Admission Control and Resource Reservation for Prioritized Slice Requests with Guaranteed SLA under Uncertainties

Quang-Trung Luu, Sylvaine Kerboeuf, Michel Kieffer

Admission Control and Resource Reservation for Prioritized Slice Requests with Guaranteed SLA under Uncertainties

Quang-Trung Luu, Sylvaine Kerboeuf, and Michel Kieffer

***Abstract*—Network slicing has emerged as a key concept in 5G systems, allowing Mobile Network Operators (MNOs) to build isolated logical networks (slices) on top of shared infrastructure networks managed by Infrastructure Providers (InP). Network slicing requires the assignment of infrastructure network resources to virtual network components at slice activation time and the adjustment of resources for slices under operation. Performing these operations *just-in-time*, on a best-effort basis, comes with no guarantee on the availability of enough infrastructure resources to meet slice requirements.**

This paper proposes a prioritized admission control mechanism for concurrent slices based on an infrastructure resource reservation approach. The reservation accounts for the dynamic nature of slice requests while being robust to uncertainties in slice resource demands. Adopting the perspective of an InP, reservation schemes are proposed that maximize the number of slices for which infrastructure resources can be granted while minimizing the costs charged to the MNOs. This requires the solution of a max-min optimization problem with a non-linear cost function and non-linear constraints induced by the robustness to uncertainties of demands and the limitation of the impact of reservation on background services. The cost and the constraints are linearized and several reduced-complexity strategies are proposed to solve the slice admission control and resource reservation problem. Simulations show that the proportion of admitted slices of different priority levels can be adjusted by a differentiated selection of the delay between the reception and the processing instants of a slice resource request.

***Index Terms*—Network slicing, resource reservation, prioritized slice processing, slice admission control, uncertainty, wireless network virtualization, 5G**

I. Introduction

In the fifth-generation communication systems [1, 2], Network slicing (NS) aims at replacing the traditional *one-size-fits-all* network architecture. NS may address diverging requirements imposed by verticals [3] while reducing operational costs [4, 5], thanks to its ability to provide higher network flexibility. NS exploits network virtualization to elastically allocate and reallocate infrastructure resources tailored to the time-varying needs of various applications [6, 7]. With NS, multiple *slices*, *i.e.*, *customized*, *isolated*, and *service-dedicated* end-to-end logical networks, can be established and operated simultaneously on a shared physical infrastructure network, provided by one or several Infrastructure Providers (InPs) [8].

Several authors, see, *e.g.*, [9–12], have recently considered the resource allocation problem raised by network slicing. This problem involves efficiently assigning infrastructure network resources to virtual network components at (or just before) slice activation time and dynamically adjusting these resources for slices under operation to maximize resource utilization and minimize operating costs. With such *just-in-time* slice management, it is difficult to guarantee the availability of enough infrastructure resources at the deployment time and during the lifetime of a slice. Slice admission control mechanisms have therefore been proposed to prioritize, accept, possibly delay, or even reject demands for slices [13–18].

Network slicing with guaranteed satisfaction of some Service Level Agreement (SLA) is facilitated by adopting an infrastructure resource *reservation* approach, as specified by [19, 20], rather than the *just-in-time* slice resource management approach considered, *e.g.*, in [21, 22]. Resource reservation aims at determining whether enough infrastructure resources are (or will be) available to satisfy a slice resource request (feasibility check). The actual resource allocation may be done later, once the reservation is successful and the slice request has been granted.

Nevertheless, slice resource reservation raises several challenging problems: Slice requests are submitted by Service Providers (SPs) at different time instants, with various activation delays, life durations, and user demands fluctuating with time. The variety of services supported by slices induces very different Quality of Service (QoS) requirements [7]. Moreover, various constraints may be imposed by the different network segments on which slices have to be deployed [12, 14]. For example, coverage constraints are imposed by slices involving the radio access network [23]. Additionally, several sources of uncertainty have to be considered in a reservation approach, *e.g.*, the number of slice users, the hardly predictable user locations [24], and the time-varying per-user resource requirements. Consequently, enough infrastructure resources should be reserved for each slice to guarantee an adequate QoS specified in the SLA and provide robustness against uncertainties. Too many infrastructure resources should not be reserved too, in order to reduce costs and leave resources to concurrent slices.

Quang-Trung Luu is with Nokia Bell Labs, 91620 Nozay, France, and also with Laboratoire des Signaux et Systèmes, Paris-Saclay University - CNRS - CentraleSupélec, 91192 Gif-sur-Yvette, France (e-mail: quangtrung.luu@centralesupelec.fr).

S. Kerboeuf is with NSSR Lab, Nokia Bell Labs, 91629 Nozay, France (e-mail: sylvaine.kerboeuf@ nokia-bell-labs.com).

M. Kieffer is with Laboratoire des Signaux et Systèmes, Paris-Saclay University - CNRS - CentraleSupélec, 91192 Gif-sur-Yvette, France (e-mail: michel.kieffer@l2s.centralesupelec.fr).

Contributions. This work considers a model where SPs submit slice service requests (possibly largely before their activation) to some Mobile Network Operator (MNO). The MNO and the InP are considered as two separate entities, possibly belonging to the same company. The MNO evaluates the amount of resources required to operate each slice efficiently and submits slice resource reservation requests to an InP. The InP has to determine whether it is able to book, as much in advance as possible, enough infrastructure resources to ensure that the MNO will have access to enough and properly located infrastructure resources with service characteristics as stated in some SLA. The first contribution of this paper is a slice admission control and resource reservation framework able to provide a probabilistic guarantee related to SLA satisfaction. Overbooking of resources by the InP is allowed, as in [14], but the probability of SLA non-satisfaction is bounded, with a controlled bound. The proposed method to reserve infrastructure resources for concurrent slices accounts for the dynamic nature of slice requests (including their arrival, activation, and deactivation times while being robust against the uncertainties in the number of users and the amount of resource employed by a typical user). We adopt the perspective of an InP and propose an approach where the InP tries to find the resource reservation scheme which maximizes the amount of slices for which the reservation is successful while minimizing the resource operation costs charged to the MNOs. The InP decides then to accept or reject each slice resource request.

The processing of a slice request submitted largely before its activation can be anticipated by the MNO who may check its feasibility in advance with one or several InPs. The second contribution of this paper is a process that anticipates more or less this processing depending on the priority level of the slice requests. Slices can be admitted, possibly largely before their activation, when enough infrastructure resources are reserved for meeting their QoS requirements. When this condition is not satisfied, the slice resource request is not granted and MNOs may address their slice resource request to alternative InPs. The proposed approach is consistent with the 3GPP views of the management aspects of network slicing [19, 20]. The proposed slice reservation and admission control takes place in the *network environment preparation* task of the *preparation* phase. In this phase, the design and capacity planning of network slices, the on-boarding and evaluation of required network functions, and the reservation of infrastructure resources have to be done before the creation and activation of network slice instances, which belong to the *commissioning* and *operation* phases (see Figure 1).

The third contribution of this paper is to formulate the processing of concurrent slice resource reservation requests and their admission control as a max-min optimization problem. Its solution provides the InP maximum earnings for granted slice requests and is also appropriate for the MNO in terms of charged reservation costs. Reservation and adaptation costs are introduced to charge variations in slice resource demands. A nonlinear objective function is then obtained. The robustness to uncertainties in the demands and the limitation of the impact of reservation on background services is considered as in [22]. This introduces nonlinear constraints in the max-min optimization problem. After linearizing the cost and some of the constraints, several reduced-complexity reservation strategies are proposed to solve the problem of slice resource reservation with dynamic resource demands.

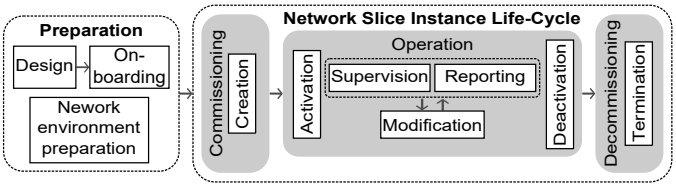


Fig. 1. 3GPP view on network slicing managements aspects [19].

The remainder of this paper is organized as follows. Section II presents some related work. In Section III, we describe the problem statement, in which the system model is detailed. Section IV introduces the proposed approaches to efficiently reserve resources for concurrent slices, while being robust to the dynamic nature of slice requests and to the uncertainties related to infrastructure and slice parameters. Numerical results are then provided in Section VI to evaluate the performance of the proposed reservation and admission control approaches. Finally, Section VII draws some conclusions and perspectives.

II. Related Work

In network slicing, a slice is composed 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.

Many papers have addressed the problem of slice/SFC resource allocation with uncertain or time-varying requirements and available physical resources, see, *e.g.*, [9, 10, 25–28]. Conservative strategies allocating resources considering worst-case peak traffic conditions [9, 10] are costly and usually lead to an inefficient utilization of resources. In [26–28], an adjustable safety factor is considered to give a probabilistic guarantee of resource availability, *e.g.*, ensuring that every slice benefits from sufficient infrastructure resources with a certain probability. Inspired by the approach in [28], [22] has extended the approach introduced in [29] to account for the impact of resource reservation on background services.

The dynamic nature of slice/SFC requests is taken into account in [21, 30–32]. In [30], a dynamic resource allocation for SFCs is investigated. The deployment of newly arrived SFCs and readjustment of in-service SFCs are taken into account. An Integer Linear Program (ILP) formulation is used to address the dynamic deployment problem, aiming at minimizing the cost of VNF deployment and migration. A pre-calculation of all possible routing paths has to be performed in advance, which requires some computational effort before using the deployment algorithm. In [21], the adaptive adjustment of allocated resources of each slice is enabled after each decision time period (slicing time). An hybrid slice reconfiguration framework is introduced in [31]. The slice can be reconfigured either within small time intervals for individual slices, or

within large time intervals to readjust resource allocation of multiple slices. A deep-learning approach is adopted in [32] for dynamic slice resource allocation, with the aim to maximize the long-term revenue of the network provider. Uncertainties related to the slice requests and occupation time are considered. Nevertheless, slices are regarded as a whole, *i.e.*, not made up of multiple elements (*e.g.*, VNFs), which somewhat over-simplifies the problem of slice resource allocation.

Slice Admission control (SAC) mechanisms have been developed recently [13–18] to address issues related to the unavailability of enough resources to satisfy all slice requests. A yield-driven approach is proposed in [14], assuming that the MNO manages the infrastructure resources and decides to accept or reject slice requests in order to maximize the revenue obtained from the SPs. Resource overbooking is allowed and a penalty in case of non-satisfaction is considered in the optimization process. Nevertheless, there is no control on the level of satisfaction of the SLA requirements imposed by the service provider due to the overbooking possibility. Consequently, a slice request may be granted with a penalty very close to the revenue provided by the slice. Slice requests in [14] are assumed to be processed as they arrive, without considering any explicit prioritization between requests. Prioritized processing results only implicitly from the differences in the revenue and penalty associated to the various requests.

In [15], SAC is formulated as a boolean linear program and a two-step sub-optimal algorithm based on variants of the knapsack problem is proposed to alleviate the complexity. Admission is done for slices with highest profit considering first the RAN and *aggregated* core network resources. In the second step, the core network resources are considered without any aggregation to determine whether a slice deployment is possible.

In [17], SAC and resource allocation are performed jointly, to minimize the power consumption of cloud nodes and network bandwidth of the infrastructure provider. Transmission delay is accounted in the slice SLA. Some elastic variables are introduced in an ILP formulation to extend the bounds on some constraints. They help determining when resources may be lacking, in which case slices are rejected starting from those with the highest requirements in terms of resource. Nevertheless, the dynamics of slice requests (time of arrival, slice duration) and the variation of slice resource demands during their life time are not considered in [15] and [17].

The dynamics of slice requests is considered by [18] in the SAC problem. If not accepted, a request is queued for being potentially served later. The case of impatient tenants, who may leave their queues before being served, is taken into account. Nevertheless, neither the dynamics of resource demands within each slice, nor the activation time of a slice are accounted for. Moreover, infrastructure resources of each type are fully aggregated. As opposed to [17] and to our work, none of the details about the structure of the slice and of the infrastructure are taken into account in the resource model. Consequently, the proposed mechanism does not allow to reserve nor allocate resource to the slice in addition to admission control.

Online SAC is considered in [13] and [16] leveraging on machine learning approaches. The aim is to maximize the revenue of the InP while guaranteeing the SLAs of the admitted slices. Both papers focus on radio resources of base stations. In [13], two different types of slices are considered to account for elastic and inelastic traffic. An admissibility region is determined first, indicating the maximum number of slices that the system can support without breaking the SLAs. Both works formalize the admission control problem into a semi MDP and derived the optimal policy obtained when the request arrival parameters are known. The approach has a high computation cost and is off-line (requires system parameters to be known *a priori*). An alternative Q-learning approach is proposed in [13] to adapt to changing environments while achieving close to optimal performance. In [16], a deep reinforcement learning method is developed to overcome the scalability issue of the Q-learning approach.

These works consider tenants submitting slice requests for an immediate deployment, contrary to our work, where slice requests are assumed to be submitted for an immediate or future deployment, which permits the development of a resource reservation strategy.

III. Problem Statement and Notations

A typical network slicing system involves several entities: one or several InPs, MNOs, and SPs (also known as slice tenants), as depicted in Figure 2 [4]. InPs own and manage the wireless and wired infrastructure such as the cell sites, the fronthaul and backhaul networks, and data centers. Section III-A details the considered model for the infrastructure network. MNOs lease resources from InPs to set up and manage slices. SPs then exploit the slices supplied by MNOs and provide their customers with the required services running within the slices. An SP forwards a slice service booking request to an MNO within an SLA denoted SM-SLA in what follows.

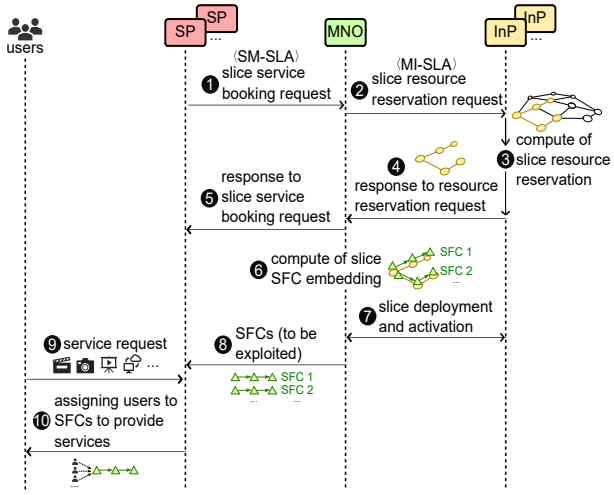


Fig. 2. Network slicing entities and their SLA-based relationships.

