Slice Management in Radio Access Network via Iterative Adaptation

Behnam Khodapanah[*], Ahmad Awada[†], Ingo Viering[‡], Andre Noll Barreto[§], Meryem Simsek[¶], Gerhard Fettweis[*]
[*]Vodafone Chair Mobile Communications Systems, Technische Universität Dresden, Germany
Email:{behnam.khodapanah, gerhard.fettweis}@tu-dresden.de
[†]Nokia Bell Labs, Munich, Germany; Email: ahmad.awada@nokia-bell-labs.com
[‡]Nomor Research GmbH, Munich, Germany; Email: viering@nomor.de
[§]Barkhausen Institut, Dresden, Germany; Email: andre.nollbarreto@barkhauseninstitut.org
[¶]International Computer Science Institute, Berkeley, USA; Email: simsek@icsi.berkeley.edu

***Abstract*—In the context of 5G systems, the emergence of various use cases with diverse requirements has attracted great attention to network slicing. In a single physical network, several instances of logical end-to-end networks, i.e. slices, will be instantiated to fulfill these requirements. To this end, slices should share the resources of the physical network, which consist of Core Network (CN) and Radio Access Network (RAN) resources. Herein, we focus on the Radio Resource Management (RRM) in the context of network slicing. To maximize the pooling gains, dynamic resource sharing is preferred over static sharing. However, dynamic resource sharing can lead to undesirable inter-slice influences, in particular, the contention on radio resources. In this article, we show that, although the radio resources are dynamically shared among the users of different slices, proper slice management can realize slice protection. This is achieved by adjusting the fraction of radio resources allocated to the different slices by the Packet Scheduler (PS) and by limiting the number of users admitted to the network via Admission Control (AC). We propose an iterative algorithm to optimize the parameters of PS and AC in order to ensure that the service level agreements are satisfied. Extensive system-level simulations have shown that a central entity that tunes these control parameters can greatly increase the network's performance.**

***Index Terms*—Network Slicing, Radio Resource Management, Slice Orchestration, 5G, Iterative Adaptation**

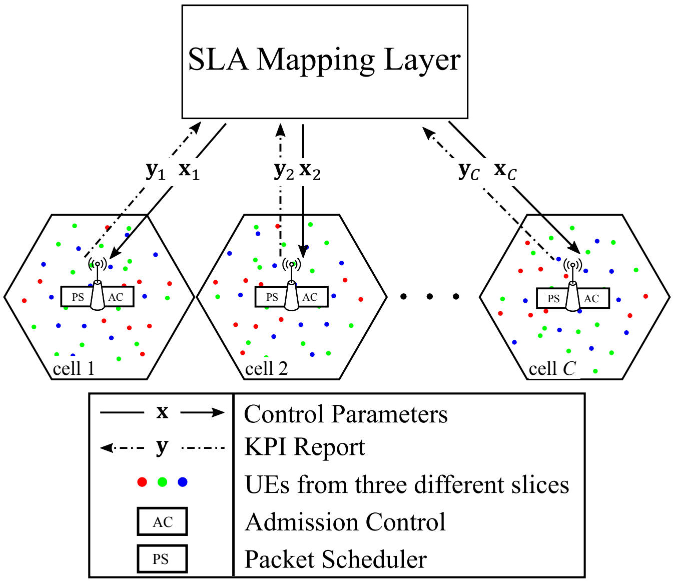

SLA Mapping Layer
$\mathbf{y}_1$ $\mathbf{x}_1$
$\mathbf{y}_2$ $\mathbf{x}_2$
$\mathbf{y}_C$ $\mathbf{x}_C$
PS
AC
cell 1
cell 2
cell C
$\mathbf{x}$ Control Parameters
$\mathbf{y}$ KPI Report
UEs from three different slices
AC Admission Control
PS Packet Scheduler

Fig. 1: SLA mapping layer as a slice orchestrator.

I. Introduction

It is anticipated that the fifth generation (5G) networks shall support a multitude of heterogeneous services, namely enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC) and massive Machine Type Communications (mMTC) [1]. Since the requirements of these services vastly differ, legacy networks with a monolithic architecture can hardly accommodate them simultaneously. On the other hand, deploying multiple service-specific networks is not an efficient and financially plausible solution. Network slicing offers a flexible and scalable solution for accommodating diverse services in a single physical network. This solution allows several logical end-to-end networks, i.e. slices, to coexist and efficiently share the physical infrastructure, which brings about massive multiplexing gains and increases resource and energy efficiency [2]. Furthermore, since the slices are separate networks, although virtual, they act as independent networks.

In a sliced network, the tenants of the network specify their service requirements in terms of Key Performance Indicators (KPIs) within a Service Level Agreement (SLA) and the network operator should instantiate the appropriate network slice to meet these SLAs [3]. At the same time, since the slices share the same physical infrastructure, they must be protected from each other such that dynamics of one slice do not adversely affect other slices [4].

