mathematics

MDPI

Article

An Analytical Model for 5G Network Resource Sharing with Flexible SLA-Oriented Slice Isolation

Natalia Yarkina [1], Yuliya Gaidamaka [1,2,*], Luis M. Correia [3] and Konstantin Samouylov [1,2]

[1] Applied Informatics and Probability Department, Peoples' Friendship University of Russia (RUDN University), Miklukho-Maklaya St. 6, Moscow 117198, Russia; natyarkina@sci.pfu.edu.ru (N.Y.); samuylov-ke@rudn.ru (K.S.)
[2] Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Vavilov St. 44-2, Moscow 119333, Russia
[3] IST/INESC-ID, Universidade de Lisboa, Rua Alves Redol, 9, 1E, 1000-029 Lisbon, Portugal; luis.m.correia@tecnico.ulisboa.pt
* Correspondence: gaydamaka-yuv@rudn.ru

Received: 16 June 2020; Accepted: 14 July 2020; Published: 17 July 2020

Abstract: Network slicing is a novel key technology in 5G networks which permits to provide a multitude of heterogeneous communication services over a common network infrastructure while satisfying strict Quality of Service (QoS) requirements. Since radio spectrum resources are inherently scarce, the slicing of the radio access network should rely on a flexible resource sharing policy that provides efficient resource usage, fairness and slice isolation. In this article, we propose such a policy for bandwidth-greedy communication services. The policy implies a convex programming problem and is formalized to allow for session-level stochastic modeling. We developed a multi-class service system with service rates obtained as a solution to the optimization problem, a Markovian Arrival Process and state-dependent preemptive priorities. We use matrix-analytic methods to find the steady state distribution of the resulting continuous-time Markov chain and the expressions for important performance metrics, such as data rates. Numerical analysis illustrates the efficiency of the proposed slicing scheme compared to the complete sharing and complete partitioning policies, showing that our approach leads to a data rate about the double of that obtained under complete partitioning for the analyzed scenario.

Keywords: Markov Chain model; matrix-analytic methods; convex programming; preemptive priority; Markovian Arrival Process; network slicing; slice isolation; 5G

1. Introduction

Network slicing is a key novel technology introduced in 5G networks as a response to two challenges:

- to efficiently transmit data with completely different characteristics and Quality of Service (QoS) requirements over the same physical network infrastructure, and
- to provide seamless support for diverse business models and market scenarios, for example, Mobile Virtual Network Operators (MVNO), which do not possess their own network infrastructure yet seek autonomy in administration and admission control.

Network slicing permits creating a multitude of logical networks for different tenants over a common infrastructure. Current industry standards define a network slice as a logical network that provides specific network capabilities and network characteristics [1,2]. According to Reference [2], the dynamic scale-in and scale-out of resources in a slice should be supported and occur with

minimal impact on QoS. ITU-T Recommendations also indicate that slices should be isolated from each other, so that their interference is minimal [3]. Moreover, depending on the slices' requirements, different levels of isolation should be available, which may result, for instance, in the preemption of a slice's resources by another slice with a higher isolation level [2].

Slices can span across several network segments—user equipment (which can belong to multiple slices simultaneously), Radio Access Network (RAN), core network, and so forth—and thus constitute complete end-to-end logical networks. The resources allocated to a slice can be dedicated and/or shared in terms of processing power, storage, bandwidth, and so forth [4]. Since network slicing implies infrastructure sharing, its efficiency is contingent upon finding an adequate policy for allocating resources among slices. In light of the requirements stated above and common industry practice, such a policy should be QoS-oriented and provide

1. efficient resource usage,
2. fair, non-discriminating resource allocation among users, and
3. flexible isolation of slices from one another.

The problem of resource allocation in network slicing has been addressed in many recent studies, some theoretical and of a more general nature, and some technology-specific. One category of works, namely References [5–7], deals with the end-to-end slicing, which relates and extends the virtual network embedding problem [5]. Generally, this approach implies representing the physical network and the virtual network formed by the slices in graph form and finding a mapping between them. Another large category of studies focuses on how to share a particular network resource or a combination of resources among a set of slices. For the most part, this category revolves around RAN and radio spectrum (channel) resources, since the latter inherently cannot be easily scaled up. However, the ways to consider and slice RAN resources vary—while many studies address the slicing of the radio resources of a single base station [8–17], others consider a combination of specific RAN-related resources, namely the base station's radio channel capacity, cache and backhaul bandwidth [18], a cluster of several homogeneous [19–21] or heterogeneous [22–24] base stations, or the capacity of a Cloud-RAN (C-RAN), in which the baseband resources of several homogeneous [25,26] or heterogeneous [27,28] radio systems are centralized into a single resource pool. Besides, some studies deal with finite sets of purely abstract [29,30] or more specific—radio, transport and computing [31]—network resources.

The assumptions regarding the network slices also vary. Some researchers tailor their slicing schemes for the standardized slice types defined in Reference [1] and corresponding to the three categories of 5G services [2]—Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC) and Massive Machine-Type Communications (mMTC) also referred to as Massive Internet of Things (MIoT). RAN resource allocation among such three slices is addressed in Reference [10,15]. Some authors generalize the previous approach and consider a finite set of slices, assuming each slice to carry either data-rate- or delay-sensitive traffic [11,16,26], which loosely corresponds to eMBB and URLLC. The trade-off between data-rate- and delay-sensitivity for each slice can be adjusted through weights in Reference [13], while in References [23,24,31] slices are characterized by delay tolerance and bit rate requested for the slice's user at every base station. In Reference [32], slices are assumed to carry either elastic or inelastic traffics. References [17,18] consider a set of slices, each providing a finite number of services with certain characteristics. Finally, many studies consider a set of slices without specifying the nature of the carried traffic [9,12,14,19–22,25,29]; in this case, the scenario of MVNOs is sometimes exploited, although from a broader perspective than that of the one currently deployed. Yet another approach is adopted in Reference [28], where each slice tenant is assumed to have a Service Level Agreement (SLA) with the infrastructure provider, specifying the slice's bounds in terms of data rate. Three types of slices are thus defined—with upper and lower bounds fixed, with only a lower bound fixed, and unrestricted (best effort).

The network slicing strategies proposed in the literature are aimed at fulfilling manifold criteria, such as to maximize the overall user data rate [15,19,27], to maximize data-rate- and latency-related

utility functions [13,24], to minimize the overall resource usage while providing performance isolation of the slices [25] or meeting data-rate- and latency-related requirements [16,23]. In References [23,24] a fixed bandwidth allocated to the slice in the core network is considered as an additional constraint. The policy proposed in Reference [12] is aimed at satisfying, on average, the prenegotiated resource shares per tenant within an allowed deviation. Preset minimum shares per slice are also assumed in References [8,9,19,29], however in References [8,29] the effective slice capacities are allowed to go below the guaranteed minima, following the slice workload, while in Reference [19] the minima serve as an optimization constraint, hence being reserved. The objective of the slicing policy proposed in Reference [29] is to minimize the weighted sum of the relative differences between allocated and demanded resource shares. QoS provision and fairness are the focus of References [15,30]. The objective in Reference [26] is to maximize the time-averaged expectation of the infrastructure provider's utility, defined as the difference between the income (data rate and delay gain) and cost (power and spectrum resources). Some works adopt auction models, where resource allocation follows the tenants' bids for their slices [14] or for their slice users [20,21]. Some works [16,17] specifically consider slice isolation, which is provided by setting a fixed upper limit on the slice's capacity. Slice isolation via guaranteed minima is proposed in References [8,9,19,29]. Finally, some authors, namely References [15,17,27], take into account slice priorities.

In the literature, the majority of studies, except for the most technology-specific ones, assume each of the considered resources to be homogeneous and divisible, and propose an algorithm for determining slices' sizes (capacities) as shares of the resources' capacities [9,12–15,18,20,21,29,30]. If the services provided in the slice to end users require a certain minimum data rate, such as Guaranteed Bit Rate (GBR) services, the slicing policy may include not only resource allocation but also admission control [17,21,23,27]. Network slicing at the admission-control level is discussed in Reference [33] along with other approaches to RAN slicing, while user blocking probabilities for such systems are estimated in References [17,21]. Besides, since network slices are to be created and terminated dynamically [2], some studies [31,32,34] investigate the slice admission control, that is, whether a new slice instance can/should be admitted to the network. The problem of associating users to base stations under slicing constraints is addressed in References [23,24]. More technology-specific works, such as References [10,11], represent radio resources in the form of a time-frequency grid composed of Resource Blocks (which reflects the physical layer of 4G/5G RAN) and consider not only the share of the total number of Resource Blocks allocated to a slice per time slot, but also their position on the grid, taking account of the corresponding technology-specific constraints.

Two major approaches to the resource slicing problem can be distinguished in the literature. Some authors [12,13,19–21,23,27] propose to perform resource allocation to slices (inter-slice allocation) by means of resource allocation to individual network users with some additional slicing-specific constraints. Then, the slice capacity can be obtained by summing up the resource shares allocated to the corresponding users. Although such a strategy can yield the optimal solution, it may also be computationally inefficient and deprive slice tenants of a certain degree of autonomy and confidentiality [15,22]. The other approach assumes a hierarchical resource slicing (also referred to as two-level [14] or distributed [22]), in which inter-slice allocation is decoupled from resource scheduling to slice users (intra-slice allocation). In this case, the slice shares are determined based upon some aggregated information about slice users and/or additional criteria (e.g., the tenants' bids [14]), after which resources are allocated to slice users within the computed slice capacities. Furthermore, some researchers assume the two steps of the hierarchical RAN slicing to be performed on the same time scale [8,15,22], usually every Transmission Time Interval, while others consider a longer slicing/decision interval [9,11], which may imply traffic prediction using machine learning techniques [25,31].

The predominant mathematical methods applied in slicing-related resource-sharing problems are those of optimization theory—the resource shares allocated to slices/slice users are often determined as the solution to a linear or non-linear optimization problem [7,11–13,15,16,18,19,23,24,26–28,31].