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, see Section III-B1. In this paper, one considers SM-SLAs composed of: (*i*) 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 typical user/device employs it, and (*iii*) a target Service Satisfaction Probability (SSP) $\underline{p}^{\text{sp}}$. Since the target number of users is usually uncertain, it is described by a random variable with a known probability mass function (pmf). The MNO has to provide a slice able to serve user demands with a probability of at least $\underline{p}^{\text{sp}}$. In addition, several time intervals may be considered in the SM-SLA, intervals over each of which the target number of users, the service characteristics, and the probability of satisfaction are assumed invariant. They may vary from a time interval to the next one. These time intervals translate, *e.g.*, day and night variations of user demands and last between tens of minutes to hours. It is 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. Following the 3GPP approach (cf. Figure 4.8.1 of [19]), the MNO is in charge of the slice admission control via assessing the feasibility of the SP's request.

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 how a given user/device consumes the slice (SFCs) resources. 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.

In what follows, one assumes that the MNO and the InP are two distinct entities (possibly belonging to a single stakeholder but having their domain responsibilities, like, *e.g.*, two business divisions of the same organization). In this case, the MNO submits resource reservation requests to one or several InPs upon the arrival of slice booking requests. Section III-B provides a model of the requests for slice resource reservation sent by the MNO to an InP and of their associated costs. Each resource reservation request contains the description of the resource demand characteristics (the SSP constraint is translated into deterministic requirements, see Section IV-E) as well as the slice priority class as part of an SLA between them (MI-SLA), see Section III-B2.

Each InP, considering the various slice resource reservation requests received during some time interval, tries to maximizes the number of slices for which the reservation can be satisfied. Costs induced by the variation with the time of the resource reservation request are taken into account by the InP, see Section III-B3. A resource reservation request for a slice is considered satisfied when *i)* enough resources are available to meet a target resource requirements and *ii)* the Impact Probability (IP) on other best-effort services running on the infrastructure network remains below some threshold $\overline{p}^{\text{im}}$, see Section III-C. The slice priority level is taken into account when processing the reservation requests. The InP answers positively or negatively to a reservation request. In the latter case, the MNO may contact alternative InPs, or, when no InP has enough available resources, the MNO may reject the slice service booking request from the SP or ask the SP to update its slice service request (SM-SLA negotiation).

Table I summarizes the main notations introduced in this paper.

TABLE I
TABLE OF MAIN NOTATIONS

INFRASTRUCTURE NETWORK	
$\mathcal{G}$	Infrastructure network graph, $\mathcal{G} = (\mathcal{N}, \mathcal{E})$
$\mathcal{N}$	Set of infrastructure nodes
$\mathcal{E}$	Set of infrastructure links
Υ	Set of node resource types, $\Upsilon = \{\text{c}, \text{m}, \text{w}\}$
$a_n(i)$	Available resource of type $n \in \Upsilon$ at node i
$a_{\text{b}}(ij)$	Available bandwidth of link ij
$c_n(i)$	Per-unit cost of resource of type $n \in \Upsilon$ for node i
$c_{\text{b}}(ij)$	Per-unit cost for link ij
$c_{\text{f}}(i)$	Fixed cost for using node i
$c_{\text{a}}(i)$	Reservation adaptation cost at node i
TEMPORAL NOTATIONS	
$\mathcal{P}_k$	Processing time interval in time slot k
T	Duration of a time slot
εT	Processing duration (of $\mathcal{P}_k$)
SLICE REQUESTS AND RESOURCE DEMANDS	
$\mathcal{G}_s$	SFC graph of slice s, $\mathcal{G}_s = (\mathcal{N}_s, \mathcal{E}_s)$
$\mathcal{N}_s$	Set of virtual network functions
$\mathcal{E}_s$	Set of virtual links
P_s^{c}	Priority class
$P_{s,k}$	Priority level at time k
$\mathcal{K}_s$	Slice active interval, $\mathcal{K}_s = \left[k_s^{\text{on}}, k_s^{\text{off}}\right]$
$\boldsymbol{r}_s$	Vector of resource demands of an SFC
$\boldsymbol{U}_{s,k}$	Vector of resource demands of a typical user in time slot k
$\boldsymbol{R}_{s,k}$	Vector of aggregate resource demands in time slot k
$\boldsymbol{B}_k$	Vector of resources consumed by background services in time slot k
$\overline{U}$, $\overline{R}$, $\overline{B}$	Mean value of $\boldsymbol{U}$, $\boldsymbol{R}$, $\boldsymbol{B}$
$\widetilde{U}$, $\widetilde{R}$, $\widetilde{B}$	Standard deviation of $\boldsymbol{U}$, $\boldsymbol{R}$, $\boldsymbol{B}$
$\underline{p}_s^{\text{sp}}$	Required service satisfaction probability
$\overline{p}^{\text{im}}$	Slice impact probability threshold (w.r.t. background services)
$\mathcal{S}_k$	Slices requests received before $(k+1)T - \varepsilon T$
$\mathcal{R}_k$	Slices requests processed during $\mathcal{P}_k$

A. Network Model

In this paper, to simplify presentation, one considers an infrastructure network owned by a single InP. The infrastructure network managed by the considered InP is represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. $\mathcal{N}$ 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 (loop-back links) of the infrastructure network.

Each infrastructure node $i \in \mathcal{N}$ is characterized by its computing $a_{\text{c}}(i)$, memory $a_{\text{m}}(i)$, and wireless $a_{\text{w}}(i)$ resources. For each node i, the InP charges the MNO a fixed cost $c_{\text{f}}(i)$ for node disposal (paid for each slice using node i), and per-unit variable costs $c_n(i)$, $n \in \Upsilon = \{\text{c}, \text{m}, \text{w}\}$, 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 is characterized by its bandwidth $a_{\text{b}}(ij)$, and an associated per-unit bandwidth cost $c_{\text{b}}(ij)$. Several distinct VNFs of the same slice may be deployed on a given infrastructure node. When communication between these VNFs is required, an

internal (loop-back) infrastructure link $ii \in \mathcal{E}$ can be used at each node $i \in \mathcal{N}$, as in [33], in the case of interconnected virtual machines (VMs) deployed on the same host. In that case, the InP charges the MNO per-unit bandwidth cost $c_{\mathrm{b}}(ii)$.

B. Slice Resource Reservation Requests and Adaptation Costs

1) Request Arrivals: One considers that time is slotted into slots of constant duration T (typically of few tens of minutes), which represents the time unit considered for the slice resource reservation duration in the booking calendar. The slot of index $k \in \mathbb{N}$ lasts over the time interval $[kT, (k+1)T[$. One considers that the slice lifetime spans over one or several time slots of duration T. Resources have to be reserved so as to be compliant with the variations of the number of users and of their demands during the slice lifetime. The service characteristics are assumed stable over each time slot, and may vary from one time slot to the next.

Let t_s be the time instant at which the reservation request for a slice s is received by the InP. This slice is also characterized by the index k_s^{on} of the time slot at the beginning of which it has to be activated (put into service), and the index k_s^{off} of the time slot at the end of which it has to be deactivated. Thus, the slice s is active over the time interval $\left[k_s^{\mathrm{on}}T, \left(k_s^{\mathrm{off}}+1\right)T\right[$. Figure 3 depicts an example of arrivals of slice resource reservation requests, as well as the time slots over which the corresponding services have to be active.

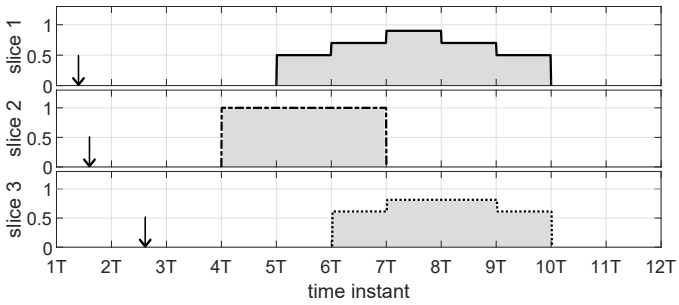


Fig. 3. Arrivals of slice resource reservation requests as a function of time; Black arrows represent the arrival times t_s of each request; The types of slices are illustrated by different plot line styles; The slice resource demands evolve with time; Peak demands have been normalized.

2) Slice Resource Demand: A demand for resources for a slice s is defined on the basis of the translation of the SM-SLA between an SP and an MNO. A priority class P_s^{c} determines the priority level with which the resource demand has to be processed. As in [34], the type of service provided by slice s is used to identify the type of SFC, *i.e.*, the ordered set of VNFs involved in s. We consider that a slice is devoted to a single type of service supplied by a given type of SFC. Several SFCs of the same type may have to be deployed so as to satisfy user demands within the slice. 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 a VNF, and each virtual link $vw \in \mathcal{E}_s$ represents the connection between virtual nodes v and w.

Based on $\mathcal{G}_s$, one introduces the vectors $\boldsymbol{r}_s$, $\mathbf{U}_{s,k}$, and $\mathbf{R}_{s,k}$, respectively representing the *resource demands* of a single SFC (SFC-RD), of a typical user (U-RD) during time slot k, and the aggregate resource demand of the users of slice s (S-RD) during time slot k.

The deterministic SFC-RD vector

$$\boldsymbol{r}_s = \left[r_{s,n}(v), r_{s,\mathrm{b}}(vw)\right]^{\top}_{n\in\Upsilon,(v,vw)\in\mathcal{G}_s} \tag{1}$$

gathers the computing ($r_{s,\mathrm{c}}(v)$), memory ($r_{s,\mathrm{m}}(v)$), wireless ($r_{s,\mathrm{w}}(v)$), and bandwidth ($r_{s,\mathrm{b}}(vw)$) resource requirements of the VNFs $v \in \mathcal{N}_s$ and the virtual links $vw \in \mathcal{E}_s$ of a single SFC. The vector $\boldsymbol{r}_s$ is assumed to be time invariant, as it characterizes the resources which need to be allocated to run an instance of the considered SFC. In the considered reservation context, $\boldsymbol{r}_s$ also represents the *maximum* amount of reserved resources that will be made available to the considered SFC.

Each user of slice s is assumed to consume a random amount of the resources of an SFC of that slice. The random vector

$$\mathbf{U}_{s,k} = \left[U_{s,n,k}(v), U_{s,\mathrm{b},k}(vw)\right]^{\top}_{n\in\Upsilon,(v,vw)\in\mathcal{G}_s} \tag{2}$$

of U-RD represents the resource demands of a single user of slice s during time slot k. $U_{s,n,k}(v)$, $n \in \Upsilon$, and $U_{s,\mathrm{b},k}(vw)$ are the random amounts of employed resources of VNF v and of virtual link vw. In addition, the resources consumed by various users are represented by independently and identically distributed random vectors. Minor variations of the user resource demand within time slot k are accounted for by the probability distribution characterizing $\mathbf{U}_{s,k}$.

The random S-RD vector

$$\mathbf{R}_{s,k} = \left[R_{s,n,k}(v), R_{s,\mathrm{b},k}(vw)\right]^{\top}_{n\in\Upsilon,(v,vw)\in\mathcal{G}_s} \tag{3}$$

gathers $R_{s,n,k}(v)$, $n \in \Upsilon$, and $R_{s,\mathrm{b},k}(vw)$, the aggregate amount of resources employed by a random number $N_{s,k}$ of independent users of slice s during time slot k. Minor variations of the number of users within time slot k are as well captured by the probability mass function characterizing $N_{s,k}$.

The probability distributions characterizing $\mathbf{U}_{s,k}$, $N_{s,k}$, and consequently $\mathbf{R}_{s,k}$ depend on the time slot index k, to represent possible large changes in the U-RD or in the number of users of slice s in successive time slots.

One considers, for a typical user and during a given time slot k, that for each virtual node $v \in \mathcal{N}_s$, the resource demands of different types $n \in \Upsilon$ are correlated. A correlation may also exist between the demands for resources of the same type among virtual nodes. Finally, the resulting traffic demands between nodes is usually also correlated with the resource demands for a given virtual node, as reported in [35]. Considering the U-RD vector $\mathbf{U}_{s,k}$, one assumes that the elements $U_{s,n,k}(v)$, $\forall n \in \Upsilon$, and $U_{s,\mathrm{b},k}(vw)$ are normally distributed during the time slot k. $\mathbf{U}_{s,k}$ thus follows a multivariate normal distribution with probability density

$$f\left(\mathbf{x}; \boldsymbol{\mu}_{s,k}, \boldsymbol{\Gamma}_{s,k}\right) = \left(2\pi\right)^{-\frac{1}{2}\mathrm{card}(\mathbf{U}_{s,k})} \left|\boldsymbol{\Gamma}_{s,k}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_{s,k}\right)^{\top}\boldsymbol{\Gamma}_{s,k}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_{s,k}\right)}, \tag{4}$$

where $\mathrm{card}(\mathbf{U}_{s,k})$ is the number of elements of $\mathbf{U}_{s,k}$,

$$\boldsymbol{\mu}_{s,k} = \left[\overline{U}_{s,n,k}(v), \overline{U}_{s,\mathrm{b},k}(vw)\right]^{\top}_{(v,vw)\in\mathcal{G}_s,n\in\Upsilon} \tag{5}$$

is its mean value and and $\mathbf{\Gamma}_{s,k}$ is its covariance matrix, with diagonal elements

$$\text{diag}\left(\mathbf{\Gamma}_{s,k}\right) = \left[\widetilde{U}^2_{s,n,k}\left(v\right), \widetilde{U}^2_{s,\text{b},k}\left(vw\right)\right]^\top_{(v,vw)\in\mathcal{G}_s, n\in\Upsilon}, \quad (6)$$

and off-diagonal elements representing the correlation between different types of resource demands. One has thus

$$U_{s,n,k}(v) \sim \mathcal{N}\left(\overline{U}_{s,n,k}\left(v\right), \widetilde{U}^2_{s,n,k}\left(v\right)\right), n \in \Upsilon, \text{ and}$$
$$U_{s,\text{b},k}\left(vw\right) \sim \mathcal{N}\left(\overline{U}_{s,\text{b},k}\left(vw\right), \widetilde{U}^2_{s,\text{b},k}\left(vw\right)\right). \quad (7)$$

The probability that the number of users $N_{s,k}$ to be supported by slice s in the k-th time slot is equal to η is

$$p_{s,k,\eta} = \Pr\left(N_{s,k} = \eta\right), \eta \in \mathbb{N}. \quad (8)$$

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

$$g\left(\mathbf{x}, \boldsymbol{\mu}_{s,k}, \mathbf{\Gamma}_{s,k}\right) = \sum_{\eta=0}^{\infty} p_{s,k,\eta} f\left(\mathbf{x}, \eta\boldsymbol{\mu}_{s,k}, \eta^2\mathbf{\Gamma}_{s,k}\right). \quad (9)$$

3) Adaptation Costs to Request Variations: During the lifetime of a slice, the amount of required slice resources may evolve from one time slot to another. An increase of the required resources may impact the resource allocation scheme by requiring more infrastructure resources to be allocated. Compared to a situation where the resource allocation is static for the whole lifespan of a slice, this induces more operations to be performed on the network infrastructure (assignment or re-assignment of resources, launching virtual machines or containers on which VNFs will be operated) and results in additional costs to the InP. The anticipation of those allocation adaptation costs are then considered when processing the slice resource reservation request. A cost $c_\text{a}\left(i\right)$ for each unit *increase* of the amount of instances (*i.e.*, virtual machines or containers) of a VNF between two time slots is assumed to be charged by the InP to the MNO. Resource release costs are assumed to be incorporated within $c_\text{a}\left(i\right)$.