Considering that slices are end-to-end networks, slicing spans both the Core Network (CN) and Radio Access Network (RAN) [3]. Slicing the CN has been studied extensively in fields like Software-Defined Network (SDN) and Network Function Virtualization (NFV), where efficient architecture design, instantiation, deployment, and maintenance of the CN functions have been investigated. In RAN, however, slicing deals with the efficient sharing of the radio resources, i.e. time, frequency and space. Contrary to the CN resources, the unpredictability of the wireless medium makes the slicing in RAN a challenging topic. In particular, Radio Resource Management (RRM) is a crucial mechanism for ensuring the fulfillment of all SLAs. That is, RRM should simultaneously make sure that the resources are shared dynamically between the slices, while the slices are protected from negative influences of each other.

The objectives of the RRM in a sliced network have been addressed separately in legacy mobile networks. Fulfilling the requirements of the users via implementing the Quality-of-Service (QoS) Class Identifier (QCI) mechanisms has been proposed in 3GPP Long-Term Evolution (LTE) systems [5]. Based on the requirements of each user, an appropriate QCI will be assigned to it to guarantee certain service with regards to throughput, delay, etc. The fundamental difference between QoS-aware RRM and slice-aware RRM is that not only the QoS should be guaranteed for all of the users belonging to a slice, the KPIs that describe their collective performance should be above some target, which is defined in the SLA. As for sharing the existing physical network, network virtualization has been studied in the context of Mobile Virtual Network Operators (MVNOs) [6]. In these networks, the resources are usually shared via a fixed sharing agreement, which ensures the isolation of the networks from each other, but inhibits the multiplexing gains. Although the dynamic sharing of the resources has been studied in [7], the impact of negative inter-network influences have not been analyzed yet.

Recently, authors in [8], [9] and [10] have proposed a general framework for resource management in a sliced network with an auction-based model. Although the approach of auctioning each (resource) block to maximize the total revenue of the operator can be easily applied in the CN, the application to RRM is not straightforward. Because of the random and dynamic nature of the wireless environments, the capacities of the radio resources are varying and difficult to abstract at the system level. Furthermore, such approaches require detailed penalties for not fulfilling the whole or parts of the SLA, which might be hard to define beforehand.

In this work, we propose an entity called SLA mapping layer that monitors the network in terms of KPIs and verifies the SLA fulfillment of the slices. If this entity detects that certain KPIs are below their targets, it tries to fine tune the control parameters of the Packet Scheduler (PS) and Admission Control (AC) such that SLA fulfillment is achieved for all of the slices. This orchestration task of the SLA mapping layer is illustrated in Fig. 1. In our previous work [11] we have demonstrated that such an entity can be helpful for minimizing the deviations of KPIs from the target SLAs in a static environment. However, unlike [11], we are considering a more realistic system, where the environment is dynamic and the slices have very diverse requirements.

This article is structured as follows. In Section II, we describe the system model of a sliced network with the presence of slices with different requirements. Next, in Section III, we introduce an algorithm for the SLA mapping layer, which orchestrates the RRM in an iterative manner such that all of the SLAs of the slices are fulfilled. The simulation results can be found in Section IV where the performance of the algorithm in different schemes is studied. Finally, in Section V we conclude the paper.

II. System Model

Consider a mobile cellular network with $c = 1, 2, ..., C$ cells and let $\mathbb{S}$ be the set of all slices. Total number of slices in the network is denoted as S, which is equal to the cardinal number of set $\mathbb{S}$, i.e. $|\mathbb{S}| = S$. The users belonging to these slices arrive in the network at a random time and location and intend to download a file and leave the network (traffic model similar to [12]). Section II-A describes the random processes behind this procedure. In Section II-B, we elaborate on the slice types. These slice types serve as a template for the slice instantiation. Finally, in Section II-C and II-D, the inner workings of the PS and AC algorithms are described. Since each of these slices has different requirements, the PS and AC are responsible to assure that these diverse demands are fulfilled throughout the network.

A. Spatial and Temporal User Distribution

We assume that the arrival process of the users of slice s is a Poisson-distributed random variable with an arrival rate of λ_s. Furthermore, the position of the users is also a two dimensional (2D) random variable. In this article, we study the impact of two different spatial distributions on the performance of the network. For the first distribution, we assume that the users are distributed uniformly across the network. In the second distribution, we simulate a spatial hot-spot, using a 2D Gaussian distribution [13]. The mean vector $[\mu_{\text{hor}}, \mu_{\text{ver}}]$ of this distribution is the center of the hot-spot and the variance vector $[\sigma^2_{\text{hor}}, \sigma^2_{\text{ver}}]$ represents, how concentrated the users are, in horizontal and vertical axes, respectively. For simplicity, we assume that $\mu_{\text{hor}} = \mu_{\text{ver}} = \mu$ and $\sigma^2_{\text{hor}} = \sigma^2_{\text{ver}} = \sigma^2$ so that the 2D Gaussian distribution is symmetric along the horizontal and vertical axis. Note that we truncate the 2D Gaussian distribution to be only limited to the network space. Uniform distribution can be considered as a special case of the truncated 2D Gaussian distribution with $\sigma^2 = \infty$. To simplify the illustration of results, we define the concentration factor as $1/\sigma$. Now, as the concentration factor approaches 0, the users are more uniformly distributed.