Algorithmic resource slicing policies are proposed in References [8,9]. Besides, numerous works opt for game theory methods [14,20–22]. Machine learning along with Markov decision processes are applied in References [25,31,32]. Continuous-Time Markov Chain (CTMC) models for network slicing analysis are proposed in References [17,34]. The authors of Reference [34] consider a one-dimensional CTMC which transitions represent slice instantiation and termination, and use it to establish a flexible auction-based slice-admission policy. A four-dimensional CTMC which transitions correspond to establishing and terminating user sessions is proposed in Reference [17] to analyze the performance of resource allocation to two network slices each offering two GBR services with different priorities.

In this article, we propose a flexible and highly customizable resource slicing scheme that satisfies the important criteria given above—efficient resource usage, fairness and performance isolation of slices. The scheme is intended for a QoS-aware, service-oriented slicing, where each slice is homogeneous with respect to traffic characteristics and QoS requirements. We adopt a hierarchical approach to slicing and focus on the inter-slice allocation, assuming that the intra-slice allocation is performed by slice tenants at their discretion. Overall, our approach and formulation of the inter-slice allocation problem is similar to the one adopted in Reference [15], however, we consider the performance isolation of slices, by which we understand, in particular, that traffic bursts in one slice do not affect performance and QoS in other slices, or, at least, effects thereof are minimized. A similar concept of performance isolation is adopted in References [9,25]. Also, similar to References [17,21,23,27], we assume user admission control as part of our slicing scheme.

The proposed slicing scheme is formulated in such a way that its efficiency and the impact of its various parameters and customizable components can be readily analyzed via session-level analytical modeling. To this end, we make use of the Markov process theory and develop a multi-class service system with state-dependent preemptive priorities, in which job service rates are found as the solution to a non-linear programming problem. Also, we use the Markovian Arrival Process (MAP, [35,36]) for one class of jobs to represent a slice which workload is less regular than that represented by a Poisson process. Being both analytically tractable and highly parametrizable, the MAP allows for applying the matrix-analytic methods and, on the other hand, permits to specify a wide range of arrival processes due to its many independent parameters; moreover, there exist techniques for MAP fitting from real traffic data (see e.g., Reference [37]). Similar to Reference [38], we apply matrix-analytic methods to obtain the stationary distribution of the CTMC representing the system and use it to find expressions for performance measures. Finally, we provide a numerical example, where the analytical model serves to compare the proposed policy with the classical resource sharing schemes—complete sharing (CS) and complete partitioning (CP, sometimes referred to as static slicing in the context of network slicing [20]). An algorithm based upon the Gradient Projection Method [39] is suggested for solving the arising optimization problem.

The contribution of the article is twofold. First, we propose a new resource slicing scheme, which focuses on flexible performance isolation of slices and fairness, and is customizable to reflect both QoS requirements and SLA terms. Second, the scheme is formalized to allow for session-level stochastic modeling, and in order to evaluate its performance we develop a CTMC model—a multi-class service system with state-dependent preemptive priorities, a MAP and the service rates obtained as a solution to a convex programming problem—for which the stationary state distribution is derived.

The article is structured as follows. Section 2 introduces the basic model assumptions. Section 3 details the proposed slicing policy and formalizes its resource allocation and preemption components. In Section 4, we present a multi-class service system for analyzing the performance of the proposed policy for three active slices. The steady-state distribution of the system is derived and the expressions of its main performance metrics are obtained. Section 5 offers numerical results, which give an insight into the performance of the proposed slicing policy, in particular, in comparison with CS and CP. Also, an algorithm for solving the optimization problem based upon the Gradient Projection Method is suggested. Section 6 concludes the article.

2. Basic Assumptions

We consider the downlink transmission from a 4G/5G co-located base station (featuring both LTE and NR radio access technologies) at which radio access resource virtualization and network slicing are implemented [40]. Note that the proposed approach is applicable to a 5G-only network as well, however, although the network architecture for 5G has not yet been standardized (in contrast to its radio interface), the likely upcoming solution is a full integration with 4G (akin to 2G/GSM or 3G/UMTS). Following Reference [28], we suppose that the network slicing is performed at the level of virtualized resources. This approach implies the resource allocation among slices/users in terms of virtual resources, which are then translated into network (physical) resources. Generally speaking, the total amount of virtual resources available for allocation depends on the radio conditions for users at each time instant; however, for simplicity, we assume the total base station capacity available for allocation to be fixed.

Suppose that S slices are active at the base station and denote the set of active slices by $\mathcal{S}$, $S = |\mathcal{S}|$. Denote by $C > 0$ the capacity of the base station (measured in bits per second, bps), that is, the total amount of resources available, and by $C_s \geq 0$ the capacity of slice $s \in \mathcal{S}$. We assume that each slice is intended for a particular type of traffic, for example, for streaming video, videoconferencing, software updates, and so forth, rather than for a mix of diverse traffic administrated by the same tenant (which may be the case namely in the MVNO scenario). Thus, we assume that each slice $s \in \mathcal{S}$ is homogeneous in terms of traffic characteristics and QoS requirements and provides to its users only one service with a data rate not smaller than $a_s > 0$ (bps) and not larger than $b_s \leq C$ (bps). The lower bound may reflect QoS requirements [25,27] and corresponds, for instance, to the data rate necessary for providing a certain maximum tolerable delay [15], while the upper bound may be due to the characteristics of a particular service, implying that a higher data rate does not improve the associated QoS (the transmission of voice is a good example of a service with such characteristics). Note that we assume all capacity/data rate parameters to be real-valued, that is, $C, C_s, a_s, b_s \in \mathbb{R}_+$.

Let n_s denote the number of users in slice $s \in \mathcal{S}$. We suppose that an effective data rate $x_s^{(i)}$, $0 < a_s \leq x_s^{(i)} \leq b_s$, allocated to a slice s user i is variable and depends on the slice's capacity and the number of users currently in it, since the sum of the users' data rates cannot exceed the capacity of the slice, that is, $\sum_{i=1}^{n_s} x_s^{(i)} \leq C_s$. Without loss of generality, we assume further that a slice's resources are equally shared among all its users, that is, $x_s^{(i)} = x_s = \frac{C_s}{n_s}$, $i = 1, \ldots, n_s$. Note that the latter assumption serves only to determine the slices' capacities C_s, $s \in \mathcal{S}$, by the slicing scheme detailed in the next session, however, once the capacities established, the actual allocation among users $x_s^{(i)}$, $i = 1, \ldots, n_s$, is not the subject of this work, and the capacity of a slice can be distributed among its users at the discretion of the slice's tenant (e.g., in function of the radio channel conditions). Finally, we assume that a user can be active in one slice only and have no more than one active connection in it.

A slicing algorithm implemented at the base station is aimed at proving (a) efficient resource usage, (b) fair resource allocation among users, and (c) performance isolation of slices. Clearly, performance isolation cannot be guaranteed for unlimited traffic in all slices, hence, following Reference [32], we assume that performance isolation of slice $s \in \mathcal{S}$ is provided as long as the number n_s of users in this slice does not exceed a threshold G_s, $0 \leq G_s \leq \left\lfloor \frac{C}{a_s} \right\rfloor = \max\left\{ y \in \mathbb{N} : y \leq \frac{C}{a_s} \right\}$. Since $a_s > 0$, the threshold can be set as a share $0 \leq \gamma_s \leq 1$ of the capacity C, in which case $G_s = \left\lfloor \frac{C\gamma_s}{a_s} \right\rfloor$, $s \in \mathcal{S}$. Note that we allow for capacity overbooking [31], and therefore, in the general case, $0 \leq \sum_{s \in \mathcal{S}} \gamma_s \leq S$.

In order to provide flexibility and efficient resource usage, we assume that the booked capacity $\gamma_s C$ is not reserved or strictly guaranteed to slice s, but rather slice s has priority in its allocation. Generally speaking, if the base station is fully loaded, the slices in which the number of users is less or equal to G_s have priority in resource allocation over slices in which the number of users is above G_s (or the share of the base station resources occupied by the slice's users at the minimum data rate a_s is above γ_s).

We suppose that the allocated user data rate x_s is not allowed to drop below a_s, because this results in the violation of the service's QoS requirements. In order to satisfy the minimum data rate requirements and provide performance isolation, admission control with request prioritization is used as a part of the slicing scheme. We assume that a request arriving to a slice where the number of users is under or equal to G_s, if free resources are not enough, will preempt users in "violator" slices—the slices in which the number of users is above the threshold. We define the proposed slicing scheme formally in the next section.

3. Network Slicing with Performance Isolation

Denote $\mathbf{a} = (a_1, \ldots, a_S)^T$ and $\mathbf{b} = (b_1, \ldots, b_S)^T$. We can represent the state of the base station under study by a row vector $\mathbf{n} = (n_s)_{s \in \mathcal{S}}$ and define the state space as

$$\Omega = \{\mathbf{n} \in \mathbb{N}^S : \mathbf{na} \leq C\}. \tag{1}$$

Note that $n_s = 0, 1, \ldots, \lfloor C/a_s \rfloor$. We partition the state space (1) as $\Omega = \Omega_0 \bigcup \Omega_1$ where $\Omega_0 = \{\mathbf{n} \in \Omega : \mathbf{nb} \leq C\}$ and $\Omega_1 = \Omega \setminus \Omega_0$. Also, for each state $\mathbf{n} \in \Omega$ we partition the set of slices as $\mathcal{S} = \mathcal{S}_0(\mathbf{n}) \bigcup \mathcal{S}_+(\mathbf{n})$, where $\mathcal{S}_0(\mathbf{n}) = \{s \in \mathcal{S} : n_s = 0\}$ and $\mathcal{S}_+(\mathbf{n}) = \{s \in \mathcal{S} : n_s > 0\}$.

