IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 8, 2020, accepted September 20, 2020, date of publication September 23, 2020,
date of current version October 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3026164

Framework for Slice-Aware Radio Resource Management Utilizing Artificial Neural Networks

BEHNAM KHODAPANAH[1], **(Graduate Student Member, IEEE),**
AHMAD AWADA[2], **(Member, IEEE), INGO VIERING**[3], **(Member, IEEE),**
ANDRÉ NOLL BARRETO[4], **(Senior Member, IEEE),**
MERYEM SIMSEK[5], **(Senior Member, IEEE), AND GERHARD FETTWEIS**[1], **(Fellow, IEEE)**
[1]Vodafone Chair Mobile Communication Systems, TU Dresden, 01062 Dresden, Germany
[2]Nokia Bell Labs, 81541 Munich, Germany
[3]Nomor Research GmbH, 81541 München, Germany
[4]Barkhausen Institut, 01187 Dresden, Germany
[5]International Computer Science Institute, Berkeley, CA 94704, USA

Corresponding author: Behnam Khodapanah (behnam.khodapanah@tu-dresden.de)

ABSTRACT For accommodating the heterogeneous services that are anticipated for the fifth-generation (5G) mobile networks, the concept of network slicing serves as a key technology. Spanning both the core network (CN) and radio access network (RAN), slices are end-to-end virtual networks that share the resources of a physical network. Slicing the RAN can be more challenging than slicing the CN since RAN slicing deals with the distribution of radio resources, which have fluctuating capacity and are harder to extend. Improving multiplexing gains, while assuring the slice isolation is the main challenging task for RAN slicing. This paper provides a flexible and configurable framework for RAN slicing, where diverse requirements of slices are simultaneously taken into account, and slice management algorithms adjust the control parameters of different radio resource management (RRM) mechanisms to satisfy the slices' service level agreements (SLAs). One of the proposed algorithms is based merely on heuristics and the other one utilizes an artificial neural network (ANN) to predict the behavior of the cellular network and make better decisions in the adjustment of the RRM mechanisms. Furthermore, a protection mechanism is devised to prevent the slices from negatively influencing each other's performances. A simulation-based analysis demonstrates that in presence of local or global overload of one of the slices, the ANN-based method increases the number of key performance indicators (KPIs) that fulfill their defined SLA targets. Finally, we show that the proposed protection mechanism can force the negative effects of an overloading slice to be contained to that slice and the other slices are not affected as severely.

INDEX TERMS Network slicing, radio resource management, slice orchestration, 5G, iterative adaptation, artificial neural networks.

I. INTRODUCTION

It is anticipated that the fifth-generation (5G) mobile networks will support a multitude of heterogeneous services, such as 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 or financially plausible solution. Network slicing offers a flexible and scalable solution for accommodating diverse services into a single physical network. The service-oriented vision of 5G enforces the decoupling of network infrastructure providers, service providers, and function providers, which allows cost-effective network sharing and reduction of capital expenditure and operating expenses [2]. Network Slicing allows several logical end-to-end networks, i.e., slices, to coexist and efficiently share the physical infrastructure, which brings massive multiplexing gains and increases resource and energy efficiency [3]. Furthermore, since the slices are logically separate networks, they act and can be treated as independent networks.

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Tsun Cheng.

Network slicing allows the slice owners to specify their service requirements in a SLA and the network owner should instantiate an appropriate network slice that meets these SLAs [4], [5]. In the SLAs, the responsibilities of both parties, i.e., slice owner and network owner, should be defined. The network owner agrees to fulfill certain requirements defined by target KPI values. These KPIs could be, for instance, average or fifth-percentile throughputs, delay, or admission rates. On the other hand, the slice owner is responsible for not exceeding traffic load in the network, i.e., the number of requests for data transmission should be limited to a target load.

Network slicing can be performed statically by assigning dedicated resources to each slice or dynamically by adaptively assigning resources to individual slices. Certainly, the dynamic sharing of the common physical infrastructure, i.e., radio resources in RAN slicing, can bring about massive multiplexing gains. This is because unoccupied resources can potentially be utilized by any slice at any point in time. However, a rudimentary dynamic resource sharing leads to slices negatively influencing each other. For instance, an overload of one slice could impact the other slices' KPIs. Therefore, a protection mechanism should be in place, such that violations of the SLA by one slice does not degrade the other slices' performance.