B. Slices Types with Diverse Requirements

To simulate the slices with different requirements, we have defined three slice types. We assume that in general, several instances of these slice types might be present in the network. One could view these slice types to be slice templates to be instantiated every time a new slice is added to the network.

- *Best Effort (BE)*:
  The users belonging to this slice do not have any rigid requirements on their instantaneous throughput, which is a function of the users' channel conditions and the PS decisions. Applications like web browsing can be considered as an example of this slice. However, the long-term average of the users' throughputs (T_{BE}) and the fifth-percentile (F_{BE}) KPIs must be above

the targets that have been declared in the SLA, i.e., $\bar{T}_{\text{BE}}$ and $\bar{F}_{\text{BE}}$, respectively. Moreover, to implement the contention control in the network, we assume that the users of BE will be dropped from the network if they linger more than a time threshold, namely θ_D. This mechanism ensures that even in the very congested conditions, the number of users will not grow indefinitely. Although this mechanism ensures network stability, the BE slice does not wish to have its users dropped frequently, therefore the dropping rate (D_{BE}) should be below a target defined in SLA, i.e., $\bar{D}_{\text{BE}}$. For convenience, we can reformulate the KPI as $1 - D_{\text{BE}}$ and wish that this KPI would be above the $1 - \bar{D}_{\text{BE}}$. We further assume that the AC admits all of the BE users.

- *Constant Bit-Rate (CBR)*:
  The admitted users of the CBR slice are guaranteed to have a constant throughput; if the AC has admitted an CBR user, regardless of the user's channel conditions, the constant throughput should be granted. Voice-over-IP (VoIP) can be considered as a service which has similar requirement. Since the throughput is constant for all the users, the only KPI that will be associated with this slice is the admission rate (A_{CBR}) which has to be above the target in the SLA, i.e., $\bar{A}_{\text{CBR}}$.
- *Minimum Bit-Rate (MBR)*:
  Similar to BE users, the MBR users' throughput is determined by the channel conditions and the PS decisions. On the other hand, similar to CBR users, a minimum bit-rate has to be guaranteed for the MBR users. Moreover, the AC controls the number of admitted MBR users. Applications such as video streaming can be examples of this service since the video codecs require a minimum bit-rate to be able to stream with the lowest quality. The average throughput of MBR users (T_{MBR}) and the admission rate (A_{MBR}) are the considered KPIs for this slice. These KPIs should be above the targets in the SLA, i.e., $\bar{T}_{\text{MBR}}$ and $\bar{A}_{\text{MBR}}$. Note that for this slice type we don't consider the fifth-percentile throughput as a KPI, because a minimum instantaneous bit-rate is guaranteed for all the users.

We define $\mathbb{S}_{\text{BE}}$, $\mathbb{S}_{\text{CBR}}$, and $\mathbb{S}_{\text{MBR}}$ to be the sets of all BE, CBR and MBR slices, respectively.

C. Packet Scheduler

To model the scheduling process in presence of different users of different slices, we first model the users' throughput. Based on Shannon's capacity formula, the throughput of user $i = 1, 2, \cdots, N_{s,c}$ from slice s in cell c is defined as

$$T^i_{s,c} = r^i_{s,c} \cdot B \cdot \log_2(1 + \gamma^i_{s,c}), \tag{1}$$

where $r^i_{s,c}$ is the resource share of the user i in slice s, $\gamma^i_{s,c}$ is the average Signal-to-Interference-plus-Noise-Ratio (SINR) of user i and B is the total bandwidth.

For the CBR users, the throughput is constant and guaranteed and given in the SLA, i.e, $\bar{G}_s$. Consequently, the amount of resource share needed to fulfill the throughput for every user belonging to slice s in $\mathbb{S}_{\text{CBR}}$ is given by

$$r^i_{s,c} = \frac{\bar{G}_s}{B \cdot \log_2(1 + \gamma^i_{s,c})}. \tag{2}$$

The admitted CBR users will take their share of resources first and collectively require

$$R_{\text{CBR},c} = \sum_{s \in \mathbb{S}_{\text{CBR}}} \sum_{i=1}^{N_{s,c}} r^i_{s,c}, \tag{3}$$

and the rest of the resources, i.e., $1 - R_{\text{CBR},c}$, will be shared between the MBR and BE users.

To model the scheduling of MBR and BE users, we propose a resource-fair scheduler with prioritization. A conventional resource-fair scheduler distributes the same amount of resources to each user. To enable prioritization of different slices, a weight vector is defined as $\mathbf{w}_{*,c} = [w_{1,c}, w_{2,c}, \cdots, w_{|\mathbb{S}_{\text{BE}} \cup \mathbb{S}_{\text{MBR}}|,c}]$ for cell c, where $\mathbb{S}_{\text{BE}} \cup \mathbb{S}_{\text{MBR}}$ constitutes all the BE and MBR slices. The resource share of user $i = 1, 2, \cdots, N_{s,c}$ belonging to slice s in $\mathbb{S}_{\text{BE}} \cup \mathbb{S}_{\text{MBR}}$ and in cell c is defined as

$$r^i_{s,c}(\mathbf{w}_{*,c}) = \frac{w_{s,c} \cdot (1 - R_{\text{CBR},c})}{\sum\limits_{s' \in \mathbb{S}_{\text{BE}}} N_{s',c} \cdot w_{s',c} + \sum\limits_{s'' \in \mathbb{S}_{\text{MBR}}} N_{s'',c} \cdot w_{s'',c}}. \tag{4}$$