3.1. Resource Allocation

Denote by $x_s(\mathbf{n})$ the data rate to each user in slice $s \in \mathcal{S}$ in state $\mathbf{n} \in \Omega$. Then, the capacities of slices are obtained as $C_s(\mathbf{n}) = n_s x_s(\mathbf{n})$. We assume, following Reference [41], that allocating data rate $x_s(\mathbf{n}) \in [a_s, b_s]$ to a user of slice s has utility $U_s(x_s)$ to the infrastructure provider, and the functions $U_s(x_s)$, $s \in \mathcal{S}$, are increasing, strictly concave and continuously differentiable of x_s, $a_s \leq x_s \leq b_s$. In this paper we focus on the utility functions

$$U_s(x_s) = \ln x_s, \; s \in \mathcal{S}, \tag{2}$$

proposed by Kelly in Reference [41] for proportionally fair resource sharing, which in our case coincide with max–min fairness [42]. However, other utility functions, which reflect better the nature of the slices' traffic, can also be considered.

For $\mathbf{n} \in \Omega_0$, we set $x_s(\mathbf{n}) = b_s$ and, by consequence, $C_s(\mathbf{n}) = n_s b_s$ for all $s \in \mathcal{S}$.

For $\mathbf{n} \in \Omega_1$, we set $x_s(\mathbf{n}) = b_s$ for $s \in \mathcal{S}_0(\mathbf{n})$ and propose to determine the data rates for $s \in \mathcal{S}_+(\mathbf{n})$ as the solution to a convex programming problem as follows:

$$\sum_{s \in \mathcal{S}_+(\mathbf{n})} w_s(n_s) n_s U_s(x_s) \longrightarrow \max, \tag{3}$$

$$\text{subject to} \quad \sum_{s \in \mathcal{S}_+(\mathbf{n})} n_s x_s = C \tag{4}$$

$$\text{over} \quad \mathbf{x} \in \mathcal{P} = \{\mathbf{x} \in \mathbb{R}^{|\mathcal{S}_+(\mathbf{n})|} : a_s \leq x_s \leq b_s, \; s \in \mathcal{S}_+(\mathbf{n})\}. \tag{5}$$

For each $s \in \mathcal{S}$, the weight function $w_s(n)$ is assumed positive for all $n \in \mathbb{N}$, equal to 1 for $n \leq G_s$, and smaller than 1 and nonincreasing for $n > G_s$. The idea behind such requirements is to ensure max–min fair resource allocation to users as long as the corresponding slices do not exceed their thresholds G_s, and to penalize (squeeze) the "violator" slices, in which the number of users exceeds G_s. In this work, we define the weight functions as

$$w_s(n) = \begin{cases} 1, & n \leq G_s, \\ \dfrac{1}{n - G_s + 1}, & n > G_s. \end{cases} \tag{6}$$

In what follows, unless specifically stated, by w_s we mean $w_s(n_s)$ and by $\mathbf{w}$, the column vector $(w_1(n_1), \ldots, w_S(n_S))^T$.

Since the objective function (3) is differentiable and strictly concave by assumption and the feasible region (4), (5) is compact and convex, there exists a unique maximum for the data rate vector $\mathbf{x}$, which can be found by Lagrangian methods. For the utility functions (2), the unique stationary point of the problem (3), (4) has the coordinates

$$x_s^* = \frac{Cw_s}{\sum_{r \in \mathscr{S}_+(\mathbf{n})} n_r w_r}, \; s \in \mathscr{S}_+(\mathbf{n}), \tag{7}$$

and is located at the intersection of the hyperplane (4) and the open ray $x_s = \frac{w_s}{y}, y > 0, s \in \mathscr{S}_+(\mathbf{n})$. However, if the stationary point (7) does not satisfy the direct constraint (5) then the solution lies on the boundary of the feasible region (4), (5).

In the case when $\mathbf{x}^* \notin \mathscr{P}$, the problem (3)–(5) can be solved numerically via the Gradient Projection Method [39]. The method is based upon the iterative process $\mathbf{d}_k = \mathbf{P}\nabla f(\mathbf{x}_k)$, $\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k \mathbf{d}_k$, $\tau_k > 0 : \mathbf{x}_{k+1} \in \mathscr{P}$, where $\mathbf{P}$ is the projection matrix and $\nabla f(\mathbf{x})$ is the gradient of the objective function. The latter, for the utility (2), equals the column vector

$$\nabla f(\mathbf{x}) = \left(\frac{n_s w_s}{x_s}\right)_{s \in \mathscr{S}_+(\mathbf{n})}. \tag{8}$$

Initially, $\mathbf{P}$ is the projection matrix onto the hyperplane (4):

$$\mathbf{P} = \mathbf{I} - \mathbf{\Pi}^T(\mathbf{\Pi}\mathbf{\Pi}^T)^{-1}\mathbf{\Pi}, \tag{9}$$

where $\mathbf{\Pi} = \mathbf{n}$. However, once a boundary of $\mathscr{P}$ is reached, matrix $\mathbf{\Pi}$ should be appended to include the hyperplane of the corresponding constraint, and the current approximation $\mathbf{x}_k$ should be placed on the boundary. Finally, the intersection point of (4) with the diagonal of $\mathscr{P}$ connecting the points $(a_s)_{s \in \mathscr{S}_+(\mathbf{n})}$ and $(b_s)_{s \in \mathscr{S}_+(\mathbf{n})}$ can be used as the initial approximation $\mathbf{x}_0$.

3.2. Admission Control and Resource Preemption

Let $\mathbf{e}_s$ represent a row vector of order S in which the sth entry is 1 and all others are zero. Let the system be in state $\mathbf{n} \in \Omega$ and denote

$$g(\mathbf{n}) = \sum_{s \in \mathscr{S}} a_s \min\{n_s, G_s\}. \tag{10}$$

Now, an arriving request to slice s will be lost only in two cases—either $\mathbf{n} + \mathbf{e}_s \notin \Omega$ and $n_s + 1 > G_s$, or $\mathbf{n} + \mathbf{e}_s \notin \Omega$, $n_s + 1 \leq G_s$ and $C - g(\mathbf{n}) < a_s$. The arriving request is, conversely, accepted if either $\mathbf{n} + \mathbf{e}_s \in \Omega$, or $\mathbf{n} + \mathbf{e}_s \notin \Omega$, $n_s + 1 \leq G_s$ and $C - g(\mathbf{n}) \geq a_s$. In the latter case resources for the arriving request are freed according to Algorithm 1. Condition $C - g(\mathbf{n}) \geq a_s$ ensures that enough resources are available to preempt. Upon acceptance of the request the system will transition into state $\hat{\mathbf{n}} + \mathbf{e}_s$ where $\hat{\mathbf{n}}$ is obtained via Algorithm 1.

Algorithm 1: Resource preemption upon an arrival into slice s in state $\mathbf{n}$

Input: $s \in \mathcal{S}$, $\mathbf{n} \in \Omega$ such that $\mathbf{n} + \mathbf{e}_s \notin \Omega$, $n_s + 1 \leq G_s$ and $C - g(\mathbf{n}) \geq a_s$

Output: $\hat{\mathbf{n}} \in \Omega$ such that, upon acceptance of the request the system will transition into state $\hat{\mathbf{n}} + \mathbf{e}_s$

$\hat{\mathbf{n}} := \mathbf{n}$ // initialization

repeat

  $\mathcal{R} := \{r \in \mathcal{S} : \hat{n}_r > G_r,\ w_r(\hat{n}_r) = \min_{j \in \mathcal{S}} w_j(\hat{n}_j)\}$ // candidates for preemption

  Choose $r \in \mathcal{R}$ // randomly or according to a preset order

  $\hat{\mathbf{n}} := \hat{\mathbf{n}} - \mathbf{e}_r$

until $C - \hat{\mathbf{n}}\mathbf{a} \geq a_s$

4. A CTMC Model

4.1. Model Assumptions

We suppose that three slices are active at the base station and that re-slicing is performed often enough so that we can assume resources to be reallocated whenever a user connection is established or terminated. Under these assumptions, we can represent the functioning of the base station described in Sections 2 and 3 as a three-class preempt-loss system with elastic jobs and make use of the matrix-analytic methods for its analysis.

Consider a loss system of continuous capacity C with no waiting spaces and three job classes, $\mathcal{S} = \{1, 2, 3\}$. We assume class 1 and 2 jobs to arrive according to Poisson processes (PP) with rates λ_1 and λ_2 respectively. Class 3 jobs form a MAP characterized by two square matrices $\mathbf{Q}_0$ and $\mathbf{Q}_1$ of order K. The matrix $\mathbf{Q} = \mathbf{Q}_0 + \mathbf{Q}_1$ represents the infinitesimal generator of the CTMC $\{\xi(t),\ t \geq 0\}$ that controls the MAP. Matrix $\mathbf{Q}_1$ contains the transition rates of $\{\xi(t),\ t \geq 0\}$ accompanied by an arrival, whereas the off-diagonal entries of $\mathbf{Q}_0$ represent the transition rates of $\{\xi(t),\ t \geq 0\}$ without arrivals. We assume $\mathbf{Q}_1$ non-zero and $\{\xi(t),\ t \geq 0\}$ irreducible. Denote by $\mathbf{1}$ a column vector of ones and by $\mathbf{0}$ a row vector of zeros of appropriate length. The row vector $\mathbf{q}$ of the stationary state probabilities of $\{\xi(t),\ t \geq 0\}$ is determined as the unique solution to the global balance equations $\mathbf{qQ} = \mathbf{0}$, $\mathbf{q1} = 1$. The mean rate of the MAP is given by

$$\lambda_3 = \mathbf{q}\mathbf{Q}_1\mathbf{1}. \tag{11}$$

We assume that jobs are served according to the Egalitarian Processor Sharing (EPS) discipline [43], however, each class s job demands no less than a_s and no more than b_s of system capacity, $0 < a_s \leq b_s \leq C$. Jobs' lengths, that is, the holding times when served by exactly one unit of capacity, are assumed to be exponentially distributed with parameters μ_s, $s \in \mathcal{S}$. An accepted job is served with a variable rate until its remaining length is zero and leaves the system thereafter. The service rate of a class s job varies with the numbers of jobs of each class currently on service, $\mathbf{n} = (n_1, n_2, n_3)$, and is determined by the allocated data rate $x_s(\mathbf{n})$, which is found as the solution to the optimization problem (3)–(6). In what follows the notations $x_s(\mathbf{n})$, $x_s(n_1, \ldots, n_S)$ and $x_s^{(n_1,\ldots,n_S)}$ are equivalent. Finally, jobs are accepted and preempted according to the rules detailed in Section 3.2.

