed for a *single slice.* JP-B and SP-B behave thus similarly. This is also the case for JP and SP.

The JP-B is compared to a variant of JP, which does not account for the uncertainty of slice resource demands. Problem 2 is solved in the latter case with a slice resource demand corresponding to its mean value. This is done by choosing $\gamma_s = 0$ in (32) and (33). Once provisioning is performed, the SFC embedding step is realized and a randomly generated number of users following the same distribution as that used in the provisioning process is considered. One gets an uncertainty-aware provisioning and embedding solution (UPE) and a deterministic provisioning and embedding solution (DPE). These solutions are compared in terms of satisfaction of the user demands.

A single slice of type 1 is considered. The U-RD, SFC-RD, and infrastructure parameters used in the previous parts

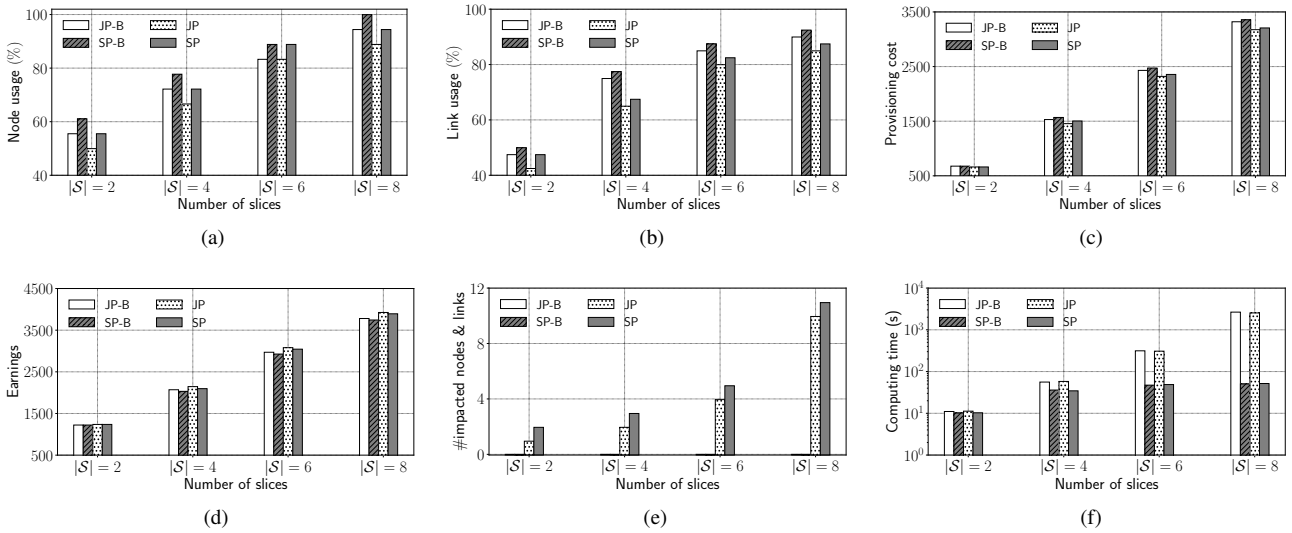



Fig. 9. Performance comparison of 4 variants in terms of utilization of infrastructure nodes (a), infrastructure links (b), provisioning costs (c), total earnings (d), number of impacted nodes and links (e), and computing time (f).

of Section VI-B are used again. For the S-RD, the number of users associated to the slice follows a binomial distribution $\mathcal{B}(m, p)$, where $m$ is fixed to 300, and $p$ varies. One thousand independent drawings of the number of users are performed. The number of SFCs that have to be actually deployed can be then deduced from the resulting number of users. SFCs can only be deployed when enough resources have been provisioned for the slice. Finally, the SFC acceptance rate, *i.e.*, the number of provisioned SFCs divided by the number of required SFCs, of the UPE and DPE solution is compared.

Figure 10 shows the average, minimum, and maximum SFC acceptance rates, when the probability $p$ of the binomial distribution ranges from $0.4$ to $0.9$. The UPE solution provides a successful deployment of all SFCs. The DPE solution, which does not take into account the uncertainties of slice resource demands, cannot ensure the deployment for all SFCs, when not enough resources have been provisioned. In addition, as expected, when $p$ is higher, *i.e.*, the slice resource demands become less uncertain, DPE yields a higher acceptance rate, with a smaller gap between the minimum, and maximum SFC acceptance rates, as shown in Figure 10.