If we only use Eq. (4) for the MBR and BE users, there might be some MBR users that do not get enough resources to achieve their minimum throughput. To simultaneously use Eq. (4) and fulfill the MBR requirement, we propose an iterative scheduling. First, the resources are shared based on Eq. (4). If any of the MBR users has lower throughput than its minimum bit-rate, similar to Eq. (2), the minimum resources will be determined and assigned to them. let $\breve{N}_{s'',c}$ be the number of users that have received this special treatment. The collective resource consumption of the users of these slices is

$$\breve{R}_{\text{MBR}} = \sum_{s'' \in \mathbb{S}_{\text{MBR}}} \sum_{i=1}^{\breve{N}_{s'',c}} r^i_{s'',c}. \tag{5}$$

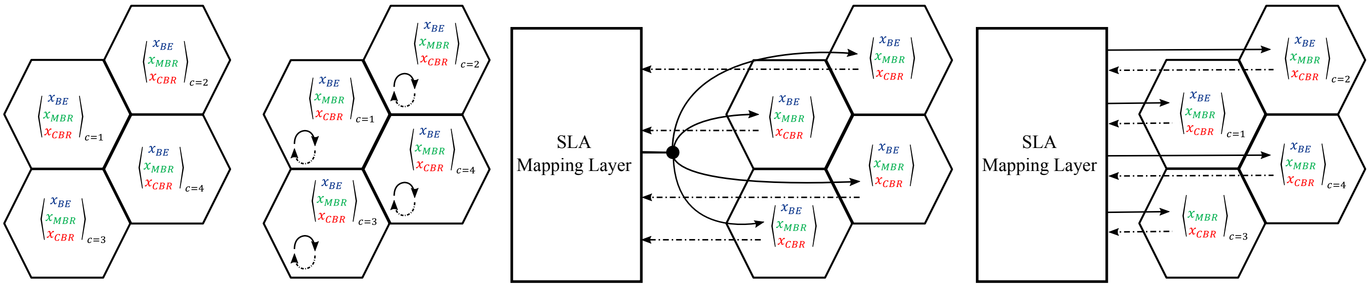
After this special treatment of some MBR users, the resource share of users of slices s in $\mathbb{S}_{\text{BE}} \cup \mathbb{S}_{\text{MBR}}$ and in cell c is defined as

$$r^i_{s,c}(\mathbf{w}_{*,c}) = \frac{w_{s,c} \cdot (1 - R_{\text{CBR},c} - \breve{R}_{\text{MBR}})}{\sum\limits_{s' \in \mathbb{S}_{\text{BE}}} N_{s',c} \cdot w_{s',c} + \sum\limits_{s'' \in \mathbb{S}_{\text{MBR}}} \hat{N}_{s'',c} \cdot w_{s'',c}}, \tag{6}$$

where $\hat{N}_{s'',c} = N_{s'',c} - \breve{N}_{s'',c}$ is the number of MBR users of slice s'' that have achieved the MBR only with the resources assigned to them by the PS. Note that after each iteration of the scheduler (using Eq. (6)), there might be some MBR users whose resource share is not sufficient. Therefore, the iteration repeats until all the MBR users are satisfied.

TABLE I: Different adaptation schemes.

Scheme I	Scheme II	Scheme III	Scheme IV
No adaptation	Distributed adaptation	Central adaptation without cell specific control	Central adaptation with cell specific control
No violation vector $\mathbf{0}$	Local violation vector $\mathbf{v}_c^t$	Global violation vector $\mathbf{v}^t$	Global and local violation vector $\mathbf{v}_c^t \odot \mathbf{v}^t$



x_{BE}: Control parameters related to BE slice x_{MBR}: Control parameters related to MBR slice x_{CBR}: Control parameters related to CBR slice

D. Admission Control

The role of AC in the network is to regulate the incoming traffic. The AC blocks some users so that the number of admitted users are limited. Tenants want the admission rate to be as high as possible. However, by admitting more users, the other KPIs of the network will be affected because the number of active users will increase. This mechanism is especially crucial in sliced networks since too many users from one slice might negatively impact the KPIs of the other slices. To implement an AC, we define resource thresholds. For all of the MBR and CBR slices that are in set $\mathbb{S}_{\text{CBR}} \cup \mathbb{S}_{\text{MBR}}$, the admission policy is

$$\begin{cases} \text{If } R_{s,c} \le th_{s,c} & \text{grant admission} \\ \text{If } R_{s,c} > th_{s,c} & \text{deny admission} \end{cases}, \qquad (7)$$

where $th_{s,c}$ is the resource threshold for slice s in cell c and $R_{s,c} = \sum_{i=1}^{N_{s,c}} \bar{G}_s/B \cdot \log_2(1+\gamma^i_{s,c})$ is the minimum resources that is required to satisfy the MBR or CBR slice.