4.2. Stationary State Distribution

Denote by $n_s(t)$ the number of class s jobs on service at time $t \geq 0$, $s \in \mathcal{S}$. The stochastic behavior of the system can be represented by the four-dimensional CTMC $\{\psi(t) = (n_1(t), n_2(t), n_3(t), \xi(t)),\ t \geq 0\}$ over the state space

$$\Psi = \{(\mathbf{n}, k) : \mathbf{n} \in \Omega,\ k = 1, 2, \ldots, K\}, \tag{12}$$

where Ω is given by (1). Now, order the states of $\{\psi(t),\ t \geq 0\}$ lexicographically and let $\mathbf{A}$ represent its infinitesimal generator. We express $\mathbf{A}$ as

$$\mathbf{A} = \tilde{\mathbf{A}} + \mathbf{H}, \tag{13}$$

where $\tilde{\mathbf{A}}$ represents an infinitesimal generator of $\{\psi(t),\ t \geq 0\}$ without preemption and $\mathbf{H}$ contains the preemption rates. Let $L = \lfloor C/a_1 \rfloor$ denote the maximum number of class 1 jobs on service simultaneously. We also introduce a notation for the maximum number of class 2 jobs when l class 1 jobs are on service and, similarly, for the maximum number of class 3 jobs when l class 1 jobs and m class 2 jobs are on service:

$$M(l) = \left\lfloor \frac{C - la_1}{a_2} \right\rfloor,\ l = 0, 1, \ldots, L, \tag{14}$$

$$N(l,m) = \left\lfloor \frac{C - la_1 - ma_2}{a_3} \right\rfloor,\ l = 0, 1, \ldots, L,\ m = 0, 1, \ldots, M(l). \tag{15}$$

Then, $\tilde{\mathbf{A}}$ is a block tridiagonal matrix of the form

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{D}_0 & \mathbf{\Lambda}_0 & & & \\ \mathbf{M}_1 & \mathbf{D}_1 & \mathbf{\Lambda}_1 & \mathbf{0} & \\ & \mathbf{M}_2 & \ddots & \ddots & \\ & & \ddots & \mathbf{D}_{L-1} & \mathbf{\Lambda}_{L-1} \\ & \mathbf{0} & & \mathbf{M}_L & \mathbf{D}_L \end{bmatrix}. \tag{16}$$

The blocks located at the intersection of the ith block row and jth block column are block matrices of block size $\left(\sum_{m=0}^{M(i-1)} (N(i-1,m)+1)\right) \times \left(\sum_{m=0}^{M(j-1)} (N(j-1,m)+1)\right)$ composed of square matrices of order K. The diagonal blocks of $\tilde{\mathbf{A}}$ are also block tridiagonal:

$$\mathbf{D}_l = \begin{bmatrix} \mathbf{\Delta}_{l,0} & \mathbf{\Gamma}_{l,0} & & & \\ \mathbf{B}_{l,1} & \mathbf{\Delta}_{l,1} & \mathbf{\Gamma}_{l,1} & \mathbf{0} & \\ & \mathbf{B}_{l,2} & \ddots & \ddots & \\ & & \ddots & \mathbf{\Delta}_{l,M(l)-1} & \mathbf{\Gamma}_{l,M(l)-1} \\ & \mathbf{0} & & \mathbf{B}_{l,M(l)} & \mathbf{\Delta}_{l,M(l)} \end{bmatrix},\ l = 0, \ldots, L. \tag{17}$$

Here, the blocks located at the intersection of the ith block row and jth block column are block matrices of K-block size $(N(l,i-1)+1) \times (N(l,j-1)+1)$.

Let us introduce subspaces $\tilde{\mathscr{A}}_s = \{\mathbf{n} \in \Omega : (\mathbf{n} + \mathbf{e}_s)\mathbf{a} \leq C\}$, $s \in \mathscr{S}$, containing the states in which class s jobs are accepted directly (without preemption), and the indicator function

$$\chi_{\mathscr{A}}(\mathbf{n}) = \begin{cases} 1,\ \mathbf{n} \in \mathscr{A}, \\ 0,\ \mathbf{n} \notin \mathscr{A}, \end{cases} \quad \mathbf{n} \in \mathbb{N}^S,\ \mathscr{A} \subseteq \mathbb{N}^S.$$

The diagonal blocks of $\mathbf{D}_l$ are block tridiagonal and for $l = 0, \ldots, L$ and $m = 0, \ldots, M(l)$ are of the form

$$\mathbf{\Delta}_{l,m} = \begin{bmatrix} \mathbf{F}_{l,m,0} & \mathbf{Q}_1 & & & \\ x_3^{(l,m,1)}\mu_3\mathbf{I} & \mathbf{F}_{l,m,1} & \mathbf{Q}_1 & \mathbf{0} & \\ & 2x_3^{(l,m,2)}\mu_3\mathbf{I} & \ddots & \ddots & \\ & & \ddots & \mathbf{F}_{l,m,N(l,m)-1} & \mathbf{Q}_1 \\ & \mathbf{0} & & N(l,m)x_3^{(l,m,N(l,m))}\mu_3\mathbf{I} & \mathbf{F}_{l,m,N(l,m)} \end{bmatrix}, \tag{18}$$

where

$$\mathbf{F}_{n_1,n_2,n_3} = \begin{cases} \mathbf{Q}_0 - \left(\lambda_1\chi_{\tilde{\mathcal{A}}_1}(\mathbf{n}) + \lambda_2\chi_{\tilde{\mathcal{A}}_2}(\mathbf{n}) + \sum_{s\in\mathcal{S}} n_s x_s(\mathbf{n})\mu_s\right)\mathbf{I}, & n_3 < N(n_1,n_2), \\ \mathbf{Q} - \left(\lambda_1\chi_{\tilde{\mathcal{A}}_1}(\mathbf{n}) + \lambda_2\chi_{\tilde{\mathcal{A}}_2}(\mathbf{n}) + \sum_{s\in\mathcal{S}} n_s x_s(\mathbf{n})\mu_s\right)\mathbf{I}, & n_3 = N(n_1,n_2). \end{cases}$$

Super- and subdiagonal blocks of $\mathbf{D}_l$ are rectangular K-block matrices of the form

$$\mathbf{\Gamma}_{l,m} = \begin{bmatrix} \lambda_2\mathbf{I} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \lambda_2\mathbf{I} \\ \mathbf{0} & \ldots & \mathbf{0} \end{bmatrix}, \; l = 0, \ldots, L, \; m = 0, \ldots, M(l) - 1, \tag{19}$$

$$\mathbf{B}_{l,m} = \begin{bmatrix} mx_2^{(l,m,0)}\mu_2\mathbf{I} & & & \mathbf{0} \\ & mx_2^{(l,m,1)}\mu_2\mathbf{I} & \mathbf{0} & \\ & \ddots & & \vdots \\ \mathbf{0} & & mx_2^{(l,m,N(l,m))}\mu_2\mathbf{I} & \mathbf{0} \end{bmatrix}, \quad \begin{matrix} l = 0, \ldots, L, \\ m = 1, \ldots, M(l). \end{matrix} \tag{20}$$

Finally, super- and subdiagonal blocks of $\tilde{\mathbf{A}}$ are block diagonal rectangular matrices of the form

$$\mathbf{\Lambda}_l = \begin{bmatrix} \mathbf{\Lambda}_{l,0} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{\Lambda}_{l,M(l+1)} \\ \mathbf{0} & \ldots & \mathbf{0} \end{bmatrix}, \; \mathbf{\Lambda}_{l,m} = \begin{bmatrix} \mathrm{diag}(\lambda_1\mathbf{I}) \\ \mathbf{0} \end{bmatrix}, \quad \begin{matrix} l = 0, \ldots, L-1, \\ m = 0, \ldots, M(l+1). \end{matrix} \tag{21}$$

$$\mathbf{M}_l = \begin{bmatrix} \mathbf{M}_{l,0} & & \mathbf{0} & \mathbf{0} \\ & \ddots & & \vdots \\ \mathbf{0} & & \mathbf{M}_{l,M(l)} & \mathbf{0} \end{bmatrix}, \tag{22}$$

$$\mathbf{M}_{l,m} = \begin{bmatrix} lx_1^{(l,m,0)}\mu_1\mathbf{I} & & & \mathbf{0} \\ & lx_1^{(l,m,1)}\mu_1\mathbf{I} & \mathbf{0} & \\ & \ddots & & \vdots \\ \mathbf{0} & & lx_1^{(l,m,N(l,m))}\mu_1\mathbf{I} & \mathbf{0} \end{bmatrix}, \quad \begin{matrix} l = 1, \ldots, L, \\ m = 0, \ldots, M(l). \end{matrix} \tag{23}$$

The structure of $\mathbf{H}$ depends on the weights $w_s(\mathbf{n})$, $s \in \mathcal{S}$, therefore we propose to construct it using a general recursive algorithm. We introduce subspaces of states in which an arriving class s job is accepted via preemption of jobs of other classes:

$$\mathcal{H}_s = \{\mathbf{n} \in \Omega : (\mathbf{n} + \mathbf{e}_s)\mathbf{a} > C, \; n_s + 1 \le G_s, \; C - g(\mathbf{n}) \ge a_s\}, \; s \in \mathcal{S}. \tag{24}$$

Here, the first condition means that the arriving job cannot be accepted directly, the condition $n_s + 1 \le G_s$ provides to the arriving job priority over possible "violators", while the last condition assures that in the system there are "violators" preemption of which will permit to vacate enough resources for the arriving job to be accepted. Note that the subspace of all states in which a class s job is accepted equals $\mathscr{A}_s = \tilde{\mathscr{A}}_s \cup \mathscr{H}_s$, and hence the subspace of all states in which an arriving class s job is lost is $\mathscr{B}_s = \Omega \setminus \mathscr{A}_s$. The number of state $\mathbf{n} \in \Omega$ in lexicographical order is given by

$$i(\mathbf{n}) = \sum_{l=0}^{n_1-1} \sum_{m=0}^{M(l)} (N(l,m)+1) + \sum_{m=0}^{n_2-1} (N(n_1,m)+1) + n_3 + 1. \tag{25}$$

Now, the preemption rate matrix $\mathbf{H} = \left(H_{i,j}\right)$, where $H_{i,j}$ are square blocks of order K, $i,j = 1,\ldots,|\Omega|$, can be obtained via Algorithm 2. Note that in Algorithm 2 the "violators" having the same weight are preempted with equal probability. Note also that in step 1 of the algorithm $\tilde{\mathbf{A}}$ can be used for initialization instead of $\mathbf{0}$, in which case the algorithm will produce the infinitesimal generator $\mathbf{A}$. However, the preemption rate matrix $\mathbf{H}$ will be used to compute performance measures in Section 4.3, so we prefer to construct it separately.