As will be seen in Section IV-F, this cost reduces SFC migrations within a given slice between consecutive time slots.

C. Resource Consumption of Background Services

In a given time slot k, we assume that infrastructure resources are partly consumed by best-effort background services for which no resource reservation has been performed. One denotes

$$\mathbf{B}_k = \left[B_{n,k}\left(i\right), B_{\text{b},k}\left(ij\right)\right]^\top_{(i,ij)\in\mathcal{G}, n\in\Upsilon} \quad (10)$$

the vector gathering all resources consumed by background services during time slot k. The elements $B_{\text{c},k}\left(i\right)$, $B_{\text{m},k}\left(i\right)$, $B_{\text{w},k}\left(i\right)$, $\forall i \in \mathcal{N}$, and $B_{\text{b},k}\left(ij\right)$, $\forall ij \in \mathcal{E}$ of $\mathbf{B}_k$ are random variables representing the aggregate amount of computing, memory, wireless, and bandwidth resources consumed by these best-effort services. As in [22], each of those variables is assumed to be uncorrelated and Gaussian distributed,

$$B_{n,k}\left(i\right) \sim \mathcal{N}\left(\overline{B}_{n,k}\left(i\right), \widetilde{B}^2_{n,k}\left(i\right)\right), \forall i \in \mathcal{N}, \forall n \in \Upsilon, \text{ and}$$
$$B_{\text{b},k}\left(ij\right) \sim \mathcal{N}\left(\overline{B}_{\text{b},k}\left(ij\right), \widetilde{B}^2_{\text{b},k}\left(ij\right)\right), \forall ij \in \mathcal{E}. \quad (11)$$

Consequently, during each time slot k, $\mathbf{B}_k$ is distributed according to $f\left(\mathbf{x}; \boldsymbol{\mu}_{\text{B},k}, \mathbf{\Gamma}_{\text{B},k}\right)$, with

$$\boldsymbol{\mu}_{\text{B},k} = \left[\overline{B}_{n,k}\left(i\right), \overline{B}_{\text{b},k}\left(ij\right)\right]^\top_{(i,ij)\in\mathcal{G}, n\in\Upsilon}, \quad (12)$$
$$\mathbf{\Gamma}_{\text{B},k} = \text{diag}\left[\widetilde{B}^2_{n,k}\left(i\right), \widetilde{B}^2_{\text{b},k}\left(ij\right)\right]_{(i,ij)\in\mathcal{G}, n\in\Upsilon}. \quad (13)$$

The evolution of resources consumed by background services over time slots may be predicted by the InP from past observations, see, *e.g.*, [36]. The smaller variations within each time slot are taken into account in the probability distribution.

IV. Slice Resource Reservation Approaches

Taking the InP perspective, slice resource reservation aims at booking, somewhat in advance, enough infrastructure resources to ensure that the MNOs will be able to provide slices with characteristics as stated in the SM-SLA. For that purpose, the InP has to identify *i*) the infrastructure nodes which will provide resources for the future deployment of VNFs and *ii*) the links able to transmit data between these nodes/VNFs, while respecting the structure of SFCs. This correspond to the network environment preparation block represented in Figure 1. Within some time slot over which a slice s is active, the slice resource reservation can be represented by a mapping between the infrastructure graph $\mathcal{G}$ and the S-RD graph $\mathcal{G}_s$ as illustrated in Figure 4. In this example, the slice s consists of several linear SFCs of the same type for which resources have been reserved from some infrastructure network.

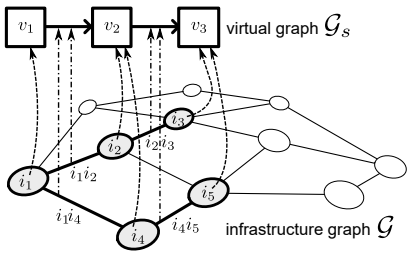


Fig. 4. Reservation of infrastructure resources for slice s: Resources from the infrastructure node i_1 and the aggregate resources from the infrastructure node pairs (i_2, i_4) and (i_3, i_5) are respectively reserved for the virtual nodes v_1, v_2, and v_3. Correspondingly, resources of the infrastructure link pairs (i_1i_2, i_1i_4) and (i_2i_3, i_4i_5) (highlighted by the bold lines) are reserved for the virtual links v_1v_2 and v_2v_3.

The mapping between $\mathcal{G}$ and $\mathcal{G}_s$ may evolve between successive time intervals due to the evolution of the characteristics of the MI-SLA for slice s, to the arrival of new slice resource reservation requests, and to resources released by terminated slices.

A. Prioritized Processing of Resource Reservation Requests

Several strategies for processing resource reservation requests can be considered to account for their dynamicity. A first approach consists in processing the resource reservation requests as soon as they are submitted by an MNO. The advantage is to immediately indicate to the MNO whether enough infrastructure resources are available to satisfy the request. A second approach is to wait some time and process several requests simultaneously. This second approach, considered in this paper, helps process the resource reservation requests since the InP has a better view of concurrent requests. Additionally, by a slice-priority differentiated processing delay of the slice reservation request a compromise between response delay to the MNO and optimization of the reservation of resources by the InP can be found.

When processing a new request, already granted resources requests may be adjusted. This update possibility gives more degrees of freedom to the InP to satisfy new requests, but comes at the price of higher computational complexity. Updates must be done while satisfying previous requests which have been indicated to the MNOs as granted. In this paper, we have chosen not to change any assignment of previously successfully processed slice requests.

Independently of the chosen strategy, the InP has to account for the time required for processing the resource reservation and performing the slice deployment and activation (lasting few minutes, as indicated in [37]). Consequently, resource reservation requests for slices to be activated at $(k+1)\,T$ should reach the InP before $(k+1)\,T - \varepsilon T$, where εT, $\varepsilon \in]0,1[$, is an upper bound of the time required for the slice resource request processing operations, the slice activation and updates.

Let $\mathcal{S}_k$ be the set of slices whose resource reservation requests have been received before $(k+1)\,T - \varepsilon T$. A flag $f_s \in \{0,1\}$ indicates for each slice $s \in \mathcal{S}_k$ whether the request has been processed ($f_s = 1$) (granted or denied) or is still to be processed ($f_s = 0$).

In what follows, we consider two classes of slices, namely Premium and Standard. The priority level is indicated by the MNO to the InP in the MI-SLA of the slice. Each slice request, when received for the first time in the interval $\mathcal{T}_k = [kT - \varepsilon T, (k+1)\,T - \varepsilon T[$, gets $f_s = 0$, and is assigned a priority level $P_{s,k} \in \mathbb{R}$ depending on its class

$$P_{s,k} = \begin{cases} P_{\max} & \text{for Premium slices,} \\ 0 & \text{for Standard slice, if } k_s^{\text{on}} > k+1, \\ P_{\max} - 1 & \text{for Standard slice, if } k_s^{\text{on}} = k+1. \end{cases} \quad (14)$$

Standard slice requests, which have to be activated in the next time slot, get thus a higher priority level. Then, only slices whose priority level is above a certain threshold

$$P_{\text{thres}} = \alpha\,(P_{\max} - 1)\,, \text{ with } \alpha \in [0,1]\,, \quad (15)$$

are processed in the time interval $\mathcal{P}_k = [(k+1)\,T - \varepsilon T, (k+1)\,T[$ of duration εT. The set of slices whose resource request has to be processed during the time interval $\mathcal{P}_k$ is

$$\mathcal{R}_k \triangleq \{s \in \mathcal{S}_k : f_s = 0, P_{s,k} \geqslant P_{\text{thres}}\}\,. \quad (16)$$

Once the resource request of a slice in $\mathcal{R}_k$ is processed, its flag is set to $f_s = 1$. All standard slice requests with $P_{s,k} < P_{\text{thres}}$ (pending requests) are delayed and may be processed in the next time interval $\mathcal{P}_{k+1}$. Their priority is updated as

$$P_{s,k+1} = \begin{cases} \min\{P_{s,k} + \Delta P, P_{\max} - 1\} & \text{if } k_s^{\text{on}} > k+2, \\ P_{\max} - 1 & \text{if } k_s^{\text{on}} = k+2, \end{cases} \quad (17)$$

where $\Delta P \geqslant 0$ is some priority increment. When several slices of equal priority have to be processed in a given time slot, a possible choice, adopted in this paper, is to process first those who have to be activated first, then those who have been submitted first. Premium slices are always processed first. The processing delay of Standard slice requests depends thus on α and ΔP. Deferring more the processing of Standard slice requests gives more chance to satisfy Premium slice requests.

When $\alpha = 0$, whatever the value of ΔP, all slices resource reservation requests received in the time interval $\mathcal{T}_k$ are processed, starting from the Premium slices, with the risk of having no resources available for Premium slice requests received in the few next time slots. This corresponds to the as-they-arrive processing approach, considered, *e.g.*, in [14].

When $\alpha = 1$ and $\Delta P = 0$, the processing of Standard slice resource reservation requests is delayed until the time slot preceding their activation, leaving a maximum amount of resources available for Premium slice resource reservation requests. Standard slice requests are always processed *just-in-time*, while processing of Premium requests is anticipated.

Figure 5 illustrates a scenario taking place during the processing time interval $\mathcal{P}_k$ when the processing of Standard slice requests is maximally delayed ($\alpha = 1$ and $\Delta P = 0$). The three slice requests s_1, s_2, and s_3 in $\mathcal{S}_k$ are assumed still to be processed. The slice request s_4 arrives within $\mathcal{P}_k$ and will thus be considered in $\mathcal{P}_{k+1}$. Among the slices s_1, s_2, and s_3, only s_3 is Premium (the time instant at which the resource reservation request is submitted is indicated by a solid arrow), and is therefore processed in $\mathcal{P}_k$. The slice requests s_1 and s_2 are Standard (reservation request time instants indicated by dashed arrows). They have to be active in the time slots $\mathcal{K}_{s_1} = \left[k_{s_1}^{\text{on}}, k_{s_1}^{\text{off}}\right]$ and $\mathcal{K}_{s_2} = \left[k_{s_2}^{\text{on}}, k_{s_2}^{\text{off}}\right]$. Since $k_{s_1}^{\text{on}} = k+1$ and $k_{s_2}^{\text{on}} = k+2$, only s_1 is processed in $\mathcal{P}_k$. Finally, the set of slice resource reservation requests to be processed in $\mathcal{P}_k$ is $\mathcal{R}_k = \{s_1, s_3\}$ (highlighted by red arrows).

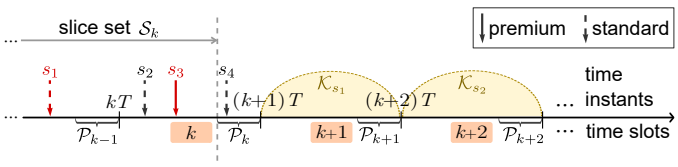


Fig. 5. Time slots, arrival times of the slice resource reservation requests, and time intervals during which the reservation is processed.

B. Decision Variables

Processing a resource reservation request for some slice $s \in \mathcal{R}_k$ amounts to defining a mapping $\boldsymbol{\kappa}_{s,\ell}$ between the graphs $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and $\mathcal{G}_s = (\mathcal{N}_s, \mathcal{E}_s)$ for each time slot $\ell \in \mathcal{K}_s \triangleq$

$\left[k_s^{\text{on}}, k_s^{\text{off}}\right]$ during which the slice s is active. This mapping describes *i*) the number $\kappa_{s,\ell}(i,v) \in \mathbb{N}$ of VNF instances of type $v \in \mathcal{N}_s$ for which node $i \in \mathcal{N}$ will reserve resources, and *ii*) the number $\kappa_{s,\ell}(ij,vw) \in \mathbb{N}$ of links $vw \in \mathcal{E}_s$ between VNF instances for which the InP will reserve resources on the infrastructure link $ij \in \mathcal{E}$, both in time slot ℓ. The amount of resource of type $n \in \Upsilon$ reserved by node i for a VNF instance of type v is $\kappa_{s,\ell}(i,v)\, r_{s,n}(v)$. The bandwidth reserved on link ij to support the traffic between two virtual nodes of type v and w is represented $\kappa_{s,\ell}(ij,vw)\, r_{s,\text{b}}(vw)$.

The mapping $\boldsymbol{\kappa}_{s,\ell}$ is thus defined as

$$\boldsymbol{\kappa}_{s,\ell} = \left\{ \begin{matrix} \kappa_{s,\ell}(i,v), \\ \kappa_{s,\ell}(ij,vw) \end{matrix} \right\}_{(i,ij)\in\mathcal{G},(v,vw)\in\mathcal{G}_s} \tag{18}$$

for each $\ell \in \mathcal{K}_s$. By convention, $\boldsymbol{\kappa}_{s,\ell} = \mathbf{0}$ when $\ell \notin \mathcal{K}_s$. Moreover, one introduces $\boldsymbol{\kappa}_s = \{\boldsymbol{\kappa}_{s,\ell} : \ell \in \mathcal{K}_s\}$ to gather all assignments performed for the slice s.