III. SLA Mapping Layer

The objective of a slice-aware RRM system is to tune the control parameters of different slices in different cells, so that the KPIs of the network are in a state that do not violate any of the slice SLAs. We can define the relationship between the control parameters and KPIs as

$$\mathbf{y} = f(\mathbf{X}), \qquad (8)$$

where $\mathbf{y} = [A_{\text{CBR}}, 1 - D_{\text{BE}}, T_{\text{BE}}, F_{\text{BE}}, T_{\text{MBR}}, A_{\text{MBR}}]^T$ is a $K \times 1$ vector of all KPIs of all slices in the whole network and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_C]$ is a $X \times C$ matrix that contains all of the control parameters vectors from all of the cells. Control parameters vector $\mathbf{x}_c = [w_{\text{BE},c}, w_{\text{MBR},c}, th_{\text{CBR},c}, th_{\text{MBR},c}]^T$ is a $X \times 1$ vector containing the control parameters pertaining to PS and AC in cell c.

The goal is to adjust the control parameters so that the KPIs are near the target values defined in the SLA, i.e. $\bar{\mathbf{y}}$. However, functional form of $f(\cdot)$ is not available in closed-form even in systems with just one slice. For example, no closed form expression for minimum throughput or fifth-percentile throughput of BE slice is known. This is exacerbated by the presence of multiple slices, the KPIs for which are coupled. Consequently, optimizing the control parameters is really challenging and conventional optimization approaches such as gradient-descent do not apply here. In Section III-A, we circumvent this issue by developing an algorithm that requires only the sign of the partial derivative of $f(\cdot)$ with regards to the different control parameters. Section III-B introduces different schemes based on how we collect network reports.

A. Iterative Adaption Algorithm

To find the best control parameters in the cells defined in Eq. (8), we introduce an iterative adaptive algorithm. In every time interval of τ seconds, the SLA mapping layer adapts the control parameters of every cell. After each interval, the KPIs are measured and reported back to the SLA mapping layer. At interval t, the network-wide KPIs can be stated as

$$\mathbf{y}^t = f(\mathbf{X}^t = [\mathbf{x}_1^t, \mathbf{x}_2^t, \cdots, \mathbf{x}_C^t]). \qquad (9)$$

To be able to fine tune the control parameters for the next interval, i.e. $[\mathbf{x}_1^{t+1}, \mathbf{x}_2^{t+1}, \cdots, \mathbf{x}_C^{t+1}]$, we require a sensible approximation of the relationships between the X control parameters and K KPIs. This approximation can be represented in a $X \times K$ matrix defined as $\mathbf{J} = [j_{x,k}]$, where

$$j_{x,k} = \begin{cases} 0 & \text{if increase in } x \text{, does not affect KPI } k \\ +1 & \text{if increase in } x \text{, increases KPI } k \\ -1 & \text{if increase in } x \text{, decreases KPI } k \end{cases}. \qquad (10)$$

This matrix allows us to increase certain KPIs by increasing or decreasing the corresponding control parameters. This matrix