Algorithm 2: Construction of the preemption rate matrix $\mathbf{H}$

Input: $\mathscr{H}_s$, $s \in \mathscr{S}$
Output: $\mathbf{H}$

1 $\mathbf{H} := \mathbf{0}$ `// initialization`
2 **foreach** $s \in \mathscr{S}$ **do**
3 **foreach** $\mathbf{n} \in \mathscr{H}_s$ **do**
4 $\alpha := a_s - (C - \mathbf{na})$ `// amount of resources to find`
5 `PREEMPTION(`s, $\mathbf{n}$, $\mathbf{n}$, α, 1`)`
6 **procedure** `PREEMPTION(`s, $\mathbf{n}$, $\hat{\mathbf{n}}$, α, p`)`**:**
7 **if** $\alpha \le 0$ `// resources are found`
8 **then**
9 **if** $s \ne 3$ **then**
10 $H_{i(\mathbf{n}),i(\hat{\mathbf{n}}+\mathbf{e}_s)} := H_{i(\mathbf{n}),i(\hat{\mathbf{n}}+\mathbf{e}_s)} + p\lambda_s \mathbf{I}$
11 $H_{i(\mathbf{n}),i(\mathbf{n})} := H_{i(\mathbf{n}),i(\mathbf{n})} - p\lambda_s \mathbf{I}$
12 **else**
13 $H_{i(\mathbf{n}),i(\hat{\mathbf{n}}+\mathbf{e}_s)} := H_{i(\mathbf{n}),i(\hat{\mathbf{n}}+\mathbf{e}_s)} + p\mathbf{Q}_1$
14 $H_{i(\mathbf{n}),i(\mathbf{n})} := H_{i(\mathbf{n}),i(\mathbf{n})} - p\mathbf{Q}_1$
15 **else**
16 $\mathscr{R} := \{r \in \mathscr{S} : \hat{n}_r > G_r,\ w_r(\hat{n}_r) = \min\limits_{j \in \mathscr{S}} w_j(\hat{n}_j)\}$ `// candidates for preemption`
17 **foreach** $r \in \mathscr{R}$ **do**
18 `PREEMPTION(`s, $\mathbf{n}$, $\hat{\mathbf{n}} - \mathbf{e}_r$, $\alpha - a_r$, $p\frac{1}{|\mathscr{R}|}$`)`

Since under our assumptions $\{\psi(t),\ t \ge 0\}$ is irreducible and its state space Ψ is finite, the stationary distribution of $\{\psi(t),\ t \ge 0\}$ exists. Let us write it in vector form in accordance with the partitioning (16) of $\tilde{\mathbf{A}}$ into blocks: $\mathbf{p} = (\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_L)$, where $\mathbf{p}_l = (\mathbf{p}_{l,0}, \mathbf{p}_{l,1}, \ldots, \mathbf{p}_{l,M(l)})$, $l = 0,\ldots,L$, $\mathbf{p}_{l,m} = (\mathbf{p}_{l,m,0}, \mathbf{p}_{l,m,1}, \ldots, \mathbf{p}_{l,m,N(l,m)})$, $m = 0,\ldots,M(l)$, and $\mathbf{p}_{l,m,n} = (p_{l,m,n,1}, p_{l,m,n,2}, \ldots, p_{l,m,n,K})$, $n = 0,\ldots,N(l,m)$. Vector $\mathbf{p}$ satisfies the global balance equations

$$\mathbf{pA} = \mathbf{0},\ \mathbf{p1} = 1. \tag{26}$$

We note that while $\tilde{\mathbf{A}}$ is block-tridiagonal, adding $\mathbf{H}$ may disrupt this structure, since the positions of non-zero entries in $\mathbf{H}$ depend on the ratios among a_s, G_s and $w_s(n_s)$. This makes it difficult to apply special block-matrix methods for solving the global balance equations, such as in References [38,44]. However, $\mathbf{A}$ is sparse, which allows for the use of sparse linear systems' routines such as UMFPACK.

4.3. Performance Measures

Using the stationary state distribution of the system, we can easily obtain a number of its performance measures. In particular, for $s \in \mathcal{S}$, the mean number of jobs in service, which represents the mean number of users in a slice, is given by

$$\bar{N}_s = \sum_{\mathbf{n}\in\Omega} n_s \mathbf{p}_{n_1,n_2,n_3}\mathbf{1} = \sum_{n_1=0}^{L} \sum_{n_2=0}^{M(n_1)} \sum_{n_3=0}^{N(n_1,n_2)} n_s \mathbf{p}_{n_1,n_2,n_3}\mathbf{1}. \tag{27}$$

The average user data rate in slice s corresponds to the capacity occupied by one class s job when class s jobs are present in the system:

$$\bar{x}_s = \frac{\sum\limits_{\mathbf{n}\in\Omega:n_s>0} x_s(\mathbf{n})\mathbf{p}_{n_1,n_2,n_3}\mathbf{1}}{\sum\limits_{\mathbf{n}\in\Omega:n_s>0} \mathbf{p}_{n_1,n_2,n_3}\mathbf{1}}. \tag{28}$$

The average capacity of slice s is given by

$$\bar{C}_s = \sum_{\mathbf{n}\in\Omega} n_s x_s(\mathbf{n})\mathbf{p}_{n_1,n_2,n_3}\mathbf{1}. \tag{29}$$

The blocking probability in slice s, that is, the probability that a slice s user will not receive full service, corresponds to the loss probability of a class s job, which is the sum of the probability for the job to be lost at arrival and the probability to be preempted: $B_s = B_s^{arr} + B_s^{pr}$. The loss probabilities upon arrival are given by

$$B_s^{arr} = \sum_{\mathbf{n}\in\mathcal{B}_s} \mathbf{p}_{n_1,n_2,n_3}\mathbf{1},\ s = 1,2, \tag{30}$$

$$B_3^{arr} = \frac{1}{\lambda_3} \sum_{\mathbf{n}\in\mathcal{B}_3} \mathbf{p}_{n_1,n_2,n_3}\mathbf{Q}_1\mathbf{1}. \tag{31}$$

To obtain the preemption probabilities we make use of the fact that all the transition rates related to preemption are combined in matrix $\mathbf{H}$:

$$B_s^{pr} = \frac{1}{\lambda_s} \sum_{\mathbf{n}\in\Omega} \mathbf{p}_{n_1,n_2,n_3} \sum_{\hat{\mathbf{n}}\in\Omega:n_s>\hat{n}_s} (n_s - \hat{n}_s) H_{i(\mathbf{n}),i(\hat{\mathbf{n}})}\mathbf{1},\ s \in \mathcal{S}. \tag{32}$$

Note that B_s^{pr} represents the probability for any incoming slice s user to be preempted, while the probability for an accepted user to be preempted is

$$P_s = \frac{B_s^{pr}}{1 - B_s^{arr}},\ s \in \mathcal{S}. \tag{33}$$

Besides, the total blocking probability can be also found via the arrival and departure rates:

$$B_s = \frac{\lambda_s - \mu_s \bar{C}_s}{\lambda_s},\ s \in \mathcal{S}. \tag{34}$$

The overall blocking probability in the system is given by

$$B = \frac{\sum_{s\in\mathcal{S}} B_s\lambda_s}{\sum_{s\in\mathcal{S}} \lambda_s}. \tag{35}$$

Finally, Little's Law provides the expression for the average user session duration (job holding time) making use of $\bar{N}_s$ and B_s^{arr}:

$$T_s = \frac{\bar{N}_s}{\lambda_s(1-B_s^{arr})},\ s\in\mathcal{S}. \tag{36}$$

5. Numerical Results

Suppose that three slices are instantiated at a base station of capacity 40 Mbps. Each slice provides a service that implies transferring files of size 2.5 MB on average with rates no less than 2 Mbps for slice 1 and no less than 1 Mbps for slices 2 and 3. We model this scenario using the service system presented in Section 4 with the following parameter values: $\mathcal{S} = \{1,2,3\}$, $C = 40$, $\mathbf{a} = (2,1,1)^T$, $b_s = C$ and $\mu_s = 0.05$, $s\in\mathcal{S}$. Additionally, we set $\lambda_s = 0.55$ for all $s\in\mathcal{S}$ and specify the MAP of slice 3 by

$$\mathbf{Q}_0 = \begin{bmatrix} -0.143625 & 0.023625 \\ 1 & -5.55 \end{bmatrix},\ \mathbf{Q}_1 = \begin{bmatrix} 0.02 & 0.1 \\ 0.15 & 4.4 \end{bmatrix}. \tag{37}$$

The MAP has been chosen so that its fundamental rate equals that of the other two arrival processes, but the variance of the interarrival time is substantially higher—20.875 vs. 3.306.

To obtain the numerical results presented in this section, we used UMFPACK routines for solving the global balance equations (26), while the optimization problem (3)–(5) was solved for $\mathbf{n}\in\Omega_1$ such that $\mathcal{S}_0(\mathbf{n}) = \emptyset$ using the Gradient Projection Method via Algorithm 3.