Enough infrastructure resources are not always available for a given slice $s \in \mathcal{R}_k$. The binary decision variable d_s indicates whether all conditions are met to satisfy the resource reservation request for slice s and consequently whether resources are actually reserved for slice s ($d_s = 1$) or not ($d_s = 0$). These conditions are detailed in the following sections.

Consequently, the set of variables which have to be assigned by the InP in the processing time interval $\mathcal{P}_k$ are

$$\mathbf{d}_{\mathcal{R}_k} = \{d_s : s \in \mathcal{R}_k\}, \text{ and} \tag{19}$$

$$\boldsymbol{\kappa}_{\mathcal{R}_k} = \{\boldsymbol{\kappa}_{s,\ell} : s \in \mathcal{R}_k, \ell \in \mathcal{K}_s\}. \tag{20}$$

The vector $\mathbf{d}_{\mathcal{R}_k}$ indicates which slice resource reservation requests in $\mathcal{R}_k$ have been granted, and $\boldsymbol{\kappa}_k$ describes the way resources have been reserved by the InP.

C. Constraints

During the processing time interval $\mathcal{P}_k$ of time slot k, the InP has to account for all resource reservation requests of slices $s \in \mathcal{S}_{k\text{-}1}$ which have been previously processed, *i.e.*, with $f_s = 1$. The set of these slices is denoted as

$$\mathcal{S}_{k\text{-}1}^{\text{p}} = \{s \in \mathcal{S}_{k\text{-}1} : f_s = 1\}. \tag{21}$$

Moreover, the mappings $\boldsymbol{\kappa}_{s,\ell}$ for all slices $s \in \mathcal{R}_k$ have to satisfy some constraints to ensure that *i*) enough resources are reserved to properly deploy the SFCs and *ii*) the SSP $\underline{p}_s^{\text{sp}}$ is reached. These constraints have to be satisfied for all time slots during which the slice is active. The InP has also to keep the impact probability on background services below $\overline{p}^{\text{im}}$. These constraints are described in what follows.

The total amount of resources reserved (and allocated for slices in service) by each infrastructure node $i \in \mathcal{N}$ and each infrastructure link $ij \in \mathcal{E}$ for all slices $s \in \mathcal{R}_k$ has to be less than their available resources, see Section III-A. Consequently, the following constraints have to be satisfied, for each $\ell = \min_{s\in\mathcal{R}_k}\{k_s^{\text{on}}\} \geqslant k, \ldots, \max_{s\in\mathcal{R}_k}\{k_s^{\text{off}}\}$,

$$\sum_{s\in\mathcal{R}_k}\sum_{v\in\mathcal{N}_s} \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \leqslant a_n(i) - \sum_{s\in\mathcal{S}_{k\text{-}1}^{\text{p}}}\sum_{v\in\mathcal{N}_s} \kappa_{s,\ell}(i,v)\, r_{s,n}(v), \forall i \in \mathcal{N}, n \in \Upsilon, \tag{22}$$

$$\sum_{s\in\mathcal{R}_k}\sum_{vw\in\mathcal{E}_s} \kappa_{s,\ell}(ij,vw)\, r_{s,\text{b}}(vw) \leqslant a_{\text{b}}(ij) - \sum_{s\in\mathcal{S}_{k\text{-}1}^{\text{p}}}\sum_{vw\in\mathcal{E}_s} \kappa_{s,\ell}(ij,vw)\, r_{s,\text{b}}(vw), \forall ij \in \mathcal{E}. \tag{23}$$

In (22) and (23), the right-hand sides of the inequalities represent the remaining part of the resources once previous resource reservation requests have been processed. When updates for granted slice requests are allowed, $\boldsymbol{\kappa}_{s,\ell}$, $s \in \mathcal{S}_{k\text{-}1}^{\text{p}}$ are considered as variables, but not d_s, $s \in \mathcal{S}_{k\text{-}1}^{\text{p}}$, since the status of successfully processed resource reservation requests should not be changed. In what follows, one considers that such updates are not allowed.

The inequalities (22) and (23) may be more compactly written for $\ell > k$ as follows

$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}_{k\text{-}1}^{\text{p}}}\sum_{v\in\mathcal{N}_s} \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \leqslant a_n(i), \tag{24}$$

$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}_{k\text{-}1}^{\text{p}}}\sum_{vw\in\mathcal{E}_s} \kappa_{s,\ell}(ij,vw)\, r_{s,\text{b}}(vw) \leqslant a_{\text{b}}(ij). \tag{25}$$

The conditions (22) and (24) are equivalent, as $\boldsymbol{\kappa}_{s,\ell} = \mathbf{0}$ when $\ell \notin \mathcal{K}_s$. The same is also true for conditions (23) and (25).

For each virtual link $vw \in \mathcal{E}_s$, resources on a sequence of infrastructure links must be reserved between *each* pair of infrastructure nodes that have reserved resources to the virtual nodes v and w. This leads to the following flow conservation constraint, for each $\ell \in \mathcal{K}_s$, $s \in \mathcal{R}_k$, $i \in \mathcal{N}$, and $vw \in \mathcal{E}_s$,

$$\sum_{j\in\mathcal{N}} \left[\kappa_{s,\ell}(ij,vw) - \kappa_{s,\ell}(ji,vw)\right] = \left(\frac{r_{s,\text{b}}(vw)}{\sum_{vu\in\mathcal{E}_s} r_{s,\text{b}}(vu)}\right)\kappa_{s,\ell}(i,v) - \left(\frac{r_{s,\text{b}}(vw)}{\sum_{uw\in\mathcal{E}_s} r_{s,\text{b}}(uw)}\right)\kappa_{s,\ell}(i,w), \tag{26}$$

which is similar to that introduced in [34]. More specifically, (26) imposes that the reserved bandwidth is commensurate with the reserved node resources. This allows the MNO to find an appropriate VNF embedding and chaining solution. Finally, when the SFC embedding is performed, each SFC node will be mapped onto one single node, and each SFC link will be mapped onto one single path. SFCs are not allowed to be mapped towards multiple physical paths.

D. Demand Satisfaction and Impact Probabilities

An assignment $\boldsymbol{\kappa}_{s,\ell}$, $\ell \in \mathcal{K}_s$ of a given slice $s \in \mathcal{R}_k$, which satisfies (22)–(26), has to ensure a SSP above $\underline{p}_s^{\text{sp}}$ for all time slots $\ell \in \mathcal{K}_s$ during which slice s is active. As introduced in Section III, the SSP for a slice is the probability that resources reserved for the deployment of the SFCs of that slice will meet the demand of users of the service. This leads to the following conditions, for all $\ell \in \mathcal{K}_s$ and $s \in \mathcal{R}_k$,

$$p_{s,\ell}(\boldsymbol{\kappa}_{s,\ell}, d_s) \geqslant \underline{p}_s^{\text{sp}}, \tag{27}$$

where

$$p_{s,\ell}\left(\boldsymbol{\kappa}_{s,\ell}, d_s\right) = \Pr\Big\{ \sum_{i\in\mathcal{N}} \kappa_{s,\ell}\left(i,v\right) r_{s,n}\left(v\right) \geqslant d_s R_{s,n,\ell}\left(v\right), \forall v \in \mathcal{N}_s, n \in \Upsilon, \sum_{ij\in\mathcal{E}} \kappa_{s,\ell}\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right) \geqslant d_s R_{s,\mathrm{b},\ell}\left(vw\right), \forall vw \in \mathcal{E}_s \Big\}. \tag{28}$$

The variable d_s in (28) is introduced to cancel the SSP constraint when the request for a slice $s \in \mathcal{R}_k$ is not granted ($d_s = 0$). In this case, as described later in Section IV-F, the minimization of the cost function yields $\boldsymbol{\kappa}_{s,\ell} = \mathbf{0}$, $\forall \ell \in \mathcal{K}_s$, which satisfies (28), thus avoiding the reservation of any resource. The evaluation of (28) is detailed in Section C.

The constraints (22) and (23) ensure that the resources reserved for the slices $s \in \mathcal{R}_k$ are less than the available resources in each infrastructure node $i \in \mathcal{N}$ and infrastructure link $ij \in \mathcal{E}$. Nevertheless, for some $s \in \mathcal{R}_k$, the assignments $\boldsymbol{\kappa}_{s,\ell}$, $\ell \in \mathcal{K}_s$ evaluated in the processing time interval $\mathcal{P}_k$, taking into account all previously processed reservation requests (described by $\boldsymbol{\kappa}_{\mathcal{S}_{k-1}} = \{\boldsymbol{\kappa}_{s,\ell} : s \in \mathcal{S}_{k-1}, \ell \in \mathcal{K}_s\}$), may be such that in time slot $\ell \geqslant k_s^{\mathrm{on}}$, not enough resources are left for the background best-effort services described in Section III-C, and may then significantly affect such services. Consequently, in the processing time interval $\mathcal{P}_k$, when evaluating $\boldsymbol{\kappa}_{\mathcal{R}_k}$, one should have, for all $\ell > k$,

$$p^{\mathrm{im}}_{n,\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, i\right) \leqslant \overline{p}^{\mathrm{im}}, \forall n \in \Upsilon, \forall i \in \mathcal{N}, \tag{29}$$

$$p^{\mathrm{im}}_{\mathrm{b},\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, ij\right) \leqslant \overline{p}^{\mathrm{im}}, \forall ij \in \mathcal{E}, \tag{30}$$

where

$$p^{\mathrm{im}}_{n,\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, i\right) = \Pr\Big\{ \sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k-1}} \sum_{v\in\mathcal{N}_s} \kappa_{s,\ell}\left(i,v\right) r_{s,n}\left(v\right) \geqslant a_n\left(i\right) - B_{n,\ell}\left(i\right) \Big\}, \tag{31}$$

$$p^{\mathrm{im}}_{\mathrm{b},\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, ij\right) = \Pr\Big\{ \sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k-1}} \sum_{vw\in\mathcal{E}_s} \kappa_{s,\ell}\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right) \geqslant a_{\mathrm{b}}\left(ij\right) - B_{\mathrm{b},\ell}\left(ij\right) \Big\}. \tag{32}$$

As indicated before, the evaluations of (31) and (32) for all $\ell > k$ involve $\boldsymbol{\kappa}_{\mathcal{S}^{\mathrm{p}}_{k-1}}$ which have already been evaluated in previous processing time intervals and are considered as constants in the current processing time interval, *i.e.*, $\mathcal{P}_k$. The dependency in $\boldsymbol{\kappa}_{\mathcal{S}^{\mathrm{p}}_{k-1}}$ of $p^{\mathrm{im}}_{n,\ell}$ and $p^{\mathrm{im}}_{\mathrm{b},\ell}$ is omitted to lighten notations.

The constraints (29) ensure that the reserved resources have a limited impact on background services at each node $i \in \mathcal{N}$. The constraints (30) have the same role for the infrastructure links. 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 reserved resources for slices on background services, but makes the resource reservation problem more difficult compared to the case of $\overline{p}^{\mathrm{im}}$ close to one, where the impact on background service is less taken into account. The InP, by adjusting the impact probability threshold $\overline{p}^{\mathrm{im}}$, can trade the revenues provided by slices and those provided by the background services. For example, choosing $\overline{p}^{\mathrm{im}} = 1$ may leave no resources for background services.

E. Relaxation of Probabilistic Constraints

This section introduces the relaxation of the probabilistic constraints (27), (29), and (30). These constraints are nonlinear due to the need to evaluate $p_{s,\ell}\left(\boldsymbol{\kappa}_{s,\ell}, d_s\right)$, $p^{\mathrm{im}}_{n,\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, i\right)$, and $p^{\mathrm{im}}_{\mathrm{b},\ell}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}, ij\right)$ with (28), (31), and (32). Using the approach introduced in [22], the MNO translates the SSP constraint (27), for all $s \in \mathcal{R}_k$ and $\ell \in \mathcal{K}_s$, into the following linear deterministic constraints

$$\sum_{i\in\mathcal{N}} \kappa_{s,\ell}\left(i,v\right) r_{s,n}\left(v\right) \geqslant d_s \widehat{R}_{s,n,\ell}\left(v, \gamma_{s,\ell}\right), \forall v, n, \tag{33}$$

$$\sum_{ij\in\mathcal{E}} \kappa_{s,\ell}\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right) \geqslant d_s \widehat{R}_{s,\mathrm{b},\ell}\left(vw, \gamma_{s,\ell}\right), \forall vw, \tag{34}$$

where

$$\widehat{R}_{s,n,\ell}\left(v, \gamma_{s,\ell}\right) = \overline{R}_{s,n,\ell}\left(v\right) + \gamma_{s,\ell} \widetilde{R}_{s,n,\ell}\left(v\right), \tag{35}$$

$$\widehat{R}_{s,\mathrm{b},\ell}\left(vw, \gamma_{s,\ell}\right) = \overline{R}_{s,\mathrm{b},\ell}\left(vw\right) + \gamma_{s,\ell} \widetilde{R}_{s,\mathrm{b},\ell}\left(vw\right), \tag{36}$$

are the target aggregate user demands, depending on some parameter $\gamma_{s,\ell} > 0$. $\overline{R}_{s,n,\ell}\left(v\right)$ and $\widetilde{R}_{s,n,\ell}\left(v\right)$ are the mean and standard deviation of $R_{s,n,\ell}\left(v\right)$, while $\overline{R}_{s,\mathrm{b},\ell}\left(vw\right)$ and $\widetilde{R}_{s,\mathrm{b},\ell}\left(vw\right)$ are the mean and standard deviation of $R_{s,\mathrm{b},\ell}\left(vw\right)$. These quantities are evaluated by the MNO. Appendix A details this evaluation when when the number of users $N_{s,\ell}$ of slice s at time slot ℓ is described by a binomial distribution. Appendix B describes the choice of $\gamma_{s,\ell}$ such that the satisfaction of (33, 34) implies that of (27). This way, the MNO can control the bound of the probability of SLA non-satisfaction. The quantities $\widehat{R}_{s,n,\ell}$ and $\widehat{R}_{s,\mathrm{b},\ell}$ are transmitted by the MNO to the InP as part of the MI-SLA.