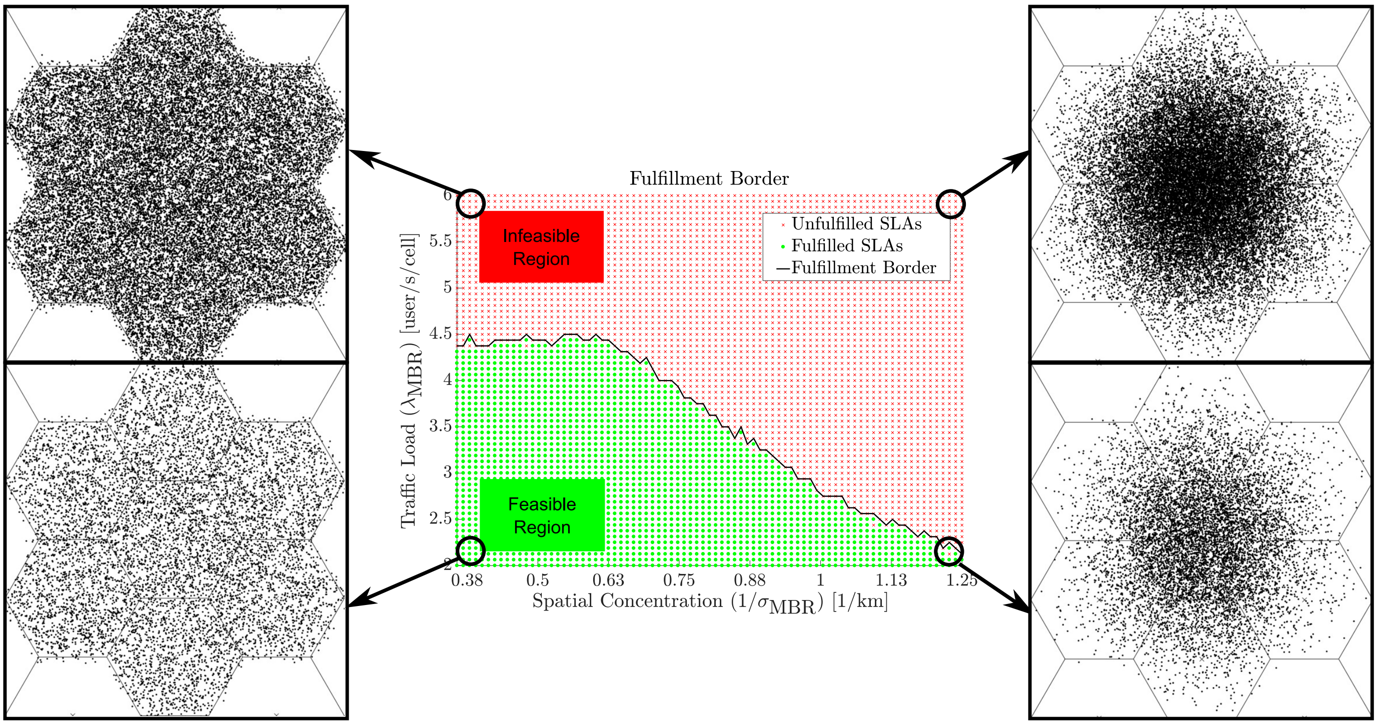


Fig. 2: Fulfillment border of an example scheme.

can be viewed as a coarse approximation of the Jacobian matrix of Eq. (8), where first-order derivative of all the KPIs with regards to all of the control parameters would be available. However, to analytically obtain the Jacobian, an accurate model of the network is needed, which is not necessarily available. Assuming that we have one instance of each slice type, one reasonable design for $\mathbf{J}$ matrix can be defined as

$$\mathbf{J} = \begin{array}{c c} \begin{array}{cccccc} A_{\text{CBR}} & 1-D_{\text{BE}} & T_{\text{BE}} & F_{\text{BE}} & T_{\text{MBR}} & A_{\text{MBR}} \end{array} & \\ \begin{bmatrix} 0 & +1 & +1 & +1 & -1 & -1 \\ 0 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & -1 & -1 & -1 \\ 0 & -1 & -1 & -1 & -1 & +1 \end{bmatrix} & \begin{array}{c} w_{\text{BE}} \\ w_{\text{MBR}} \\ th_{\text{CBR}} \\ th_{\text{MBR}} \end{array} \end{array} . \quad (11)$$

The single KPI of the CBR slice is the admission rate A_{CBR} and it will only increase if the AC threshold th_{CBR} is increased. The other control parameters do not affect this KPI since the CBR users’ resource share is guaranteed for the admitted users. The KPIs of the BE slice will only increase if the scheduler prioritizes them over the MBR users. This can be done by increasing w_{BE} or decreasing w_{MBR}. Additionally increasing th_{CBR} or th_{MBR} slices decreases the KPIs of the BE slice since more of these users will be admitted to the network. Hence, the overall load increases. Finally for the KPIs of the MBR slice, the throughput increases if the MBR users have more priority in the scheduler. Moreover, an increase in th_{CBR} or th_{MBR} will increase the number of users and cause congestion. The admission rate of the MBR slice increases if the scheduler prioritizes them more because they will have better throughput and will leave the network earlier and make room for the new users. Similarly, a decrease in th_{CBR} will influence the A_{MBR} positively. Finally, it is clear that an increase in th_{MBR} clearly positively affects A_{MBR}.

To determine which KPI needs increasing, we define a $K \times 1$ violation vector $\mathbf{v}^t = H(\bar{\mathbf{y}} - \mathbf{y}^t)$, where $H(\cdot)$ is the element-wise step function, i.e.,

$$v_k^t = H(\bar{y}_k - y_k^t) = \begin{cases} 1 & \text{if } y_k^t < \bar{y}_k \\ 0 & \text{if } y_k^t > \bar{y}_k \end{cases}, \quad (12)$$

where v_k^t, y_k^t and $\bar{y}_k$ are the kth KPI in the violation, tth iteration’s KPI and target KPI vectors.

Using the violation vector $\mathbf{v}^t$, we know which KPIs are not satisfied and with the relationship matrix $\mathbf{J}$, we know which control parameters should be changed. Therefore, the update rule is defined as

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \delta \mathbf{J} \mathbf{v}^t, \quad (13)$$

where δ is the step size for the control parameter update.

Note that the step function $H(\cdot)$ is used in Eq. (12) rather than the conventional Mean Square Error (MSE) metric, i.e., $(\bar{y}_k - y_k^t)^2$. The reason is that the KPIs have different units (e.g. Admission rate [%] and average throughput [Mbps]) and different scales (e.g. average throughput is usually much larger than fifth-percentile throughput). Consequently, to avoid implicitly weighting different KPIs, we only consider whether the KPI was violated or not.

B. Adaptation Schemes

So far we have assumed that the KPI reports $\mathbf{y}^t$ are collected over the whole network. However, we can define the local KPI reports in cell c as $\mathbf{y}_c^t$ and define $\mathbf{v}_c^t$ (c.f. Eq. (12)) as local violation vector. Based on this classification of the KPI reports, we can devise four different schemes and compare their performance:

- *Scheme I - No adaptation:*
  We do not change the initial control parameters in any of the cells. This scheme is for comparison only.