First, we set the booked capacity shares of the slices equal to each other, $\gamma_s = \gamma$, $s\in\mathcal{S}$, and vary γ from 0 to 1 with step 0.025 (which yields overbooking for $\gamma > 1/3$). The charts in Figures 1–3 permit to compare the proposed slicing scheme with the CS and CP policies in terms of blocking probabilities (Figure 1) average user data rate (Figure 2) and average user session duration (Figure 3). Indeed, $\gamma = 1$ corresponds to CS with max–min resource allocation and first-come, first served (FCFS) discipline, since all the weights equal 1 and there are no preemptive priorities among slices. Such CS provides high resource utilization and max–min fair resource allocation to users but no slice isolation. The values of the performance measures under CS are not explicitly indicated in Figures 1–3; they can be found on the corresponding curves for the slicing policy for $\gamma = 1$. The respective performance measures under CP such that the capacity of each slice is fixed and equals exactly $C/3$ are shown by dashed lines. These values were obtained separately for each slice via the CTMC model of Section 4 with the rates of the arrival processes corresponding to the other two slices set to zero and the system's capacity set to $C/3$. Note that CP provides perfect slice isolation but may result in poor utilization and fairness.

The zigzag shape of the curves is due to the fact that, as we have stated in Section 2, γ_s is not used in the slicing scheme directly but through parameter G_s via the relation $G_s = \left\lfloor \frac{C\gamma_s}{a_s} \right\rfloor$, $s\in\mathcal{S}$. As γ grows from 0 to 1, G_1 increases by 1 at $\gamma = 0.05n$, $n = 1,2,\ldots,20$, while $G_{2,3}$ increase by 1 at $\gamma = 0.025n$, $n = 1,2,\ldots,40$. In particular, this explains that the average user data rate in slice 2 is higher than that in slice 1 for small $\gamma > 0$ (see Figure 2). Indeed, for $\gamma = 0.025$ we have $G_1 = 0$ and $G_{2,3} = 1$, which gives advantage to slices 2 and 3 over slice 1 in resource allocation. This advantage is slightly compensated at $\gamma = 0.05$, since here G_1 increases to 1, while $G_{2,3} = 2$, but becomes important again at $\gamma = 0.075$ since G_1 is still 1 while $G_{2,3} = 3$.

Overall, we can see that the proposed slicing scheme leads to a much better network performance compared to the CP policy (static slicing) not only in terms of the blocking probability (which are almost by one order of magnitude greater under CP for all slices), but also when considering average data rate (which are roughly half as large under CP) and average session duration (factors depending

on the slices due to their different characteristics). A higher blocking probability in slice 1 compared to slice 2 under CP or CS is due to a higher minimum data rate in slice 1, while an even higher blocking probability in slice 3 stem from the MAP parameters. As expected, the difference in the arrival processes' characteristics between slices 1 and 2, on the one side, and slice 3, on the other side, clearly determine the behavior of the curves, with $\gamma = 1/3$ being an obvious turning point. Figures 1–3 demonstrate that the slicing policy protects slices 1 and 2 from the irregular slice 3 traffic—for smaller γ through resource allocation, and for larger γ also through resource preemption. Indeed, for $\gamma = 0$ there is no preemption of resources, but the more users a slice has the less capacity per user it receives. Interestingly, the overall blocking probability for $\gamma = 0$ is slightly lower than for $\gamma = 1$.

Algorithm 3: Numerical solution of (3)–(5) using the Gradient Projection Method

Input: $\mathbf{n} \in \Omega_1 : \mathcal{S}_0(\mathbf{n}) = \emptyset$, some small $\delta_0 > 0$
Output: solution $\mathbf{x}$ to problem (3)–(5)

$x_s := \dfrac{Cw_s}{\mathbf{nw}}, s \in \mathcal{S}$ `// the stationary point`
if $\mathbf{x} \notin \mathcal{P}$ **then**
  Choose $\hat{\mathbf{x}} \in \mathcal{P}$ such that $\mathbf{n}\hat{\mathbf{x}} = C$ `// for example, the point of intersection with the diagonal of` $\mathcal{P}$ `connecting` $(a_1,\ldots,a_S)$ `and` $(b_1,\ldots,b_S)$
  $\mathbf{\Pi} := \mathbf{n}$ `// the matrix of constraints to be projected on`
  $\mathbf{P} := \mathbf{I} - \dfrac{1}{\mathbf{n}\mathbf{n}^T}\mathbf{n}^T\mathbf{n}$ `// the initial projection matrix`
  repeat
    $\mathbf{d} := \mathbf{P}\nabla f(\hat{\mathbf{x}})$
    $\mathbf{x} := \hat{\mathbf{x}} + \mathbf{d}$
    if $\mathbf{x} \notin \mathcal{P}$ **then**
      Find the constraint $x_i = h_i$ that was violated *first* when moving from $\hat{\mathbf{x}}$ toward $\mathbf{x}$ `// here,` h_i `equals either` a_i `or` b_i
      $\mathbf{x} := (x_i = h_i) \cap \hat{X}X$
      if $\mathbf{\Pi}$ *has less than* $S-1$ *rows* **then**
        Append $\mathbf{\Pi}$ with $\mathbf{e}_i$
      else
        Replace the last row of $\mathbf{\Pi}$ with $\mathbf{e}_i$
      $\mathbf{P} := \mathbf{I} - \mathbf{\Pi}^T(\mathbf{\Pi}\mathbf{\Pi}^T)^{-1}\mathbf{\Pi}$ `// the projection matrix updated`
    $\delta := \|\hat{\mathbf{x}} - \mathbf{x}\|$
    $\hat{\mathbf{x}} := \mathbf{x}$
  until $\delta < \delta_0$
return $\mathbf{x}$

While Figures 1–3 give insight into the functioning of the proposed slicing scheme, Figures 4–6 illustrate how the scheme can be used to accommodate slices with different traffic characteristics and balance their performance. Here, the booked capacity share of slice 3, γ_3, is plotted along the abscissa, while the booked capacities of slices 1 and 2 are determined as $\gamma_2 = \frac{1-\gamma_3}{3}$, $\gamma_1 = 2\gamma_2$ for fully booked capacity (solid lines) and as $\gamma_2 = \frac{1.1-\gamma_3}{3}$, $\gamma_1 = 2\gamma_2$ for 10% overbooking (dashed lines). Figures 4–6 show, respectively, the blocking probabilities, average user data rates the average user session duration as functions of γ_3. As one can see from Figure 4, by varying γ_s it is possible to protect some slices from the others to a different extent as well as to bring the blocking probability in each slice to the base-station average level. Note that the overall blocking probability is, again, slightly higher on average in the case of overbooking.

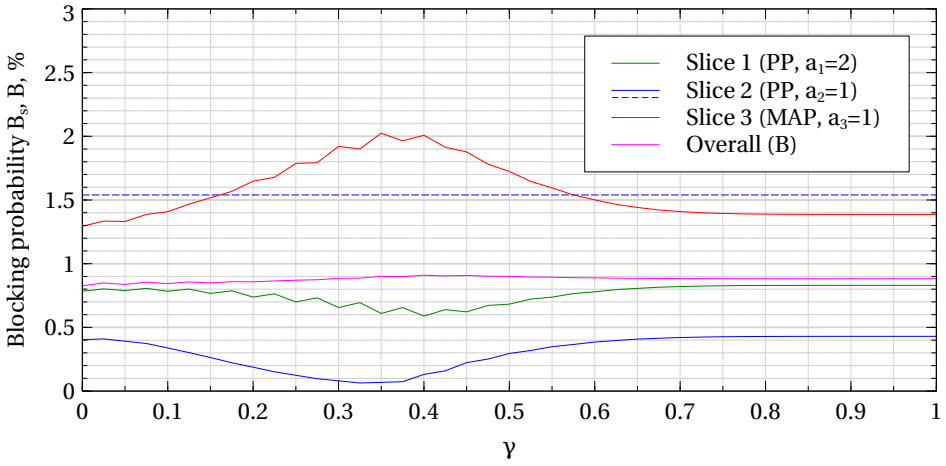

Figure 1. The blocking probabilities vs. the booked capacity shares $\gamma_s = \gamma$, $s \in \mathcal{S}$ (solid lines—slicing, dashed lines—complete partitioning (CP)). The blocking probabilities under CP for slices 1 and 3 are not plotted due to high values and equal, respectively, 7.5 % and 16.9 %. Note that the complete sharing (CS) values coincide with that under slicing for $\gamma = 1$ and are not shown explicitly.

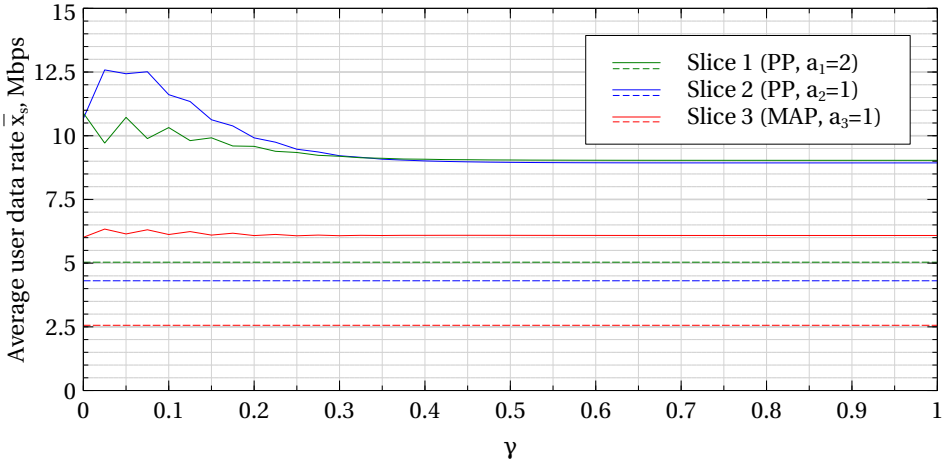

Figure 2. The average user data rates vs. the booked capacity shares $\gamma_s = \gamma$, $s \in \mathcal{S}$ (solid lines—slicing, dashed lines—CP). Note that the CS values coincide with that under slicing for $\gamma = 1$ and are not shown explicitly.