Similarly, the InP translates the IP constraints (29, 30), $\forall\left(i, ij\right) \in \mathcal{G}$ and $\forall n \in \Upsilon$, into

$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k-1}, v\in\mathcal{N}_s} \kappa_{s,\ell}\left(i,v\right) r_{s,n}\left(v\right) \leqslant a_n\left(i\right) - \widehat{B}_{n,\ell}\left(i, \gamma_{\mathrm{B},\ell}\right), \tag{37}$$

$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k-1}, vw\in\mathcal{E}_s} \kappa_{s,\ell}\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right) \leqslant a_{\mathrm{b}}\left(ij\right) - \widehat{B}_{\mathrm{b},\ell}\left(ij, \gamma_{\mathrm{B},\ell}\right), \tag{38}$$

for $\ell > k$, where

$$\widehat{B}_{n,\ell}\left(i, \gamma_{\mathrm{B},\ell}\right) = \overline{B}_{n,\ell}\left(i\right) + \gamma_{\mathrm{B},\ell} \widetilde{B}_{n,\ell}\left(i\right), \tag{39}$$

$$\widehat{B}_{\mathrm{b},\ell}\left(ij, \gamma_{\mathrm{B},\ell}\right) = \overline{B}_{\mathrm{b},\ell}\left(ij\right) + \gamma_{\mathrm{B},\ell} \widetilde{B}_{\mathrm{b},\ell}\left(ij\right) \tag{40}$$

are the considered target level of background service demands. The parameter $\gamma_{\mathrm{B},\ell} > 0$ has to be chosen such that the satisfaction of the constraints (37, 38) implies the satisfaction of the IP constraints (29, 30), see Appendix C for more details. Moreover, if the constraints (37, 38) are satisfied by some assignment $\boldsymbol{\kappa}_{\mathcal{R}_k}$, then the conditions (24) and (25) are also satisfied.

F. Costs and Incomes

Consider the processing time interval $\mathcal{P}_k$ during which a resource reservation scheme for all slices $s \in \mathcal{R}_k$ has to be evaluated. This amounts at evaluating $\mathbf{d}_{\mathcal{R}_k}$ and the assignments $\boldsymbol{\kappa}_{\mathcal{R}_k}$.

The costs charged by the InP to the MNO for a reservation scheme for slice $s \in \mathcal{R}_k$ described by $\boldsymbol{\kappa}_{s,\ell}$ in time slot ℓ are spread between node and bandwidth *resource* reservation costs

$$C_{\mathrm{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right) = \sum_{i\in\mathcal{N}}\sum_{v\in\mathcal{N}_s}\sum_{n\in\Upsilon} \kappa_{s,\ell}\left(i,v\right) r_n\left(v\right) c_n\left(i\right) + \sum_{ij\in\mathcal{E}}\sum_{vw\in\mathcal{E}_s} \kappa_{s,\ell}\left(ij,vw\right) r_{\mathrm{b}}\left(vw\right) c_{\mathrm{b}}\left(ij\right) \quad (41)$$

as well as *fixed* node disposal costs

$$C_{\mathrm{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right) = \sum_{i\in\mathcal{N}} \widetilde{\kappa}_{s,\ell}\left(i\right) c_{\mathrm{f}}\left(i\right) \quad (42)$$

for the infrastructure nodes used, where

$$\widetilde{\kappa}_{s,\ell}\left(i\right) = \begin{cases} 1 & \text{if } \sum_{v\in\mathcal{N}_s} \kappa_{s,\ell}\left(i,v\right) > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (43)$$

indicates whether node i is used by slice s in time slot ℓ.

Additionally, when the amount of reserved resources for slice s increases during two consecutive time slots, resource variation adaptation costs are also charged by the InP to the MNO

$$C_{\mathrm{a}}\left(\boldsymbol{\kappa}_{s,\ell},\boldsymbol{\kappa}_{s,\ell-1}\right) = \sum_{i\in\mathcal{N}}\sum_{v\in\mathcal{N}_s} \max\left\{\kappa_{s,\ell}\left(i,v\right) - \kappa_{s,\ell\text{-}1}\left(i,v\right), 0\right\} c_{\mathrm{a}}\left(i\right), \quad (44)$$

see Section III-B3.

Once a reservation request for a slice $s \in \mathcal{R}_k$ has been granted by the InP, the MNO will be able to deploy the slice (see the *commissioning* and *operation* blocks of Figure 1) and receives from the SP some income I_s depending on the complexity and of the load of the slice.

G. Optimization Problem

For a given assignment $\boldsymbol{\kappa}_{\mathcal{R}_k}$, the earnings of the InP are the costs charged to the MNOs

$$E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}\right) = \sum_{s\in\mathcal{R}_k}\sum_{\ell\in\mathcal{K}_s} \left(C_{\mathrm{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right) + C_{\mathrm{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right) + C_{\mathrm{a}}\left(\boldsymbol{\kappa}_{s,\ell},\boldsymbol{\kappa}_{s,\ell-1}\right)\right). \quad (45)$$

The InP may be interested in an assignment $\boldsymbol{\kappa}_{\mathcal{R}_k}$ that maximizes $E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}\right)$. Nevertheless, such assignment would be detrimental for the earnings of the MNOs expressed as

$$E_k^{\mathrm{MNO}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}\right) = \sum_{s\in\mathcal{R}_k} d_s I_s - E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}\right). \quad (46)$$

Consequently, MNOs may not be interested by InPs applying an optimization strategy trying to maximizing $E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}\right)$.

Alternatively, the InP may try to find an assignment which maximizes $E_k^{\mathrm{MNO}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}\right)$. This approach reduces the per-slice income for the InP, but allows more slice resource reservation requests to be granted. Nevertheless, InPs are usually unaware of the income I_s obtained by the MNOs from the SP, therefore $E_k^{\mathrm{MNO}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}\right)$ cannot be evaluated by the InP.

Consequently, we will consider an approach where the InP tries, for a given $\mathbf{d}_{\mathcal{R}_k}$ (which determines the number of accepted slices), to find an assignment $\boldsymbol{\kappa}_{\mathcal{R}_k}$ minimizing the reservation costs charged to the MNO. Moreover, the InP also tries to maximize, with respect to $\mathbf{d}_{\mathcal{R}_k}$, its earning $\min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}\right)$. This approach leads to a max-min optimization problem. Its solution provides the InP maximum earnings for granting slice requests and is also appropriate for the MNO in terms of charged reservation costs. Moreover, this approach potentially saves infrastructure resources to satisfy future slice resource reservation requests.

Consequently, the *joint* reservation of resources for all slices $s \in \mathcal{R}_k$ during the processing time interval $\mathcal{P}_k$ is formulated as Problem 1.

Problem 1: Max-Min Joint Slice Resource Reservation

$$\max_{\mathbf{d}_{\mathcal{R}_k}} \min_{\boldsymbol{\kappa}_{\mathcal{R}_k}} E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k}\right)$$

s.t. $\forall \ell > k$:

$$\sum_{i\in\mathcal{N}} \kappa_{s,\ell}\left(i,v\right) r_{s,n}\left(v\right) \geqslant d_s \widehat{R}_{s,n,\ell}\left(v,\gamma_{s,\ell}\right), \forall n \in \Upsilon, v \in \mathcal{N}_s,$$

$$\sum_{ij\in\mathcal{E}} \kappa_{s,\ell}\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right) \geqslant d_s \widehat{R}_{s,\mathrm{b},\ell}\left(vw,\gamma_{s,\ell}\right), \forall vw \in \mathcal{E}_s,$$

and s.t. $\forall \ell > k, \forall i \in \mathcal{N}$:

$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}_{k-1}^{\mathrm{p}}, v\in\mathcal{N}_s} \kappa_{s,\ell}\left(i,v\right) r_{s,n}\left(v\right) \leqslant a_n\left(i\right) - \widehat{B}_{n,\ell}\left(i,\gamma_{\mathrm{B},\ell}\right),$$

and s.t. $\forall \ell > k, \forall ij \in \mathcal{E}$:

$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}_{k-1}^{\mathrm{p}}, vw\in\mathcal{E}_s} \kappa_{s,\ell}\left(ij,vw\right) r_{s,\mathrm{b}}\left(vw\right) \leqslant a_{\mathrm{b}}\left(ij\right) - \widehat{B}_{\mathrm{b},\ell}\left(ij,\gamma_{\mathrm{B},\ell}\right),$$

$$\sum_{j\in\mathcal{N}} \left[\kappa_{s,\ell}\left(ij,vw\right) - \kappa_{s,\ell}\left(ji,vw\right)\right] = \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{vu\in\mathcal{E}_s} r_{s,\mathrm{b}}(vu)}\right)\kappa_{s,\ell}\left(i,v\right) - \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{uw\in\mathcal{E}_s} r_{s,\mathrm{b}}(uw)}\right)\kappa_{s,\ell}\left(i,w\right),$$

$\forall s \in \mathcal{R}_k, i \in \mathcal{N}, \forall vw \in \mathcal{E}_s$.

The max-min optimization makes Problem 1 difficult to solve. This motivates us to develop some efficient heuristics in Section V.

V. Slice Resource Reservation Algorithms

In this section, two heuristics are introduced to provide approximate solutions to Problem 1, performing either joint or sequential slice resource reservation.

A. Linearization of the Cost Function

In (45), the term $C_{\mathrm{a}}\left(\boldsymbol{\kappa}_{s,\ell},\boldsymbol{\kappa}_{s,\ell-1}\right)$ makes the objective function nonlinear. To address this issue, consider the set of variables

$$\mathbf{y}_s = \left\{y_{s,\ell}\left(i,v\right) : \ell \in \mathcal{K}_s, i \in \mathcal{N}, v \in \mathcal{N}_s\right\} \quad (47)$$

for each $s \in \mathcal{R}_k$, $\mathbf{y}_{\mathcal{R}_k} = \{\mathbf{y}_s : s \in \mathcal{R}_k\}$, and reformulate the objective function (45) as

$$E_k^{\mathrm{InP}}\left(\boldsymbol{\kappa}_{\mathcal{R}_k},\mathbf{y}_{\mathcal{R}_k}\right) = \sum_{s\in\mathcal{R}_k}\sum_{\ell\in\mathcal{K}_s} \Big(C_{\mathrm{r}}\left(\boldsymbol{\kappa}_{s,\ell}\right) + C_{\mathrm{f}}\left(\boldsymbol{\kappa}_{s,\ell}\right) + \sum_{i\in\mathcal{N}}\sum_{v\in\mathcal{N}_s} y_{s,\ell}\left(i,v\right) c_{\mathrm{a}}\left(i\right)\Big) \quad (48)$$

with the additional constraints, to be satisfied for all $s \in \mathcal{R}_k$, $\ell \in \mathcal{K}_s$, $i \in \mathcal{N}$, and $v \in \mathcal{N}_s$

$$y_{s,\ell}(i,v) \geqslant \kappa_{s,\ell}(i,v) - \kappa_{s,\ell\text{-}1}(i,v), \tag{49}$$

$$y_{s,\ell}(i,v) \geqslant 0. \tag{50}$$

For a given value of $\mathbf{d}_{\mathcal{R}_k}$, the objective function has now to be minimized with respect to $\boldsymbol{\kappa}_{s,\ell}$, $s \in \mathcal{R}_k$, $\ell \in \mathcal{K}_s$, and $\mathbf{y}_{\mathcal{R}_k}$.

Moreover, the evaluation of $C_{\mathrm{f}}(\boldsymbol{\kappa}_{s,\ell})$ involves $\widetilde{\kappa}_{s,\ell}(i)$ defined in (43). The variable $\widetilde{\kappa}_{s,\ell}(i)$ can be related to $\sum_v \kappa_{s,\ell}(i,v)$ using the following linear inequality constraints

$$\sum_v \kappa_{s,\ell}(i,v) \geqslant 0, \tag{51}$$

$$\widetilde{\kappa}_{s,\ell}(i)\,\overline{N} \geqslant \sum_v \kappa_{s,\ell}(i,v), \tag{52}$$

where $\overline{N}$ is an upper bound on the number of VNF instances of all types for which resources may be reserved by a given infrastructure node.

B. Relaxed Joint Max-Min Optimization Problem

Even with the results of Section V-A, the solution of Problem 1 requires addressing a constrained max-min optimization problem, which is still quite complex. To address this issue, for a fixed value of $\mathbf{d}_{\mathcal{R}_k}$, we introduce the following optimization problem.

Problem 2: Joint Slice Resource Reservation Given $\mathbf{d}_{\mathcal{R}_k}$

$$\min_{\boldsymbol{\kappa}_{\mathcal{R}_k},\mathbf{y}_{\mathcal{R}_k}} E_k^{\mathrm{InP}}(\boldsymbol{\kappa}_{\mathcal{R}_k},\mathbf{y}_{\mathcal{R}_k})$$

s.t. $\forall s \in \mathcal{R}_k, \forall \ell > k$:

$$\sum_{i\in\mathcal{N}} \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \geqslant d_s \widehat{R}_{s,n,\ell}(v,\gamma_{s,\ell}), \forall n \in \Upsilon, v \in \mathcal{N}_s,$$

$$\sum_{ij\in\mathcal{E}} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant d_s \widehat{R}_{s,\mathrm{b},\ell}(vw,\gamma_{s,\ell}), \forall vw \in \mathcal{E}_s,$$

$$y_{s,\ell}(i,v) \geqslant \kappa_{s,\ell}(i,v) - \kappa_{s,\ell\text{-}1}(i,v), \forall i \in \mathcal{N}, v \in \mathcal{N}_s,$$

$$y_{s,\ell}(i,v) \geqslant 0, \forall i \in \mathcal{N}, v \in \mathcal{N}_s,$$

and s.t. $\forall \ell > k, \forall i \in \mathcal{N}$:

$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k\text{-}1}, v\in\mathcal{N}_s} \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \leqslant a_n(i) - \widehat{B}_{n,\ell}(i,\gamma_{\mathrm{B},\ell}),$$

and s.t. $\forall \ell > k, \forall ij \in \mathcal{E}$:

$$\sum_{s\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k\text{-}1}, vw\in\mathcal{E}_s} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij) - \widehat{B}_{\mathrm{b},\ell}(ij,\gamma_{\mathrm{B},\ell}),$$

$$\sum_{j\in\mathcal{N}} [\kappa_{s,\ell}(ij,vw) - \kappa_{s,\ell}(ji,vw)] =$$

$$\left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{vu\in\mathcal{E}_s} r_{s,\mathrm{b}}(vu)}\right)\kappa_{s,\ell}(i,v) - \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{uw\in\mathcal{E}_s} r_{s,\mathrm{b}}(uw)}\right)\kappa_{s,\ell}(i,w),$$

$\forall s \in \mathcal{R}_k, i \in \mathcal{N}, \forall vw \in \mathcal{E}_s$.