- *Scheme II - Distributed adaptation:*
  Within each cell, we use Eq. (13), where instead of $\mathbf{v}^t$, we use $\mathbf{v}_c^t$. This implies that the cells are unaware of the performance of the surrounding cells.

- *Scheme III - Centralized adaptation without cell-specific parameters:*
  The central entity, i.e., SLA mapping layer, collects reports from the whole network and uses Eq. (13) for all $c = 1, 2, \cdots, C$ with violation vector $\mathbf{v}^t$. In this scheme, we have the information about the whole network, but we do not have the freedom to tune each cell's control parameters individually.

- *Scheme IV - Centralized adaptation with cell-specific parameters:*
  Similar to Scheme III, the central entity collects reports from the whole network. However, Eq. (13) is utilized with $\mathbf{v}^t \odot \mathbf{v}_c^t$ as the violation vector, where $\odot$ is element-wise multiplication. With this scheme, we only change the control parameters if the respective global and local KPIs are violated.

For clarity, Table I summarize the aforementioned four schemes.

IV. Performance Analysis

A. Simulation Scenario

To evaluate the proposed schemes and algorithm, we first describe the simulation setup. We assume that we have three slices, one from each slice type, i.e. BE, CBR and MBR. Associated with these slices, the default load, user distribution, and control parameters are given in Table II. These default values are chosen so that the network can fulfill the SLAs without the need to update the control parameters.

B. Evaluation Methodology

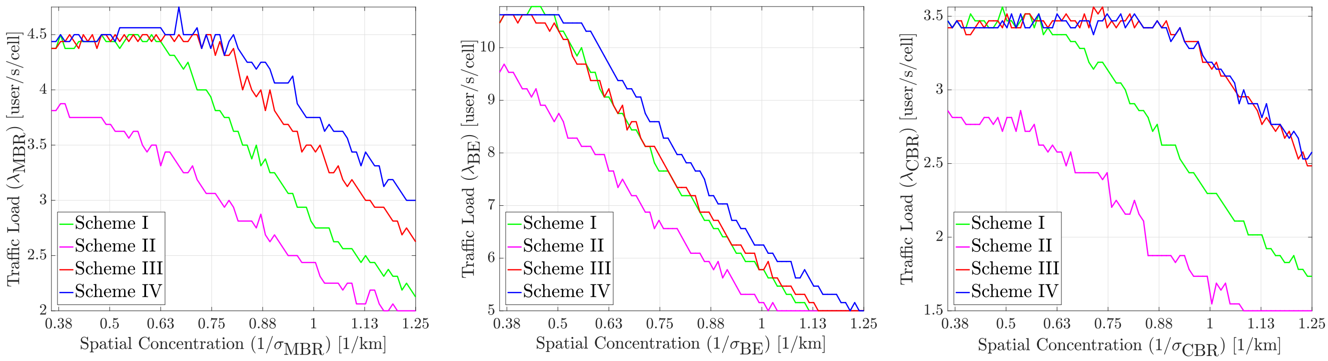
To assess the performance of different schemes, we introduce anomalies to the network and observe how much each scheme can react to these anomalies. These anomalies can be an increase in the traffic load or the concentration factor. The scheme that can fulfill all of the KPIs in higher traffic load and spatial concentration is superior to others. Fig. 2 illustrates the assessment of an example scheme. The infeasible region represents the points (traffic load and spatial concentration) that at least one of the KPIs is under its target from the SLA. On the other hand, the feasible region represents the points that all of the KPIs of all slices are not violated. The border between these two regions is called fulfillment border. As the traffic load or the spatial concentration increases, it is harder to fulfill all the SLAs. Therefore, we are looking for the algorithms that push the fulfillment border to the right and up, meaning that it is able to tolerate more anomalies. Note that even a "genie" algorithm has a fulfillment border since the resources are not sufficient to fulfill all KPIs under very high load or concentration.

TABLE II: Simulation parameters

File size	16 [Mb]
Update interval (τ)	1 [min]
Adaptation step size (δ)	0.1
Simulation duration	1 [hours]
Drop time threshold (θ_D)	8 [sec]
Carrier frequency	2 [GHz]
Downlink transmit power	45 [dBm]
Noise power density	-174 [dBm/Hz]
Propagation model	Free-space path loss + Log-normal shadowing
Interference	Full interference from surrounding cells
Total bandwidth	90 [MHz]
Number of serving cells	7
Number of surrounding cells	12
Cell radius	1 [km]
Shadowing std. dev.	8 [dB]
Antenna Model	Omni-directional
Default load of CBR (λ_{CBR})	3 [users/s/cell]
Default load of BE (λ_{BE})	10 [users/s/cell]
Default load of MBR (λ_{MBR})	4 [users/s/cell]
Default user distribution for all slices	Uniform
Default th_{CBR}	0.33
Default th_{MBR}	0.33
Default w_{BE}	0.5
Default w_{MBR}	0.5
Guaranteed minimum bit rate ($\bar{G}_{\text{MBR}}$)	5 [Mbps]
Guaranteed constant bit rate ($\bar{G}_{\text{CBR}}$)	5 [Mbps]

C. Simulation Results

Fig. 3 illustrates the fulfillment border for different schemes with slice anomalies from different slices. Starting from the Scheme II's performance, we notice that it is actually worse than Scheme I in all of the slice anomalies. The reason for



(a) MBR slice anomalies.