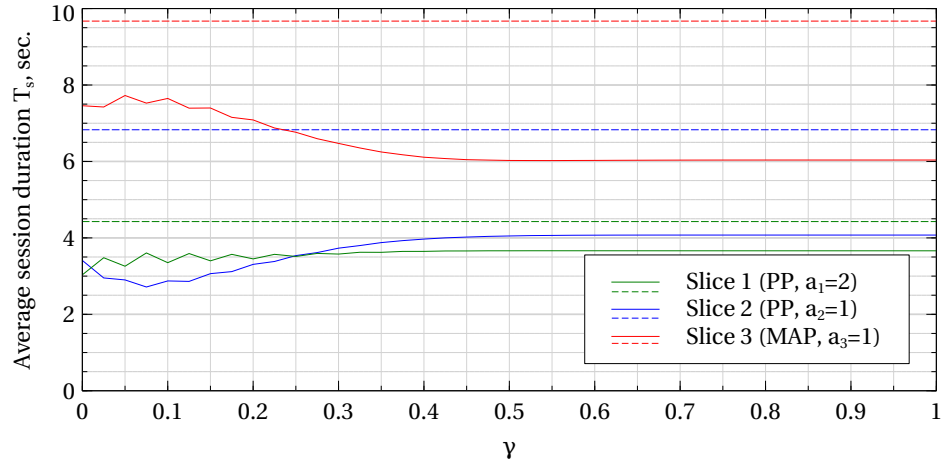

Figure 3. The average session duration vs. the booked capacity shares $\gamma_s = \gamma$, $s \in \mathcal{S}$ (solid lines—slicing, dashed lines—CP). Note that the CS values coincide with that under slicing for $\gamma = 1$ and are not shown explicitly.

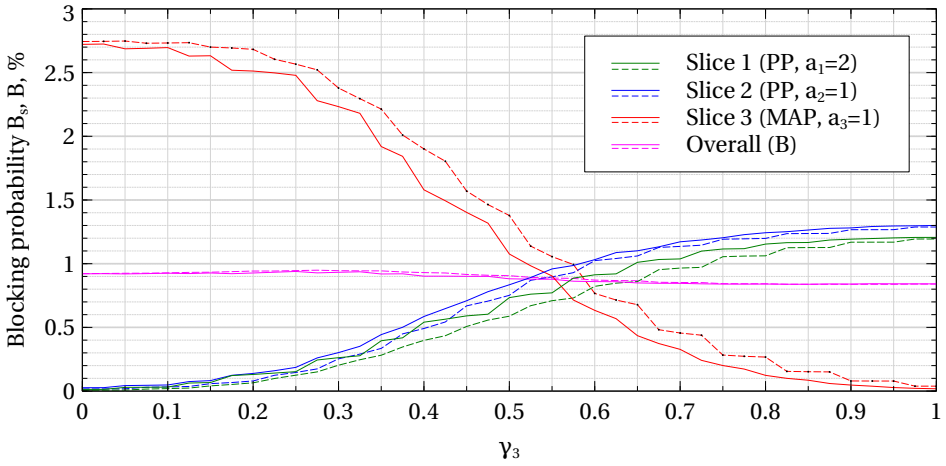

Figure 4. The blocking probabilities vs. γ_3; $\gamma_2 = \frac{\alpha-\gamma_3}{3}$, $\gamma_1 = 2\gamma_2$ (solid lines—$\alpha = 1$, dashed lines—$\alpha = 1.1$).

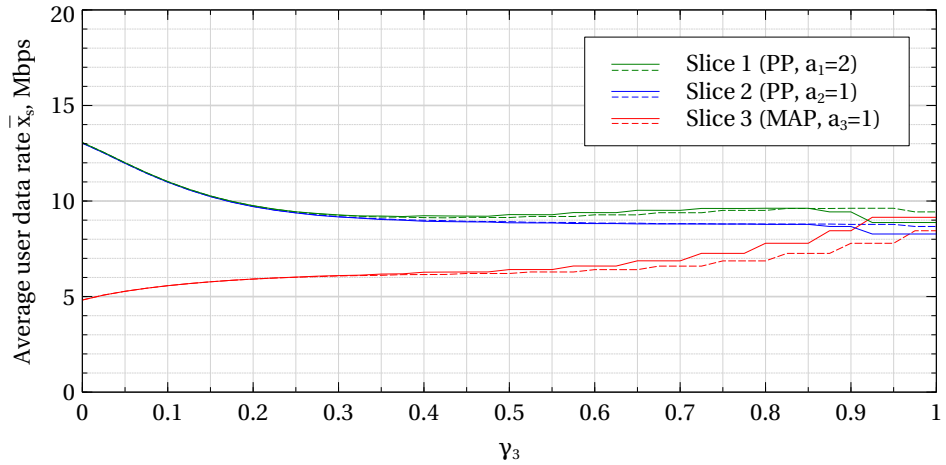

Figure 5. The average data rates $\bar{x}_s$ vs. γ_3; $\gamma_2 = \frac{\alpha-\gamma_3}{3}$, $\gamma_1 = 2\gamma_2$ (solid lines—$\alpha = 1$, dashed lines—$\alpha = 1.1$).

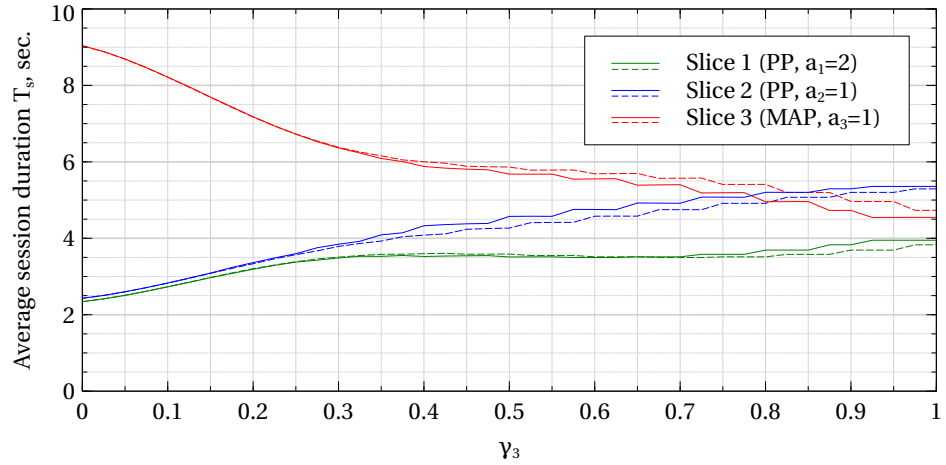

Figure 6. The average session duration T_s vs. γ_3; $\gamma_2 = \frac{\alpha-\gamma_3}{3}$, $\gamma_1 = 2\gamma_2$ (solid lines—$\alpha = 1$, dashed lines—$\alpha = 1.1$).

6. Discussion

The paper addresses the problem of inter-slice resource allocation in a shared capacity infrastructure among slices with different QoS and SLA requirements. A flexible and customizable resource slicing scheme, which leads to efficient resource usage, fairness and performance isolation of slices, is proposed along with an analytical CTMC model for its performance evaluation.

We assume each slice homogeneous in terms of traffic characteristics and QoS requirements and assign to it three parameters—non-zero minimum and maximum data rates per user and the booked number of users. While the former two parameters reflect the QoS requirements, the latter corresponds to the number of slice users for whom the capacity is booked, although not reserved in the strict sense, meaning that it may still be allocated to another slice if not used up. As long as the number of users in each slice remains within the booked threshold, all users are treated equally, which implies FCFS admission and fair (e.g., max–min fair) resource allocation. However, if the number of users in some slice goes above the threshold, the slice gets penalized—first, by way of resource allocation (it gets "squeezed") and then, if capacity is insufficient, by way of admission control through assignment of a lower preemptive priority. Thus, slice performance isolation is provided to slices as long as their number of users remains below a preset threshold, yet a larger number of users may receive service if the overall capacity and workload in other slices permit.

The slicing policy is formulated as a combination of a convex programming problem for resource allocation and an admission control policy with resource preemption. It is formalized in a way to allow for session-level modeling. We make use of the Markov process theory and develop a multi-class service system with state-dependent preemptive priorities, in which job service rates are found as the solution to the optimization problem. The MAP is used for one class of jobs to represent a slice which workload is less regular than the one represented by a Poisson process. We apply matrix-analytic methods to obtain the stationary distribution of the CTMC representing the system and use it to derive expressions for performance measures.

The numerical results provided in Section 5 give an insight into the performance of the proposed slicing policy, namely in comparison with CS and CP. Also, an algorithm for solving the optimization problem based upon the Gradient Projection Method is suggested. The numerical results show the capability of the proposed slicing scheme to accommodate heterogeneous traffic and provide performance isolation of slices while maintaining resource utilization at the CS level, thus clearly outperforming CP. The scheme is highly flexible and customizable, and further research is hence needed in order to provide a clear guidance as to its parametrization and the choice of utility/weight functions. Moreover, future work will be aimed at extending the slicing policy to allow for slice prioritization and leaving the admission control to the discretion of slice tenants.

Author Contributions: Conceptualization, L.M.C., K.S., Y.G. and N.Y.; methodology, N.Y.; validation, Y.G. and L.M.C.; formal analysis, N.Y.; writing—original draft preparation, N.Y.; writing—review and editing, L.M.C. and N.Y.; visualization, N.Y.; supervision, Y.G. and K.S.; project administration, K.S.; funding acquisition, K.S. and Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: The publication has been prepared with the support of the "RUDN University Program 5-100" (recipient K.S.). This work was supported in part by RFBR under Project 19-07-00933 and Project 20-07-01064 (Y.G.).

Acknowledgments: The authors are sincerely grateful to the reviewers for their helpful comments, which have significantly improved the quality of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. ETSI TS 123 501 V15.2.0 (06/2018) 5G; System Architecture for the 5G System (3GPP TS 23.501 version 15.2.0 Release 15). Available online: http://www.etsi.org/deliver/etsi_ts/123500_123599/123501/15.02.00_60/ts_123501v150200p.pdf (accessed on 9 April 2020).
2. ITU-T Rec. Y.3101 (01/2018) Requirements of the IMT-2020 Network. Available online: http://www.itu.int/rec/T-REC-Y.3101-201801-I/en (accessed on 9 April 2020).