In this work, we start by specifying the different types of slices and propose an entity called *RAN slice orchestrator* that monitors the slice KPIs and network conditions. It ensures the simultaneous fulfillment of the slices' KPIs. 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 the admission control (AC) such that SLA fulfillment is achieved for all slices. We aim to provide a flexible and configurable framework, in which multiple RRM control parameters are fine-tuned such that multiple objectives, i.e., KPIs of the SLAs are satisfied. The slice management algorithms monitor KPIs and network conditions and dynamically alter the control parameters. Note that we aim to introduce a general framework that is agnostic to the RRM mechanisms and the definition of the KPIs.

The algorithms that govern the RAN slice orchestrator entity are iteratively reacting to the monitored KPIs which are measured locally and globally. To respond to different violations, coming from different slices, this entity knows in advance what are the best reactions of the RRM mechanisms that can eliminate these violations. The domain expertise can provide rough guidelines about these reactions and we call this method a heuristics-based approach. We further introduce a prediction ANN that can provide appropriate reactions, based merely on the data, eliminating the need for domain expertise. To implement a protection mechanism, along with monitoring the KPIs, we also monitor the responsibilities of the slice owner, i.e., the load that the slice is introducing. By ignoring the violations from the overloading slices, we protect the slices that are not violating their terms.

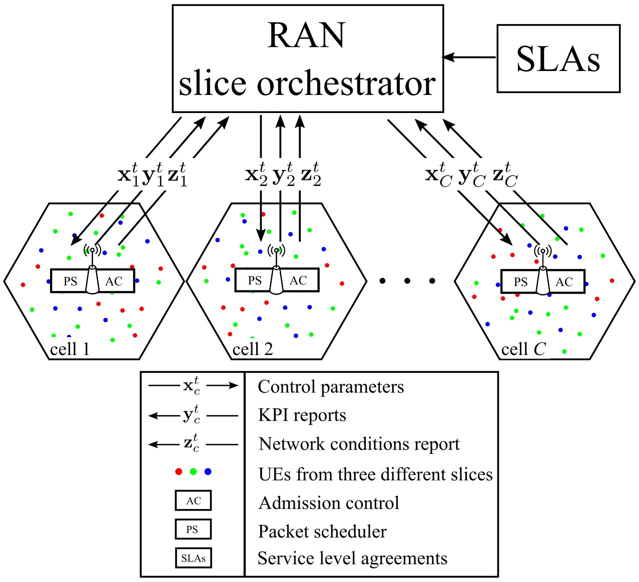


FIGURE 1. **The RAN slice orchestrator takes reports from the network and issues control commands, in order to satisfy the SLAs.**

This article is structured as follows. Section II reviews related work and the state of the art in RAN slicing. In Section III, we describe our system model for a sliced network with slices that have different requirements and formulate the problem. Next, in Section IV, we introduce different slice management algorithms, which orchestrate RRM in an iterative manner such that the KPIs of the slices are fulfilled, while non-overloading slices are prioritized. Simulation results are evaluated in Section V. Finally, we conclude the paper in Section VI.

II. RELATED WORK

The vision of service-oriented networks in 5G and the need for end-to-end network slicing to achieve that goal are reflected in the views multiple standardization bodies and industry forums. International Telecommunication Union (ITU) [6], 3rd Generation Partnership Project (3GPP) [7], Next Generation Mobile Networks (NGMN) [8] and 5G Infrastructure Public Private Partnership (5G PPP) [9] have outlined the necessity of network slicing for future networks. Authors in [10] layout the roadmap for a multi-tenant and multi-service architecture in the future evolution of mobile networks. Such architectures should enable flexible end-to-end slicing via softwarization, virtualization, and disaggregation [11].