One considers then a greedy solution approach to Problem 1, where the slices to be processed in $\mathcal{R}_k = \{s_1, \ldots, s_{R_k}\}$ are assumed to be ordered, see Section IV-A. When several MNOs have submitted slice resource reservation requests in the same time slot, the InP may also prioritize MNOs.

C. Relaxed Single Slice Max-Min Optimization Problem

The number of variables involved in Problem 2 introduced in Section V-B may become relatively large when the resource reservation is performed simultaneously for several slices. For this reason, we introduce a reduced-complexity version of Problem 2, where the resource reservation is performed slice by slice. One focuses on a slice $s \in \mathcal{R}_k$ which resource reservation request has to be processed. Some resource reservation requests for slices $s' \in \mathcal{R}_k$, $s' \neq s$ may have been previously processed, in which case, when the request is granted, $d_{s'} = 1$ and $\boldsymbol{\kappa}_{s'} \neq \mathbf{0}$ and when it is not granted, $d_{s'} = 0$ and $\boldsymbol{\kappa}_{s'} = \mathbf{0}$. For not yet processed requests of slices $s' \in \mathcal{R}_k$, $s' \neq s$, one considers $d_{s'} = 0$ and $\boldsymbol{\kappa}_{s'} = \mathbf{0}$. With these assumptions, reserving resources for slice $s \in \mathcal{R}_k$ requires the solution of Problem 3.

Problem 3: Single Slice Resource Reservation

$$\max_{d_s} \min_{\boldsymbol{\kappa}_s,\mathbf{y}_s} E_k^{\mathrm{InP}}(\boldsymbol{\kappa}_s,\mathbf{y}_s)$$

s.t. $\forall \ell \in \mathcal{K}_s$:

$$\sum_{i\in\mathcal{N}} \kappa_{s,\ell}(i,v)\, r_{s,n}(v) \geqslant d_s \widehat{R}_{s,n,\ell}(v,\gamma_{s,\ell}), \forall n \in \Upsilon, v \in \mathcal{N}_s,$$

$$\sum_{ij\in\mathcal{E}} \kappa_{s,\ell}(ij,vw)\, r_{s,\mathrm{b}}(vw) \geqslant d_s \widehat{R}_{s,\mathrm{b},\ell}(vw,\gamma_{s,\ell}), \forall vw \in \mathcal{E}_s,$$

$$y_{s,\ell}(i,v) \geqslant \kappa_{s,\ell}(i,v) - \kappa_{s,\ell\text{-}1}(i,v), \forall i \in \mathcal{N}, v \in \mathcal{N}_s,$$

$$y_{s,\ell}(i,v) \geqslant 0, \forall i \in \mathcal{N}, v \in \mathcal{N}_s,$$

and s.t. $\forall \ell \in \mathcal{K}_s, \forall i \in \mathcal{N}$:

$$\sum_{s'\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k\text{-}1}, v\in\mathcal{N}_s} \kappa_{s',\ell}(i,v)\, r_{s',n}(v) \leqslant a_n(i) - \widehat{B}_{n,\ell}(i,\gamma_{\mathrm{B},\ell}),$$

and s.t. $\forall \ell \in \mathcal{K}_s, \forall ij \in \mathcal{E}$:

$$\sum_{s'\in\mathcal{R}_k\cup\mathcal{S}^{\mathrm{p}}_{k\text{-}1}, vw\in\mathcal{E}_s} \kappa_{s',\ell}(ij,vw)\, r_{s',\mathrm{b}}(vw) \leqslant a_{\mathrm{b}}(ij) - \widehat{B}_{\mathrm{b},\ell}(ij,\gamma_{\mathrm{B},\ell}),$$

$$\sum_{j\in\mathcal{N}} [\kappa_{s,\ell}(ij,vw) - \kappa_{s,\ell}(ji,vw)] =$$

$$\left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{vu\in\mathcal{E}_s} r_{s,\mathrm{b}}(vu)}\right)\kappa_{s,\ell}(i,v) - \left(\frac{r_{s,\mathrm{b}}(vw)}{\sum_{uw\in\mathcal{E}_s} r_{s,\mathrm{b}}(uw)}\right)\kappa_{s,\ell}(i,w),$$

$\forall i \in \mathcal{N}, vw \in \mathcal{E}_s$.

Assuming again that the slice resource reservation requests are ordered, see Section IV-A, one may get a second greedy reservation algorithm where slice resource reservation requests are processed slice by slice solving Problem 3 for each slice. The highest priority slice is processed first. The lower priority slices are then processed, whatever the reservation result of a higher priority slice. Even if high-priority slices may have their resource reservation request rejected, lower-priority slice requests may be granted for slices with smaller resource requirements.

D. Slice Resource Reservation Approaches

For the suboptimal algorithms introduced in Sections V-B and V-C, two Prioritized slice resource Reservation (PR) variants are considered, depending on whether slices resource reservation requests are processed jointly (J-PR) or sequentially (S-PR).

1) Joint Approach: In the J-PR approach, all slices in $\mathcal{R}_k$ are processed jointly. This is done by solving Problem 2, starting by fixing $\mathbf{d}_{\mathcal{R}_k} = (d_1, \ldots, d_{R_k}) = (1, \ldots, 1)$, *i.e.*, we try initially to satisfy all slice resource reservation requests. If the reservation is successful, the algorithm stops. If no solution is returned, the resource reservation request of the slice with lowest priority is not granted, *i.e.*, $d_{R_k} = 0$. Problem 2 is solved again considering $\mathbf{d}_{\mathcal{R}_k} = (1, \ldots, 1, 0)$. If there is still no solution, the resource reservation request for the slice with second lowest priority is not granted, and so forth. If more than two slice requests have the same lowest priority, the last arrived one is not granted. The first part of Algorithm 1 (Lines 4–13) summarizes the J-PR approach.

2) Sequential Approach: In the S-PR approach, slice resource reservation requests in $\mathcal{R}_k$ are processed sequentially. This is done by solving Problem 3, for each slice $s \in \mathcal{R}_k$, $d_s \in \{0, 1\}$, starting from that with highest priority. The maximization of the cost function considers $d_s = 1$. If the following minimization problem admits a solution, one keeps $d_s = 1$. If the minimization problem admits no solution, $d_s = 0$ and $\boldsymbol{\kappa}_s = \mathbf{0}$ is the solution.

The second part of Algorithm 1 (Lines 14–17) summarizes the S-PR approach. Note that, S-PR when $\alpha = 0$ implements a first-arrived first-served processing policy.

Algorithm 1: Prioritized Slice Resource Reservation

```
1  foreach processing time interval P_k do
2      Get sorted slice request set R_k from MNO;
3      switch reservation_variant do
4          case J-PR (joint prioritized reservation) do
5              Initialize d_{R_k} = (1, ..., 1);
6              i = |R_k|;
7              while i > 0 do
8                  Solve Problem 2 to get κ_{R_k};
9                  if no feasible solution found then
10                     d_i = 0;
11                     i = i − 1;
12                 else
13                     break;
14         case S-PR (sequential prioritized reservation) do
15             Initialize d_s = 0, κ_s = 0, ∀s ∈ R_k;
16             foreach s ∈ R_k do
17                 Solve Problem 3 for slice s to get d_s and κ_s;
18     # Update flag of processed slice requests
19     Set f_s = 1, ∀s ∈ R_k;
```

3) Complexity Analysis: Each variant in Algorithm 1 requires the solution of one or several ILPs, whose complexity increases exponentially with the number of variables in the worst case. The J-PR variant considers a single ILP involving $\sum_{s\in\mathcal{R}_k}\sum_{\ell\in\mathcal{K}_s}(|\mathcal{N}| + 2|\mathcal{N}||\mathcal{N}_s| + |\mathcal{E}||\mathcal{E}_s|)$ variables and $\sum_{s\in\mathcal{R}_k}\sum_{\ell\in\mathcal{K}_s}(|\mathcal{N}||\mathcal{E}_s| + |\mathcal{N}_s||\Upsilon| + |\mathcal{E}_s| + 2|\mathcal{N}||\mathcal{N}_s|) + \sum_{\ell\in\mathcal{K}_s}(|\mathcal{N}| + |\mathcal{E}|)$ constraints. The S-PR variant instead splits the task into $|\mathcal{R}_k|$ subproblems, each of which involves $\sum_{\ell\in\mathcal{K}_s}(|\mathcal{N}| + 2|\mathcal{N}||\mathcal{N}_s| + |\mathcal{E}||\mathcal{E}_s|)$ variables and $\sum_{\ell\in\mathcal{K}_s}(|\mathcal{N}||\mathcal{E}_s| + |\mathcal{N}_s||\Upsilon| + |\mathcal{E}_s| + 2|\mathcal{N}||\mathcal{N}_s| + |\mathcal{N}| + |\mathcal{E}|)$ constraints. Therefore, each subproblem in the sequential approach implies $|\mathcal{R}_k|$ times less variables than the joint variant. Consequently, due to the exponential complexity of the NP-hard ILP, the sequential approach may provide a solution faster than the joint variant. In Section VI, the two proposed variants are compared via numerical simulations.

VI. Evaluation

This section presents simulations to evaluate the performance of the two reservation algorithms, J-PR and S-PR, described in Section V. The simulation setup is described in Section VI-A. All simulations described in Section VI-B are performed with the CPLEX MILP solver interfaced with MATLAB. Our work focuses on the slice admission control and resource reservation mechanisms, both taking place before any slice deployment and activation. Consequently, in the simulation, aspects related to user admission control, radio coverage/interference, or packet queuing and propagation delays are not considered.

A. Simulation Conditions

1) Infrastructure Topology: The considered infrastructure network is represented by the binary fat tree topology depicted in Figure 6, taken from [38, 39]. The leaf nodes represent the far-edge hosts of Radio Unit (RU)/Distributed Unit (DU). The other nodes represent the edge, regional, and central data centers. Infrastructure nodes 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. Infrastructure links provide a given bandwidth a_b, expressed in Gbps. The per-unit resource cost ($c_n(i)$ and $c_\mathrm{b}(ij)$), fixed cost $c_\mathrm{f}(i)$, and reservation adaptation cost $c_\mathrm{a}(i)$ are respectively 1, 10, and 20, $\forall(i, ij) \in \mathcal{G}$.

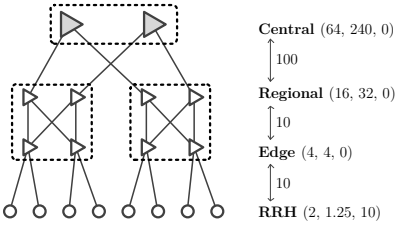


Fig. 6. Description of the binary fat-tree infrastructure network considered in the simulations; Nodes provide a given amount of computing a_c, memory a_m, and wireless a_w resources expressed in number of used CPUs, Gbytes, and Gbps; Links are able to transmit data at a rate a_b expressed in Gbps.

2) Slice Resource Demand (S-RD): The number of users of a slice s is assumed to follow a binomial distribution of parameter $p_{s,k}$, see [40, 41]. Considering a Gaussian distribution for the individual user resource demands, the resulting resource demand for the slice follows a distribution close to a log-normal distribution, as observed in [42]. One considers two patterns to represent the evolution with time of $p_{s,k}$, which

impacts the evolution of the slice resource demands. The first, illustrated in Figure 7a, corresponds to a constant demand $p_{s,k} = 1$ during the whole lifetime of the slice. The second, shown in Figure 7b, describes a slice whose resource demand evolves from one time slot to the next.

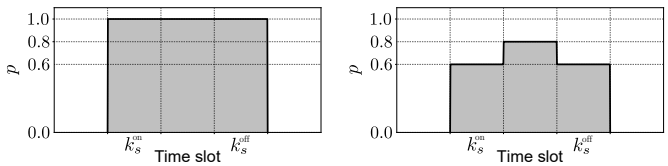


Fig. 7. Probability pattern of service usage: (a) constant over a time interval and (b) piece-wise constant.

Three types of slices are considered.

- Slices of type 1 aim to provide an HD video streaming service at an average rate of 6 Mbps for VIP users, *e.g.*, in a stadium. The number of users of a slice s of type 1 follows a binomial distribution $\mathcal{B}(500, p_{s,k})$. The function architecture of slices of type 1 is composed of 3 VNFs: a virtual Video Optimization Controller (vVOC), a virtual Gateway (vGW), and a virtual Base Band Unit (vBBU). The required SSP for type 1-slices is $\underline{p}_s^{\text{sp}} = 0.99$;
- Slices of type 2 are dedicated to provide an SD video streaming service at an average rate of 4 Mbps. The number of users of a slice s of type 2 follows a binomial distribution $\mathcal{B}(2000, p_{s,k})$. The function architecture of slices of type 2 is similar to that of type 1. The required SSP for type 2-slices is $\underline{p}_s^{\text{sp}} = 0.95$;
- Slices of type 3 aim to provide a video surveillance and traffic monitoring service at an average rate of 2 Mbps for 200 cameras, *e.g.*, installed along a highway. The demand pattern of such type of slice is thus always that of Figure 7a. 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). The required SSP for type 3-slices is $\underline{p}_s^{\text{sp}} = 0.9$.

The slice type is chosen uniformly at random. For slices of type 1 and 2, the demand pattern is also chosen uniformly at random.

A normalized unit duration time slot is considered, *i.e.*, $T = 1$. The processing duration is chosen as $\varepsilon T = 0.1T$. The number of reservation request arrivals in each time slot obeys a Poisson distribution $\text{Pois}(\mu)$ of parameter $\mu = 2$. The arrival time of each slice request is uniformly distributed within each time interval $\mathcal{T}_k$. The activation delay (*i.e.*, $k_s^{\text{on}} - k_s$) follows the uniform distribution $\mathcal{U}(1, 6)$ and the lifetime follows the uniform distribution $\mathcal{U}(1, 3)$.

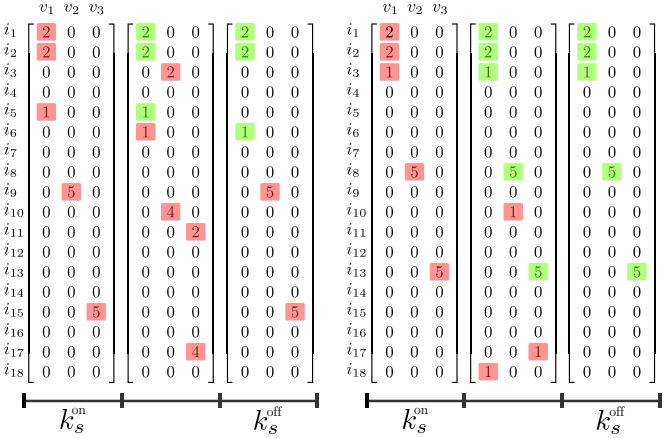
As detailed in Section III-B2, the resource requirements for the various SFCs that will have to be deployed within a slice are aggregated within an S-RD graph that mimics the SFC-RD graph. S-RD nodes and links are characterized by the aggregated resources needed to support the targeted number of users. Details of each resource type as well as the associated U-RD, SFC-RD, and S-RD graph are given in Table II. Numerical values in Table II have been adapted from [43]. Each slice request is randomly assigned one type among these three types.