3. ITU-T Rec. Y.3112 (12/2018) Framework for the Support of Network Slicing in the IMT-2020 Network. Available online: http://www.itu.int/rec/T-REC-Y.3112-201812-I/en (accessed on 9 April 2020).
4. GSM Association Official Document NG.116 (05/2019) Generic Netw. Slice Template, Version 1.0. Available Online: http://www.gsma.com/newsroom/wp-Content/uploads//NG.116 (accessed on 9 April 2020).
5. Vassilaras, S.; Gkatzikis, L.; Liakopoulos, N.; Stiakogiannakis, I.; Qi, M.; Shi, L.; Liu, L.; Debbah, M.; Paschos, G. The Algorithmic Aspects of Network Slicing. *IEEE Commun. Mag.* **2017**, *55*, 112–119. [CrossRef]
6. Guan, W.; Wen, X.; Wang, L.; Lu, Z.; Shen, Y. A Service-Oriented Deployment Policy of End-to-End Network Slicing Based on Complex Network Theory. *IEEE Access* **2018**. [CrossRef]
7. Zhang, N.; Liu, Y.-F.; Farmanbar, H.; Chang, T.-H.; Hong, M.; Luo, Z.-Q. Network Slicing for Service-Oriented Networks Under Resource Constraints. *IEEE J. Select. Areas Commun.* **2017**. [CrossRef]
8. Ksentini, A.; Nikaein, N. Toward Enforcing Network Slicing on RAN: Flexibility and Resources Abstraction. *IEEE Commun. Mag.* **2017**, *55*, 102–108. [CrossRef]
9. Kokku, R.; Mahindra, R.; Zhang, H.; Rangarajan, S. CellSlice: Cellular wireless resource slicing for active RAN sharing. In Proceedings of the 5th International Conference on Communication Systems and Networks, COMSNETS, Bangalore, India, 7–10 January 2013. [CrossRef]
10. Popovski, P.; Trillingsgaard, K.; Simeone, O.; Durisi, G. 5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View. *IEEE Access* **2018**. [CrossRef]
11. García-Morales, J.; Lucas-Estañ, M.C.; Gozalvez, J. Latency-Sensitive 5G RAN Slicing for Industry 4.0. *IEEE Access* **2019**. [CrossRef]
12. Akgül, Ö.; Malanchini, I.; Capone, A. Dynamic Resource Trading in Sliced Mobile Networks. *IEEE Trans. Netw. Service Manag.* **2019**. [CrossRef]
13. Lieto, A.; Malanchini, I.; Capone, A. Enabling Dynamic Resource Sharing for Slice Customization in 5G Networks. *GLOCOM* **2018**. [CrossRef]
14. Tun, Y.K.; Tran, N.H.; Ngo, D.T.; Pandey, S.R.; Han, Z.; Hong, C.S. Wireless Network Slicing: Generalized Kelly Mechanism Based Resource Allocation. *IEEE J. Select. Areas Commun.* **2019**, *37*, 1794–1807. [CrossRef]
15. Marabissi, D.; Fantacci, R. Highly Flexible RAN Slicing Approach to ManageIsolation, Priority, Efficiency. *IEEE Access* **2019**, *7*. [CrossRef]
16. Papa, A.; Kluegel, M.; Goratti, L.; Rasheed, T.; Kellerer, W. Optimizing Dynamic RAN Slicing in Programmable 5G Networks. In Proceedings of the IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019. [CrossRef]
17. Vilà, I.; Sallent, O.; Umbert, A.; Pérez-Romero, J. An Analytical Model for Multi-Tenant Radio Access Networks Supporting Guaranteed Bit Rate Services. *IEEE Access* **2019**, *7*, 57651–57662. [CrossRef]
18. Vo, P.L.; Nguyen, M.N.H.; Le, T.A.; Tran, N.H. Slicing the Edge: Resource Allocation for RAN Network Slicing. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 970–973. [CrossRef]
19. Parsaeefard, S.; Dawadi, R.; Derakhshani, M.; Le-Ngoc, T. Joint User-Association and Resource-Allocation in Virtualized Wireless Networks. *IEEE Access* **2015**. [CrossRef]
20. Caballero, P.; Banchs, A.; de Veciana, G.; Costa-Pérez, X. Networkslicing games: Enabling customization in multi-tenant networks. In Proceedings of the IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017. [CrossRef]
21. Caballero, P.; Banchs, A.; de Veciana, G.; Costa-Pérez, X.; Azcorra, A. Network Slicing for Guaranteed Rate Services: Admission Control and Resource Allocation Games. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 6419–6432. [CrossRef]
22. D'Oro, S.; Restuccia, F.; Melodia, T.; Palazzo, S. Low-Complexity Distributed Radio Access Network Slicing: Algorithms and Experimental Results. *IEEE/ACM Trans. Netw.* **2018**. [CrossRef]
23. Sun, Y.; Qin, S.; Feng, G.; Zhang, L.; Imran, M. Service Provisioning Framework for RAN Slicing: User Admissibility, Slice Association and Bandwidth Allocation. *IEEE Trans. Mob. Comput.* **2020**. [CrossRef]
24. Zhao, G.; Qin, S.; Feng, G.; Sun, Y. Network Slice Selection in Softwarization-Based Mobile Networks. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*. [CrossRef]
25. Yan, M.; Feng, G.; Zhou, J.H.; Sun, Y.; Liang, Y.-C. Intelligent Resource Scheduling for 5G Radio Access Network Slicing. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7691–7703. [CrossRef]
26. Zhou, G.; Zhao, L.; Liang, K.; Zheng, G.; Hanzo, L. Utility Analysis of Radio Access Network Slicing. *IEEE Trans. Veh. Technol.* **2019**. [CrossRef]

27. Lee, Y.L.; Loo, J.; Chuah, T.; Wang, L.-C. Dynamic Network Slicing for Multitenant Heterogeneous Cloud Radio Access Networks. *IEEE Trans. Wirel. Commun.* **2018**. [CrossRef]
28. Khatibi, S.; Correia, L.M. Modelling virtual radio resource management in full heterogeneous networks. *EURASIP J. Wirel. Commun. Netw.* **2017**, *73*. [CrossRef]
29. Narmanlioglu, O.; Zeydan, E.; Arslan, S.S. Service-Aware Multi-Resource Allocation in Software-Defined Next Generation Cellular Networks. *IEEE Access* **2018**. [CrossRef]
30. Fossati, F.; Moretti, S.; Perny, P.; Secci, S. Multi-Resource Allocation for Network Slicing. *IEEE/ACM Trans. Netw.* **2020**, *28*, 1311–1324. [CrossRef]
31. Salvat, J.X.; Zanzi, L.; Garcia-Saavedra, A.; Sciancalepore, V.; Costa-Perez, X. Overbooking Network Slices through Yield-Driven End-to-End Orchestration. In Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '18), New York, NY, USA, 4–7 December 2018; pp. 353–365. [CrossRef]
32. Bega, D.; Gramaglia, M.; Banchs, A.; Sciancalepore, V.; Costa-Pérez, X. A Machine Learning Approach to 5G Infrastructure Market Optimization. *IEEE Trans. Mob. Comput.* **2019**, *19*, 498–512. [CrossRef]
33. Sallent, O.; Pérez-Romero, J.; Ferrus, R.; Agustí, R. On Radio Access Network Slicing from a Radio Resource Management Perspective. *IEEE Wirel. Commun.* **2017**, *24*, 166–174. [CrossRef]
34. Vincenzi, M.; Lopez-Aguilera, E.; Garcia-Villegas, E. Maximizing Infrastructure Providers' Revenue Through Network Slicing in 5G. *IEEE Access* **2019**, *7*. [CrossRef]
35. Lucantoni, D.M. New Results on the Single Server Queue with a Batch Markovian Arrival Process. *Commun. Stat. Stochastic Models* **1991**, *7*, 1–46. [CrossRef]
36. Naumov, V.A. Markovskiye modeli potokov trebovaniy [Markovian models of jobs streams]. In *Sistemy Massovogo Obsluzhivaniya i Informatika*; RUDN Publisher: Moscow, Russia, 1987; pp. 67–73. (In Russian)
37. Buchholz, P. An EM-Algorithm for MAP Fitting from Real Traffic Data. In *Computer Performance Evaluation. Modelling Techniques and Tools. TOOLS 2003. Lecture Notes in Computer Science*; Kemper, P., Sanders, W.H., Eds.; Springer: Berlin/Heidelberg, Germany, 2003.
38. Vishnevskiy, V.M.; Samouylov, K.E.; Yarkina, N.V. Mathematical Model of LTE Cells with Machine-Type and Broadband Communications. *Autom. Remote Control* **2020**, *81*, 622–636. [CrossRef]
39. Rosen, J.B. The gradient projection method for nonlinear programming. *J. Soc. Ind. Appl. Math.* **1960**, *8*, 181–217. [CrossRef]
40. Rouzbehani, B.; Correia, L.M.; Caeiro, L. A Service-Oriented Approach for Radio Resource Management in Virtual RANs. *Hindawi Wirel. Commun. Mob. Comput.* **2018**. [CrossRef]
41. Kelly, F. Charging and rate control for elastic traffic. *Eur. Trans. Telecommun.* **1997**, *8*, 33–37. [CrossRef]
42. Bertsekas, D.; Gallager, R. *Data Networks*, 2nd ed.; Prentice Hall: Englewood Cliffs, NJ, USA, 1992; ISBN 978-01-3196-825-7.
43. Yashkov, S.; Yashkova, A. Processor sharing: A survey of the mathematical theory. *Autom. Remote Control* **2007**, *68*, 1662–1731. [CrossRef]
44. Naumov, V.A.; Gaidamaka, Y.V.; Samouylov, K.E. On Truncation of the Matrix-Geometric Stationary Distributions. *Mathematics* **2019**, *7*, 798. [CrossRef]