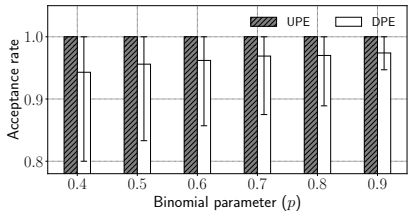



Fig. 10. Performance comparison of the UPE and DPE solutions in terms of SFC acceptance rate.

## VII. Conclusions

This paper investigates a resource provisioning method for network slicing robust to a partly unknown number of users whose resource demands are uncertain. Adopting the point of view of the InP, one tries to maximize its earnings, while providing a probabilistic guarantee that the slice resource demands are fulfilled. In addition to that, the proposed resource provisioning method is performed to keep the impact on the background services under a threshold imposed by the InP.

The uncertainty-aware slice resource provisioning is formulated as a nonlinear constrained optimization problem. A parameterized ILP formulation is then proposed. With the ILP formulation, four variants (JP, SP, JP-B, and SP-B) are introduced, for the solution of the provisioning problem for multiple slices jointly or sequentially, without or with consideration of the impact of provisioning on background services.

The impact-limiting variants (JP-B, and SP-B) have a controlled impact on the background services. The JP and SP variants, on the other hand, do not account for the impact on those services. Consequently, all resources of several infrastructure nodes and links may be consumed when using the JP and SP variants. This may impose a reconfiguration of background services. The price to be paid for the InP when performing impact-limiting variants are lower earnings.

Moreover, due to the exponential worst-case complexity in the number of variables of the ILP formulation, as expected, sequential approaches are shown to better scale to a larger number of slices. The price to be paid by the sequential approaches is a somewhat degraded node and link utilization, a higher provisioning cost, and lower earnings, compared to the joint approaches. To further reduce the complexity, column generation (CG) approaches could be used, see, *e.g.*, [34], where CG has been used to the relaxation of ILP-based SFC embedding problems.

The benefit of our proposed uncertainty-aware slice resource provisioning for network slicing over a deterministic provi-

TABLE IV
PARAMETERS OF U-RD, SFC-RD, AND S-RD GRAPHS

**Type** 1: HD video streaming at 4 Mbps. $N_s \sim \mathcal{B}(300, 0.9)$, $I_s = 900$, $\underline{p}_s = 0.99$

| *Node* | $(\mu_{s,\mathrm{c}}, \sigma_{s,\mathrm{c}})$ | $(\mu_{s,\mathrm{m}}, \sigma_{s,\mathrm{m}})$ | $(\mu_{s,\mathrm{w}}, \sigma_{s,\mathrm{w}})$ | $(r_\mathrm{c}, r_\mathrm{m}, r_\mathrm{w})$ | *Link* | $(\mu_\mathrm{b}, \sigma_\mathrm{b})$ | $r_{s,\mathrm{b}}$ |
|---|---|---|---|---|---|---|---|
| vVOC | $(5.4, 0.54)$ e-3 | $(1.5, 0.15)$ e-2 | — | $(0.29, 0.81, 0)$ | vVOC→vGW | $(4, 0.4)$ e-3 | 0.22 |
| vGW | $(9.0, 0.90)$ e-4 | $(5.0, 0.50)$ e-4 | — | $(0.05, 0.03, 0)$ | vGW→vBBU | $(4, 0.4)$ e-3 | 0.22 |
| vBBU | $(8.0, 0.80)$ e-4 | $(5.0, 0.50)$ e-4 | $(4, 0.4)$ e-3 | $(0.04, 0.03, 0.2)$ | | | |