(b) BE slice anomalies.

(c) CBR slice anomalies.

Fig. 3: SLA fulfillment border in presence of anomalies from different slices.

this is that in the distributed control systems, we don’t have any knowledge about the slice performance in the surrounding cells and any slight change in the KPIs will rapidly change the control parameters. In the case of concentration, the central cell observes that almost all of the KPIs are in violation, therefore based on Eq. (11) it decides to decrease th_{CBR}. This effect could be seen as a hasty reaction of the algorithm that ultimately degrades this scheme’s performance. On the other hand, Schemes III and IV have the global knowledge and can avoid making hasty decisions by looking at the KPIs in the whole network. In Scheme I, since the initial control parameters are chosen correctly, with lower concentrations, the performance is similar to Schemes III and IV. This behavior is observable for all slice anomalies. As the concentration factor increases, the initial control parameters are not the right choice anymore and we need an entity that updates these control parameters. Therefore, Schemes III and IV have superior performance. Moreover, in Fig. 3a and 3b we observe that the Scheme IV outperforms Scheme III because the former has the freedom to adjust each cells control parameters individually. However, in Fig. 3c Schemes III and IV have identical performances. The reason is that according to Eq. (11) the Admission rate of the CBR slice (A_{CBR}) is merely affected by the CBR admission threshold th_{CBR} in the central cell and increasing or decreasing the th_{CBR} in the surrounding cells separately does not impact the A_{CBR}.

V. Conclusion

In this article, we have shown that the SLA mapping layer can enhance the performance and robustness of the network against the increase in the traffic load or concentration of the users in a particular cell. This is achieved by iteratively properly orchestrating the control parameters of the PS and AC. The central orchestrating entity, i.e., SLA mapping layer, has been compared with decentralized adaptation and no-adaptation schemes. Simulation results have shown that SLA mapping layer can achieve significant improvements in the network resilience to the anomalies. Additionally, the freedom for choosing the cell-specific control parameters can further improve the network performance.

Acknowledgments

We would like to thank the Center for Information Services and High Performance Computing (ZIH) at TU Dresden for generous allocations of computer time. We would also like to thank Dr. Jobin Francis for his comments and assistance that greatly improved the manuscript.

References

[1] NGMN Alliance, “NGMN 5G White Paper,” Tech. Rep., Feb. 2015.

[2] 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,” *IEEE Communications Magazine*, vol. 55, no. 5, pp. 72–79, May 2017.

[3] NGMN Alliance, “Description of Network Slicing Concept,” Tech. Rep., Jan. 2016.

[4] I. da Silva, G. Mildh, A. Kaloxylos, P. Spapis, E. Buracchini, A. Trogolo, G. Zimmermann, and N. Bayer, “Impact of network slicing on 5G Radio Access Networks,” in *2016 European Conference on Networks and Communications (EuCNC)*, June 2016, pp. 153–157.

[5] 3GPP, “Policy and charging control architecture,” 3rd Generation Partnership Project (3GPP), TR 23.203, Mar. 2018.

[6] C. Liang and F. R. Yu, “Wireless Network Virtualization: A Survey, Some Research Issues and Challenges,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 358–380, Firstquarter 2015.

[7] M. I. Kamel, L. B. Le, and A. Girard, “LTE Wireless Network Virtualization: Dynamic Slicing via Flexible Scheduling,” in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, Sept 2014, pp. 1–5.

[8] M. Jiang, M. Condoluci, and T. Mahmoodi, “Network slicing in 5G: An auction-based model,” in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[9] D. Zhang, Z. Chang, and T. Hamalainen, “Reverse Combinatorial Auction Based Resource Allocation in Heterogeneous Software Defined Network with Infrastructure Sharing,” in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, May 2016, pp. 1–6.

[10] O. Narmanlioglu, E. Zeydan, and S. S. Arslan, “Service-Aware Multi-Resource Allocation in Software-Defined Next Generation Cellular Networks,” *IEEE Access*, vol. 6, pp. 20 348–20 363, 2018.

[11] B. Khodapanah, A. Awada, I. Viering, D. Oehmann, M. Simsek, and G. P. Fettweis, “Fulfillment of Service Level Agreements via Slice-Aware Radio Resource Management in 5G Networks,” in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, June 2018, pp. 1–6.

[12] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects,” 3rd Generation Partnership Project (3GPP), TR 23.203, Mar. 2017.

[13] A. A. Khalek, L. Al-Kanj, Z. Dawy, and G. Turkiyyah, “Optimization Models and Algorithms for Joint Uplink/Downlink UMTS Radio Network Planning With SIR-Based Power Control,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1612–1625, May 2011.