3) Background Services: At each infrastructure node $i \in \mathcal{N}$ and link $ij \in \mathcal{E}$ and for all time slots k, we assume that the resources consumed by best-effort background services follow a normal distribution with mean and standard deviation equal to respectively 20% and 5% of the available resources at a node and at a link, *i.e.*, $\left\{\overline{B}_{n,k}(i), \widetilde{B}_{n,k}(i)\right\} = \{0.2a_n(i), 0.05a_n(i)\}\ \forall i \in \mathcal{N}, \forall n \in \Upsilon$ and $\left\{\overline{B}_{\text{b},k}(ij), \widetilde{B}_{\text{b},k}(ij)\right\} = \{0.2a_{\text{b}}(ij), 0.05a_{\text{b}}(ij)\}$, $\forall ij \in \mathcal{E}$. The reservation impact probability threshold $\overline{p}^{\text{im}}$ is set to 0.1.

B. Results

The performance of the reservation variants (J-PR and S-PR) is evaluated considering the following metrics: slice request acceptance rate, per-slice reservation cost, average response delay (*i.e.*, time between the time instant the request arrives and the time instant at which it is processed), average number of adjusted VNF instances per slice, and average computing time for each slice resource reservation request.

1) Resource Reservation for a Single Slice: The first simulation aims at illustrating the impact of the adaptation costs described in Section III-B3, on the adjustments of the reserved resources between consecutive time slots. A single slice s is considered. Consequently, the J-PR and S-PR reservation variants yield the same assignment $\boldsymbol{\kappa}_{s,\ell}$.



(a) $\kappa_{s,\ell}(i,v)$ when $c_{\text{a}} = 0$. (b) $\kappa_{s,\ell}(i,v)$ when $c_{\text{a}} > 0$.

Fig. 8. Evolution of the assignment $\kappa_{s,\ell}(i,v)$ for a single slice (for each matrix, rows correspond to i, columns to v) when (a) $c_{\text{a}} = 0$ and (b) $c_{\text{a}} > 0$; the matrix entries with $\kappa_{s,\ell}(i,v) - \kappa_{s,\ell\text{-}1}(i,v) > 0$ are highlighted in red, whereas entries with $\kappa_{s,\ell}(i,v) - \kappa_{s,\ell\text{-}1}(i,v) \leqslant 0$ and $\kappa_{s,\ell}(i,v) > 0$ are in green.

Figure 8b illustrates the evolution with the time index ℓ of $\boldsymbol{\kappa}_{s,\ell}$ for a slice s of type 1, characterized by an activation duration of three time slots and a demand pattern of type 2 (increasing for the second time slot and decreasing for the third one). In Figure 8b, the entries for which $y_{s,\ell}(i,v) = \max\{\kappa_{s,\ell}(i,v) - \kappa_{s,\ell\text{-}1}(i,v), 0\} > 0$ are highlighted in red, indicating an increase of the reserved resources for slice s

TABLE II
PARAMETERS OF U-RD, SFC-RD, AND S-RD

Type 1: HD video streaming at 6 Mbps, $\underline{p}_s^{\text{sp}} = 0.99$

Node	$(\overline{U}_{s,\text{c}}, \widetilde{U}_{s,\text{c}})$	$(\overline{U}_{s,\text{m}}, \widetilde{U}_{s,\text{m}})$	$(\overline{U}_{s,\text{w}}, \widetilde{U}_{s,\text{w}})$	$(r_\text{c}, r_\text{m}, r_\text{w})$	*Link*	$(\overline{U}_{s,\text{b}}, \widetilde{U}_{s,\text{b}})$	$r_{s,\text{b}}$
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 4 Mbps, $\underline{p}_s^{\text{sp}} = 0.95$

Node	$(\overline{U}_{s,\text{c}}, \widetilde{U}_{s,\text{c}})$	$(\overline{U}_{s,\text{m}}, \widetilde{U}_{s,\text{m}})$	$(\overline{U}_{s,\text{w}}, \widetilde{U}_{s,\text{w}})$	$(r_\text{c}, r_\text{m}, r_\text{w})$	*Link*	$(\overline{U}_{s,\text{b}}, \widetilde{U}_{s,\text{b}})$	$r_{s,\text{b}}$
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 2 Mbps, $\underline{p}_s^{\text{sp}} = 0.9$

Node	$(\overline{U}_{s,\text{c}}, \widetilde{U}_{s,\text{c}})$	$(\overline{U}_{s,\text{m}}, \widetilde{U}_{s,\text{m}})$	$(\overline{U}_{s,\text{w}}, \widetilde{U}_{s,\text{w}})$	$(r_\text{c}, r_\text{m}, r_\text{w})$	*Link*	$(\overline{U}_{s,\text{b}}, \widetilde{U}_{s,\text{b}})$	$r_{s,\text{b}}$
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	—	$(0.214, 0.003, 0)$			

during consecutive time slots. Comparing Figure 8a, where $c_\text{a} = 0$ and Figure 8b, where $c_\text{a} > 0$, one observes that the number of adjustments of node assignment $\kappa_{s,\ell}(i, v)$ is reduced when $c_\text{a} > 0$, as expected.

2) Resource Reservation for Multiple Slices: In this simulation, 1000 slice requests are generated among which 250 are tagged as Premium uniformly at random. Four choices are considered for the parameters α and ΔP, all with $P_\text{max} = 3$, see Section IV-A. These choices impact the processing strategy of Premium and Standard slice requests. When $(\alpha, \Delta P) = (0.5, 0)$, Premium requests are processed immediately and Standard requests are processed in the time slot preceding their activation time slot. When $\alpha = 0$, whatever the value of ΔP, Premium and Standard requests are processed immediately, starting with the Premium requests. With $(\alpha, \Delta P) = (0.5, 0.5)$ and $(\alpha, \Delta P) = (0.5, 1)$, intermediate processing delays are obtained for Standard slices. We also evaluate the J-PR and S-PR variants with slice resource reservation requests processed just before slice activation (*just-in-time* processing). This approach is close to that considered, *e.g.*, in [14]. Nevertheless, Premium slice requests are still processed first.

Figure 9 compares the performance of the J-PR and S-PR reservation variants considering the four slice requests processing strategies induced by the choices of $(\alpha, \Delta P)$ and the *just-in-time* approach.

The average response delay (expressed as a multiple of T) for each slice request is shown in Figure 9a. The J-PR and S-PR variants share the same prioritized processing policy, therefore, both variants provide the same result in terms of response delay. When $\alpha = 0$, all requests are processed immediately, independently of their priority. The observed delay is only due to the processing which takes place at the end of each time slot during the processing time interval of duration εT. When $\alpha = 0.5$, the processing delay remains constant for Premium slices and increases when ΔP decreases for Standard slices. As expected, the delay is maximum for the *just-in-time* processing.

Figure 9b illustrates the acceptance rate for the various processing strategies. Processing the slices jointly yields a slightly higher acceptance rate compared to a sequential approach. The acceptance rate of Premium slice requests is higher than that of Standard ones. The difference decreases when the average processing delay of Standard slice requests decreases. The difference is minimum when Standard slices are processed just after Premium slices in the same processing slot, *i.e.*, when $(\alpha, \Delta P) = (0, \cdot)$ or with the *just-in-time* approach. Selecting the processing strategy allows one to adjust the acceptance rate difference between Premium slices and Standard slices. In practice, the income associated to both types of slices, as well as the share among these types of slice requests may help the MNO in determining the best value of the pair $(\alpha, \Delta P)$, *e.g.*, that maximizes its expected income.

Figure 9c illustrates the average number of adjustments of node assignments $y_{s,\ell}(i, v)$ per slice and per time slot. A joint approach is again more efficient than a sequential approach. Moreover, when the processing delay of Standard slices decreases, the number of adjustments for Standard slices decreases too, while the average number of adjustments of node assignments increases for Premium slices. This is explained by the fact that delaying more the processing of Standard slices facilitates finding assignments with fewer adjustments during the lifetime of Premium slices. The price to be paid is more adjustments for Standard slices. The number of adjustments is maximum with the *just-in-time* approach.

Figure 9d shows the average per-slice reservation cost charged by the InP to the MNO. Joint resources reservation leads to lower costs compared to a sequential reservation. The reservation costs increase for Premium slices when the processing of Standard slices is less delayed. For Standard and Premium slices, the reservation costs are consistent with the evolution of the average number of adjustments of node assignments observed in Figure 9c. The costs for the *just-in-*

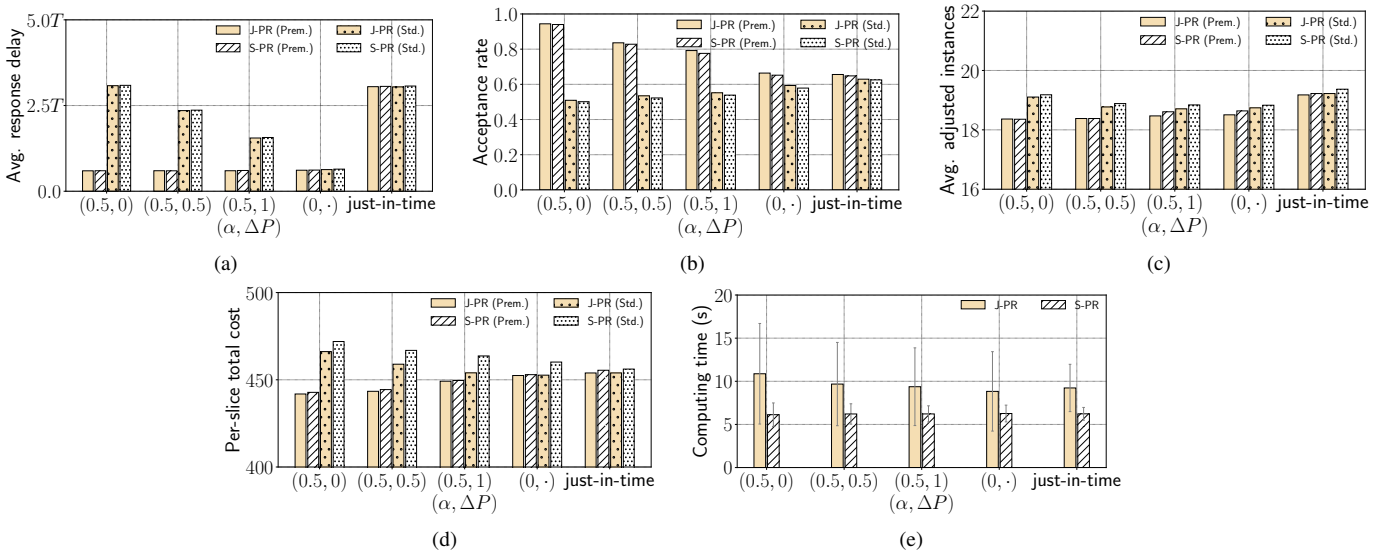


Fig. 9. Performance comparison of the different processing strategies $(\alpha, \Delta P)$ with the J-PR and S-PR variants, in terms of (a) average response delay (expressed as a multiple of T), (b) acceptance rate of slice requests, (b) average adjusted instances, (d) average cost per slice, and (e) computing time.

time approach are similar to those obtained when the slices requests are processed as they arrive, when $(\alpha, \Delta P) = (0, \cdot)$.

Figure 9e shows that the computing times are independent of the processing strategy of Premium and Standard slice requests. As expected the S-PR variant is less time-consuming than the J-PR variant, due to the reduced number of variables involved. The computing time is significantly less than the typical time slot duration T when it is of few tens of minutes. As discussed in Section V-D3, there is a linear relation between the number of nodes and links of the infrastructure network and the number of variables and constraints of the max-min optimization problem. The computational complexity is exponential in the worst case in the number of variables. The fact that reservation slot duration is of a few tens of minutes allows InPs to deal with moderate-sized infrastructure networks. Nevertheless, for large networks, additional heuristics have to be developed to be able to employ the proposed approach.

VII. Conclusions and Perspectives

This paper considers a network slicing scenario with slice requests characterized by variable delays between their submission and activation and by different priority levels (*e.g.*, Premium and Standard). Considering these hypotheses, we introduce a prioritized slice resource reservation and admission control mechanism. Resources required for admitted slices are reserved, and admission decisions are provided with a response delay depending on the slice priority and on time left before its activation.

Adopting the perspective of the InPs, slice resource reservation and admission control is formulated as a max-min optimization problem. The InP aims to maximize the number of admitted slices, *i.e.*, slices for which enough resources can be reserved, while minimizing the cost charged to the MNOs. Uncertainties in the slice resource demands and the presence of background service sharing the infrastructure are taken into account. Two reduced-complexity resource reservation variants, namely J-PR and S-PR, are proposed to solve the max-min problem.

Numerical results show that the proportion of admitted slices can be adjusted depending on the difference in the processing delay between Premium and Standard slices. When the delay difference increases, resource reservation requests for Premium slices are granted significantly more frequently, with fewer adjustments with time in the reservation scheme. This directly impacts the reservation costs, which are reduced for Premium slices compared to Standard slices when the delay difference is large. We also illustrate the benefits in terms of satisfaction differentiation for Premium slice reservation requests provided by an anticipated processing, compared to a *just-in-time* processing.

The approach presented in this paper may be incorporated in more realistic simulation environments such as that proposed in [14] to confirm the additional flexibility provided to MNO by differentiated processing of resource reservation requests.

In the simulations presented in this work, only resource reservation requests for finite-duration slices have been considered. Periodic resource reservation requests or requests for slices of very long duration may be considered without changing the approach. Nevertheless, this would significantly increase the number of variables to store the reservation decisions for such requests and would require additional developments to the proposed approach.