**Type 2**: SD video streaming at 2 Mbps. $N_s \sim \mathcal{B}(1000, 0.8)$, $I_s = 1000$, $\underline{p}_s = 0.95$

| *Node* | $(\mu_{s,\mathrm{c}}, \sigma_{s,\mathrm{c}})$ | $(\mu_{s,\mathrm{m}}, \sigma_{s,\mathrm{m}})$ | $(\mu_{s,\mathrm{w}}, \sigma_{s,\mathrm{w}})$ | $(r_\mathrm{c}, r_\mathrm{m}, r_\mathrm{w})$ | *Link* | $(\mu_\mathrm{b}, \sigma_\mathrm{b})$ | $r_{s,\mathrm{b}}$ |
|---|---|---|---|---|---|---|---|
| vVOC | $(1.1, 0.11)$ e-3 | $(7.5, 0.75)$ e-3 | — | $(0.17, 1.20, 0)$ | vVOC→vGW | $(2, 0.2)$ e-3 | 0.32 |
| vGW | $(1.8, 0.18)$ e-4 | $(2.5, 0.25)$ e-4 | — | $(0.03, 0.04, 0)$ | vGW→vBBU | $(2, 0.2)$ e-3 | 0.32 |
| vBBU | $(0.8, 0.08)$ e-4 | $(2.5, 0.25)$ e-4 | $(2, 0.2)$ e-3 | $(0.01, 0.04, 0.3)$ | | | |

**Type 3**: Video surveillance and traffic monitoring at 1 Mbps. $N_s = 50$, $I_s = 800$, $\underline{p}_s = 0.9$

| *Node* | $(\mu_{s,\mathrm{c}}, \sigma_{s,\mathrm{c}})$ | $(\mu_{s,\mathrm{m}}, \sigma_{s,\mathrm{m}})$ | $(\mu_{s,\mathrm{w}}, \sigma_{s,\mathrm{w}})$ | $(r_\mathrm{c}, r_\mathrm{m}, r_\mathrm{w})$ | *Link* | $(\mu_\mathrm{b}, \sigma_\mathrm{b})$ | $r_{s,\mathrm{b}}$ |
|---|---|---|---|---|---|---|---|
| vBBU | $(2.0, 0.20)$ e-4 | $(1.3, 0.13)$ e-4 | $(1, 0.1)$ e-3 | $(0.4, 0.25, 2)$ e-2 | vBBU→vGW | $(1, 0.1)$ e-3 | 0.02 |
| vGW | $(9.0, 0.90)$ e-4 | $(1.3, 0.13)$ e-4 | — | $(0.018, 0.003, 0)$ | vGW→vTM | $(1, 0.1)$ e-3 | 0.02 |
| vTM | $(1.1, 0.11)$ e-3 | $(1.3, 0.13)$ e-4 | — | $(0.266, 0.003, 0)$ | vTM→vVOC | $(1, 0.1)$ e-3 | 0.02 |
| vVOC | $(5.4, 0.54)$ e-3 | $(3.8, 0.38)$ e-3 | — | $(0.108, 0.080, 0)$ | vVOC→vIDPS | $(1, 0.1)$ e-3 | 0.02 |
| vIDPS | $(1.1, 0.11)$ e-2 | $(1.3, 0.13)$ e-4 | — | $(0.214, 0.003, 0)$ | | | |

sioning approach, such as that in [26], is also illustrated. Numerical results show that an SFC embedding using the provisioned resources provided by the proposed method yields a higher acceptance rate than that using a deterministic provisioning method.

In this paper, uncertainties related to the fluctuation of user demands and the background services have been taken into account for the slice resource provisioning. An extension to this work is to design an adaptive resource provisioning mechanism to cope with dynamic changes in the network infrastructure and slice resource demands [35]. To that end, one possible approach is to let the InP, if necessary, update the already provisioned resources for some slices during their lifetime.

## REFERENCES


[1] 5G Americas, "Network Slicing for 5G Networks & Services," *White Paper*, 2016.
[2] IETF, "Network Slicing Architecture," *Internet-Draft*, pp. 1–8, 2017.
[3] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G Network Slicing Using SDN and NFV: A Survey of Taxonomy, Architectures and Future Challenges," *Computer Networks*, vol. 167, 2020.
[4] C. Liang and F. R. Yu, "Wireless Network Virtualization: A Survey, Some Research Issues and Challenges," *IEEE Commun. Surveys Tuts.*, pp. 1–24, 2014.
[5] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker, "Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks," in *IEEE Commun. Mag.*, vol. 55, no. 5, 2017, pp. 72–79.
[6] GSM Alliance, "An Introduction to Network Slicing," *White Paper*, 2017.
[7] M. Richart, J. Baliosian, J. Serrat, and J. L. Gorricho, "Resource Slicing in Virtual Wireless Networks: A Survey," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 462–476, 2016.
[8] N. Huin, B. Jaumard, and F. Giroire, "Optimization of Network Service Chain Provisioning," in *Proc. IEEE ICC*, 2017.
[9] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, "Resource Allocation for Network Slicing in 5G Telecommunication Networks: A Survey of Principles and Models," *IEEE Network*, vol. 33, no. 6, pp. 172–179, 2019.
[10] G. Wang, G. Feng, W. Tan, S. Qin, W. Ruihan, and S. Sun, "Resource Allocation for Network Slices in 5G with Network Resource Pricing," in *Proc. IEEE GLOBECOM*, 2017, pp. 1–6.
[11] T. Trinh, H. Esaki, and C. Aswakul, "Quality of Service Using Careful Overbooking for Optimal Virtual Network Resource Allocation," in *Proc. ECTI*, 2011, pp. 296–299.
[12] S. Coniglio, A. M. Koster, and M. Tieves, "Virtual Network Embedding Under Uncertainty: Exact And Heuristic Approaches," in *Proc. DRCN*. IEEE, 2015, pp. 1–8.
[13] S. Mireslami, L. Rakai, M. Wang, and B. H. Far, "Dynamic Cloud Resource Allocation Considering Demand Uncertainty," *IEEE Trans. on Cloud Comput.*, vol. 7161, no. c, pp. 1–1, 2019.
[14] X. Zhang and J. Wang, "Heterogeneous Statistical QoS Provisioning Over 5G Mobile Wireless Networks," *IEEE Network*, vol. 28, no. 6, pp. 46–53, 2014.
[15] K. Xiong, S. Samuel Rene Adolphe, G. O. Boateng, G. Liu, and G. Sun, "Dynamic Resource Provisioning and Resource Customization for Mixed Traffics in Virtualized Radio Access Network," *IEEE Access*, vol. 7, pp. 115 440–115 453, 2019.
[16] G. Sun, K. Xiong, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and W. Jiang, "Autonomous Resource Provisioning and Resource Customization for Mixed Traffics in Virtualized Radio Access Network," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2454–2465, 2019.
[17] H. Alshaer and H. Haas, "Bidirectional LiFi Attocell Access Point Slicing Scheme," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 909–922, 2018.
[18] A. Fendt, C. Mannweiler, L. C. Schmelz, and B. Bauer, "An Efficient Model for Mobile Network Slice Embedding under Resource Uncertainty," in *Proc. ISWCS*, 2019, pp. 602–606.
[19] A. Baumgartner, T. Bauschert, F. D'Andreagiovanni, and V. S. Reddy, "Towards Robust Network Slice Design under Correlated Demand Uncertainties," in *Proc. ICC*, 2018, pp. 1–7A.
[20] D. Bertsimas and M. Sim, "Robust Discrete Optimization and Network Flows," *Mathematical Programming*, vol. 98, no. 1-3, pp. 49–71, 2003.
[21] T. Bauschert and V. S. Reddy, "Genetic Algorithms for the Network Slice Design Problem Under Uncertainty," in *Proc. GECCO Companion*, 2019, pp. 360–361.
[22] R. Wen, G. Feng, J. Tang, T. Q. Quek, G. Wang, W. Tan, and S. Qin, "On Robustness of Network Slicing for Next-Generation Mobile Networks," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 430–444, 2019.
[23] R. Riggio, A. Bradai, D. Harutyunyan, T. Rasheed, and T. Ahmed, "Scheduling Wireless Virtual Networks Functions," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 2, pp. 240–252, 2016.
[24] P. Vizarreta, M. Condoluci, C. M. Machuca, T. Mahmoodi, and W. Kellerer, "QoS-driven Function Placement Reducing Expenditures in NFV Deployments," in *Proc. IEEE ICC*, 2017.
[25] Q.-T. Luu, S. Kerboeuf, A. Mouradian, and M. Kieffer, "A Coverage-Aware Resource Provisioning Method for Network Slicing," *IEEE/ACM Trans. Netw.*, vol. 28, no. 6, pp. 2393–2406, 2020.
[26] Q.-T. Luu, M. Kieffer, A. Mouradian, and S. Kerboeuf, "Aggregated Resource Provisioning for Network Slices," in *Proc. IEEE GLOBECOM*, Abu Dhabi, UAE, 2018, pp. 1–6.
[27] J. Wang, K. L. Wright, and K. Gopalan, "XenLoop: A Transparent High Performance Inter-VM Network Loopback," *Cluster Comput.*, vol. 12, no. 2 SPEC. ISS., pp. 141–152, 2009.
[28] B. Tschaen, Y. Zhang, T. Benson, S. Banerjee, J. Lee, and J. M. Kang, "SFC-Checker: Checking the Correct Forwarding Behavior of Service Function Chaining," in *Proc. IEEE NFV-SDN*. IEEE, 2016, pp. 134–140.
[29] C. Jiang, G. Han, J. Lin, G. Jia, W. Shi, and J. Wan, "Characteristics of Co-Allocated Online Services and Batch Jobs in Internet Data Centers: A Case Study from Alibaba Cloud," *IEEE Access*, vol. 7, pp. 22 495–22 508, 2019.
[30] R. L. Burden and J. Douglas Faires, *Numerical Analysis*, 9th ed. Brooks/Cole, Cengage Learning, 2011.
[31] A. Genz, "Numerical Computation of Rectangular Bivariate and Trivariate Normal and t Probabilities," *Statistics and Computing*, vol. 14, no. 3, pp. 251–260, 2004.
[32] N. Bouten, R. Mijumbi, J. Serrat, J. Famaey, S. Latre, and F. De Turck, "Semantically Enhanced Mapping Algorithm for Affinity-Constrained Service Function Chain Requests," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 2, pp. 317–331, 2017.
[33] M. Savi, M. Tornatore, and G. Verticale, "Impact of Processing-Resource Sharing