Another possible extension of this work is to enable an adjustment in future time slots of the resource reservation scheme for slices that have been admitted, as long as they are not yet activated. In a given time slot, when Premium slices are processed, often several assignments lead to the same costs. As seen in [44], accounting for known slice requests that will have to be processed in future time slots, may help in the selection among assignments of Premium slices requests with the same costs, to finally reduce the adaptation cost of future reservation assignments. Radio coverage constraints

may also be considered in the resource reservation process, using an approach inspired, *e.g.*, from [34]. Accounting for user mobility would also require a model of the mobility patterns of typical slice users.

APPENDIX A
DERIVATION OF MEAN AND VARIANCE OF S-RD

Consider an active slice s within time slot $\ell \in \mathcal{K}_s$. The number of users $N_{s,\ell}$ of slice s and the resource demands $U_{s,n,\ell}(v)$ and $U_{s,\mathrm{b},\ell}(vw)$, $v \in \mathcal{N}_s$, $vw \in \mathcal{E}_s$, $n \in \Upsilon$, of each user of this slice are assumed as independently distributed. Denoting $\mathbb{E}(N_{s,\ell}) = \overline{N}_{s,\ell}$, $\mathrm{Var}(N_{s,\ell}) = \widetilde{N}^2_{s,\ell}$, $\mathbb{E}[U_{s,n,\ell}(v)] = \overline{U}_{s,n,\ell}(v)$, and $\mathrm{Var}(U_{s,n,\ell}(v)) = \widetilde{U}^2_{s,n,\ell}(v)$, the mean value and variance of $R_{s,n,\ell}(v)$, can be evaluated, $\forall n \in \Upsilon$ and $\forall v \in \mathcal{N}_s$, as

$$\begin{aligned}\overline{R}_{s,n,\ell}(v) =& \mathbb{E}(N_{s,\ell}U_{s,n,\ell}(v)) = \overline{N}_{s,\ell}\overline{U}_{s,n,\ell}(v), \qquad (53)\\ \widetilde{R}^2_{s,n,\ell}(v) =& \overline{N}^2_{s,\ell}\widetilde{U}^2_{s,n,\ell}(v) + \overline{U}^2_{s,n,\ell}(v)\,\widetilde{N}^2_{s,\ell}\\ &+ \widetilde{N}^2_{s,\ell}\widetilde{U}^2_{s,n,\ell}(v), \qquad (54)\end{aligned}$$

see [45]. Similarly, for all $vw \in \mathcal{E}_s$, one obtains

$$\begin{aligned}\overline{R}_{s,\mathrm{b},\ell}(vw) &= \overline{N}_{s,\ell}\overline{U}_{s,\mathrm{b},\ell}(vw), \qquad (55)\\ \widetilde{R}^2_{s,\mathrm{b},\ell}(vw) &= \overline{N}^2_{s,\ell}\widetilde{U}^2_{s,\mathrm{b},\ell}(vw) + \overline{U}^2_{s,\mathrm{b},\ell}(vw)\,\widetilde{N}^2_{s,\ell}\\ &\quad + \widetilde{N}^2_{s,\ell}\widetilde{U}^2_{s,\mathrm{b},\ell}(vw). \qquad (56)\end{aligned}$$

APPENDIX B
RELAXATION OF THE SSP CONSTRAINT

For a given slice $s \in \mathcal{R}_k$, , the MNO has to determine for each time slot $\ell \in \mathcal{K}_s$ the smallest value of $\gamma_{s,\ell}$ such that the satisfaction of (33), (34) implies that of (28).

Consider the following probability

$$\begin{aligned}p_{s,\ell}(\gamma_{s,\ell}) = \Pr\Big\{&\widehat{R}_{s,n,\ell}(v,\gamma_{s,\ell}) \geqslant R_{s,n,\ell}(v), \forall v, n,\\ &\widehat{R}_{s,\mathrm{b},\ell}(vw,\gamma_{s,\ell}) \geqslant R_{s,\mathrm{b},\ell}(vw), \forall vw\Big\},\end{aligned} \qquad (57)$$

which only involves the slice resource demands as well as $\widehat{R}_{s,n,\ell}(v,\gamma_{s,\ell})$ and $\widehat{R}_{s,\mathrm{b},\ell}(vw,\gamma_{s,\ell})$ introduced in (35) and (36). For a given value of $\gamma_{s,\ell}$, if $\boldsymbol{\kappa}_{s,\ell}$ is such that (33), (34) are satisfied, then, from (57), one has

$$p_{s,\ell}(\boldsymbol{\kappa}_{s,\ell}, d_s) \geqslant p_{s,\ell}(\gamma_{s,\ell}). \qquad (58)$$

Consequently, choosing $\gamma_{s,\ell}$ such that $p_{s,\ell}(\gamma_{s,\ell}) \geqslant \underline{p}^{\mathrm{sp}}_s$ implies the satisfaction of (28).

Using (8) and (9), one has

$$p_{s,\ell}(\gamma_{s,\ell}) = \sum_{\eta=1}^{m_{s,\ell}} p_\eta \int_{\widehat{\mathcal{R}}(\gamma_{s,\ell})} f\left(\mathbf{x}, \eta\boldsymbol{\mu}, \eta^2\boldsymbol{\Gamma}\right) \mathrm{d}\mathbf{x}, \qquad (59)$$

where $\widehat{\mathcal{R}}(\gamma_{s,\ell}) = \left\{\mathbf{x} \in \mathbb{R}^{n_s} \,|\, \mathbf{x} \leqslant \widehat{\mathbf{R}}(\gamma_{s,\ell})\right\}$ with

$$\begin{aligned}\widehat{\mathbf{R}}(\gamma_{s,\ell}) = \big(&\widehat{R}_{s,\mathrm{c},\ell}(v_1,\gamma_{s,\ell}), \widehat{R}_{s,\mathrm{m},\ell}(v_1,\gamma_{s,\ell}), \ldots\\ &\ldots, \widehat{R}_{s,\mathrm{b},\ell}(v_1v_2,\gamma_{s,\ell}), \ldots\big)^\top\end{aligned} \qquad (60)$$

of size $n_s = |\Upsilon|\,|\mathcal{N}_s| + |\mathcal{E}_s|$. If the pmf p_η of the number $N_{s,\ell}$ of users is known, the value of $\gamma_{s,\ell}$ such that $p_{s,\ell}(\gamma_{s,\ell}) = \underline{p}^{\mathrm{sp}}_s$ can be obtained by bisection search methods, see, *e.g.*, [46]. The multidimensional integral in (59) can be evaluated, *e.g.*, using a quasi-Monte Carlo integration algorithm presented in [47].

The evaluation of $\gamma_{s,k}$ can be performed as soon as the reservation request for slice s is received, prior to the optimization of the reservation.

APPENDIX C
RELAXATION OF THE IP CONSTRAINT

For a given slice $s \in \mathcal{R}_k$ and an assignment $\boldsymbol{\kappa}_{s,\ell}$, $\ell \in \mathcal{K}_s$ that satisfies (37, 38) and (22, 23, 26), the IP defined in (31) can be evaluated as, $\forall i \in \mathcal{N}$, $\forall n \in \Upsilon$,

$$\begin{aligned}p^{\mathrm{im}}_{n,\ell}(i) &= \Pr\left\{B_{n,\ell}(i) \geqslant \widehat{B}_{n,\ell}(i,\gamma_{\mathrm{B},\ell})\right\}\\ &= 1 - \int_{-\infty}^{\widehat{B}_{n,\ell}(i,\gamma_{\mathrm{B},\ell})} f\left(x; \overline{B}_{n,\ell}(i), \widetilde{B}^2_{n,\ell}(i)\right) \mathrm{d}x\\ &= 1 - \Phi(\gamma_{\mathrm{B},\ell}), \qquad (61)\end{aligned}$$

where Φ is the cumulative distribution function (CDF) of the zero-mean, unit-variance normal distribution. Similarly, the IP defined in (32) can also be evaluated $\forall ij \in \mathcal{E}$ as,

$$\begin{aligned}p^{\mathrm{im}}_{s,\mathrm{b},\ell}(ij) &= \Pr\left\{B_{\mathrm{b},\ell}(ij) \geqslant \widehat{B}_{\mathrm{b},\ell}(ij,\gamma_{\mathrm{B},\ell})\right\}\\ &= 1 - \Phi(\gamma_{\mathrm{B},\ell}). \qquad (62)\end{aligned}$$

Both (61) and (62) are independent of $\boldsymbol{\kappa}_{s,\ell}$, $\forall s \in \mathcal{R}_k$. To satisfy the impact constraints imposed by (31, 32), $\gamma_{\mathrm{B},\ell}$ has to be chosen such that $1 - \Phi(\gamma_{\mathrm{B},\ell}) \leqslant \overline{p}^{\mathrm{im}}$, *i.e.*, such that

$$\gamma_{\mathrm{B},\ell} \geqslant \Phi^{-1}\left(1 - \overline{p}^{\mathrm{im}}\right). \qquad (63)$$

Since the larger $\gamma_{\mathrm{B},\ell}$, the more difficult the satisfaction of (37) and (38), the optimal $\gamma_{\mathrm{B},\ell}$ would be $\gamma_{\mathrm{B},\ell} = \Phi^{-1}\left(1 - \overline{p}^{\mathrm{im}}\right)$.

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] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.
[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] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network Slicing for 5G: Challenges and Opportunities," *IEEE Internet Computing*, 2018.
[8] 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 Netw.*, vol. 33, no. 6, pp. 172–179, 2019.
[9] N. Huin, B. Jaumard, and F. Giroire, "Optimization of Network Service Chain Provisioning," in *Proc. IEEE ICC*, 2017.
[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] 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 Netw.*, vol. 33, no. 6, pp. 172–179, 2019.
[12] 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.
[13] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, and K. Samdanis, "Optimising 5G Infrastructure Markets: The Business of Network Slicing," in *IEEE INFOCOM*, 2017.
[14] J. X. Salvat, L. Zanzi, A. Garcia-Saavedra, V. Sciancalepore, and X. Costa-Perez, "Overbooking Network Slices through Yield-driven End-to-End Orchestration," in

Proc. 14th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT), 2018, pp. 353–365.
[15] K. Noroozi, M. Karimzadeh-Farshbafan, and V. Shah-Mansouri, "Service Admission Control for 5G Mobile Networks with RAN and Core Slicing," in *Proc. IEEE GLOBECOM*. IEEE, 2019, pp. 6–11.
[16] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, and X. Costa-Perez, "A Machine Learning Approach to 5G Infrastructure Market Optimization," *IEEE Trans. Mobile Comput.*, vol. 19, no. 3, pp. 498–512, 2020.
[17] S. Ebrahimi, A. Zakeri, B. Akbari, and N. Mokari, "Joint Resource and Admission Management for Slice-enabled Networks," in *Proc. IEEE NOMS*, 2020, pp. 1–7.
[18] B. Han, V. Sciancalepore, X. Costa-Perez, D. Feng, and H. D. Schotten, "Multiservice-Based Network Slicing Orchestration with Impatient Tenants," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 5010–5024, 2020.
[19] 3GPP, "Management and Orchestration; Concepts, Use Cases and Requirements," *3GPP TS 28.530 V17.0.0*, 2020.
[20] ——, "Management and Orchestration; Provisioning," *3GPP TS 28.531 V16.8.0*, 2020.
[21] 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.
[22] Q.-T. Luu, S. Kerboeuf, and M. Kieffer, "Uncertainty-Aware Resource Provisioning for Network Slicing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 79–93, 2021.
[23] Q.-T. Luu, S. Kerboeuf, A. Mouradian, and M. Kieffer, "Radio Resource Provisioning for Network Slicing with Coverage Constraints," in *Proc. IEEE ICC*, 2020.
[24] 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.
[25] 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.
[26] 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.
[27] 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.
[28] 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.
[29] 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.
[30] 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.
[31] G. Wang, G. Feng, T. Q. S. Quek, S. Qin, R. Wen, and W. Tan, "Reconfiguration in Network Slicing-Optimizing the Profit and Performance," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 591–605, 2019.
[32] N. V. Huynh, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "Real-Time Network Slicing with Uncertain Demand : A Deep Learning Approach," in *Proc. IEEE ICC*, 2019.
[33] 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.
[34] 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.
[35] 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.
[36] J. Tan, P. Dube, X. Meng, and L. Zhang, "Exploiting Resource Usage Patterns for Better Utilization Prediction," in *Proc. ICDCS*. IEEE, 2011, pp. 14–19.
[37] A. Boubendir, F. Guillemin, C. Le Toquin, M. L. Alberi-Morel, F. Faucheux, S. Kerboeuf, J. L. Lafragette, and B. Orlandi, "Federation of Cross-Domain Edge Resources: A Brokering Architecture for Network Slicing," in *Proc. IEEE NetSoft*. IEEE, 2018, pp. 494–499.
[38] 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.
[39] 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.
[40] Y. Dong, Z. Chen, P. Fan, and K. B. Letaief, "Mobility-Aware Uplink Interference Model for 5G Heterogeneous Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 2231–2244, 2016.
[41] G. George, A. Lozano, and M. Haenggi, "Distribution of the Number of Users per Base Station in Cellular Networks," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 520–523, 2019.
[42] Z. Zhao, M. Li, R. Li, and Y. Zhou, "Temporal-Spatial Distribution Nature of Traffic and Base Stations in Cellular Networks," *IET Communications*, vol. 11, no. 16, pp. 2410–2416, 2017.
[43] 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.
[44] Q. T. Luu, S. Kerboeuf, and M. Kieffer, "Foresighted Resource Provisioning for Network Slicing," in *Proc. IEEE HPSR*. IEEE, 2021, pp. 1–8.
[45] L. A. Goodman, "On the Exact Variance of Products," *Journal of the American Statistical Association*, vol. 55, no. 292, pp. 708–713, 1960.
[46] R. L. Burden and J. Douglas Faires, *Numerical Analysis*, 9th ed. Brooks/Cole, Cengage Learning, 2011.
[47] A. Genz, "Numerical Computation of Rectangular Bivariate and Trivariate Normal and t Probabilities," *Statistics and Computing*, vol. 14, no. 3, pp. 251–260, 2004.

Quang-Trung Luu is currently a postdoctoral fellow at the Laboratory for Analysis and Architecture of Systems (LAAS), French National Centre for Scientific Research (CNRS), Toulouse, France. He received a Ph.D in networking and telecommunications from Paris-Saclay University, France in 2021. During the Ph.D, he was also a research engineer at Nokia Bell Labs, France. His research focuses on the optimization of resource management in communication networks, in particular on key enabling technologies for next-generation mobile systems.

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.