on the Placement of Chained Virtual Network Functions," in *Proc. IEEE NFV-SDN*, 2016, pp. 191–197.
[34] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On Dynamic Service Function Chain Deployment and Readjustment," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 3, pp. 543–553, 2017.
[35] H. Alshaer, "An Overview of Network Virtualization and Cloud Network as a Service," *Internation Journal of Network Management*, vol. 25, pp. 1–30, 2014.

**Quang-Trung Luu** is currently a Ph.D candidate at Nokia Bell Labs and the Signal and Systems Laboratory (L2S) of the University of Paris-Sud and CentraleSupélec, France. His research focuses on the optimization of resource management in wireless networks, in particular on key enabling technologies for 5G systems such as network slicing.

**Sylvaine Kerboeuf** received the M.S. degree in physics and the Ph.D. degree in solid-state physics from the University of Paris-Sud, Orsay France, in 1991 and 1994, respectively, and the Ph.D. degree in superconductivity from the Centre National d'Etude des Télécommunications, France Telecom, Paris, France. She joined the Research and Innovation Department, Alcatel-Lucent Bell Labs, Nozay, France, where she was involved in research projects on optoelectronics for several years. In 2004, she joined a project involved in radio access networks and focusing on fourth generation discontinuous networks and on caching technology. She is currently a Senior Researcher in the Wireless Program with Nokia Bell Labs. Her current research interests include software defined network architecture, network slicing and end-to-end orchestration of micro-services for 5G networks.

**Michel Kieffer** (M'02, SM'07) received the Ph.D. degree in control theory from the University of Paris XI, Orsay, France, in 1999. He is a Full Professor in signal processing for communications with the University of Paris-Sud and a Researcher with the Laboratoire des Signaux et Systèmes (L2S), Gif-sur-Yvette, France. Since 2009, he is a part-time Invited Professor with the Laboratoire Traitement et Communication de l'Information, Télécom ParisTech, Paris, France. He is coauthor of more than 150 contributions in journals, conference proceedings, or books. He is one of the coauthors of the books *Applied Interval Analysis* (Springer-Verlag, 2001) (this book was translated in Russian in 2005) and *Joint Source-Channel Decoding: A Cross-Layer Perspective With Applications in Video Broadcasting* (Academic, 2009). His research interests are in signal processing for multimedia, communications, and networking; distributed source coding; network coding; joint source-channel coding and decoding techniques; and joint source-network coding. Applications are mainly in the reliable delivery of multimedia contents over wireless channels. He is also interested in guaranteed and robust parameter and state bounding for systems described by nonlinear models in a bounded-error context. Prof. Kieffer was a junior member of the *Institut Universitaire de France* from 2011 to 2016. He serves as an Associate Editor of SIGNAL PROCESSING since 2008 and of the IEEE TRANSACTIONS ON COMMUNICATIONS from 2012 to 2016.