Considering that slices are end-to-end networks, slicing spans both the CN and RAN [8]. Slicing the CN has been studied extensively in [12]–[16]. The use of technologies like software-Defined Network (SDN) and Network Function Virtualization (NFV) in efficient architecture design, instantiation, deployment, and maintenance of the CN functions have been investigated. However, RAN slicing deals with the efficient sharing of the radio resources, i.e., time, frequency and space, among slices. Differently from the

CN slicing, the unpredictability and variability of the wireless medium makes the RAN slicing a more challenging topic [11]. In particular, Radio Resource Management (RRM) is a crucial mechanism for ensuring the simultaneous fulfillment of the demands of the different slices. According to [17]–[19], the RAN slicing requirements are resource sharing, RAN slice-awareness, Quality-of-Service (QoS) support, SLA enforcement, slice isolation (protection), performance monitoring and slice-tailored SDN.

The objectives of RRM in a sliced network have been addressed separately for legacy mobile networks. Fulfilling user requirements via QoS Class Identifier (QCI) mechanisms has been proposed in 3GPP Long-Term Evolution (LTE) systems [20]. Based on the requirements of each user, an appropriate QCI is assigned to guarantee a certain service quality with regard 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 users belonging to a slice, the KPIs that describe their collective performance should meet at least a target 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) [21]–[24]. 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 dynamic sharing of radio resources has been studied in [25], the impact of negative inter-network influences have not been analyzed.

Some previous works have focused on improving the utility of the whole network and maximizing its sum rate. In these studies, optimizing the distribution of resources in PS is approached via reinforcement learning [26], auction-based models [27]–[29] or game theory [30]. Because of the potentially diverse nature of the slices, maximizing the sum-rate of the network is not solely sufficient and the SLAs of the slices should be taken into consideration. Furthermore, the protection mechanism is one of the main challenges of RAN slicing since in a sliced network, traffic load anomalies of one slice (e.g. in terms of introduced load) can heavily influence the performance of the other slices.

Many studies have proposed a two-layer scheduling for slicing RAN, where the inter-slice scheduler decides how to distribute the resources among the slices and an intra-slice scheduler distributes the resources among the users of that slice. The inter-slice scheduling in [31], [32] is based on fixed allocation and in [33], [34] it is based on heuristics. While guaranteeing the slice isolation in such frameworks is more straightforward, the multiplexing gains remain inevitably reduced. In [35], a flexible design allows for switching between multiplexing gains and isolation properties. In [36] and [37] dynamic resource provisioning and deep reinforcement learning techniques for inter- and intra-slice RRM, with the presence of diverse slices, have been proposed. In these works, the performance evaluation metrics are system satisfaction, utility, slice isolation. However, the fulfillment of slice-specific KPIs, which represent the collective performance of the users of a slice has not been defined.

In this work, we try to address the shortcomings of the previously mentioned studies. Not only QoS requirements of individual users are considered, but also the collective behavior of the slices is tracked and compared with SLA targets, going significantly beyond studies that focus on single objectives. Moreover, we prioritize slices based on their deviation from loads agreed on in their SLAs and the conforming slices are isolated from the negative influences of the slices that are violating their slice owner responsibilities.

III. SYSTEM MODEL AND PROBLEM FORMULATION

Consider a mobile cellular network with $c = 1, 2, \ldots, C$ cells and let $\mathcal{S}$ be the set of all slices. The total number of slices in the network is denoted by $S = |\mathcal{S}|$. The users belonging to these slices arrive in the network at random times and locations. They intend to download a file and leave the network (FTP traffic model [38]). Subsection III-A describes the random processes behind this procedure. In Subsection III-B, we elaborate on the slice type with diverse requirements. Moreover, in Subsection III-C, the details of the PS and AC mechanisms are described. Finally, the slice-aware RRM problem is formulated and discussed in Subsection III-D.

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 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 truncated 2D Gaussian distribution [39]. 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 radially symmetric with respect to its origin. Note that we truncate the 2D Gaussian distribution to be 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 $1/\sigma$. As the concentration factor approaches 0, the users are more uniformly distributed. Fig. 2 illustrates the network layout with 21 sectors and the non-uniform spatial distribution. We can observe that sectors 19, 20 and 21 face the highest load. Sectors 6, 10 and 17 experience medium load and the rest of the sectors have a relatively low load.

B. SLICES TYPES WITH DIVERSE REQUIREMENTS

To model slices with different requirements, we define three slice types. We assume that in general, several instances of these slice types might be present in the network. We view

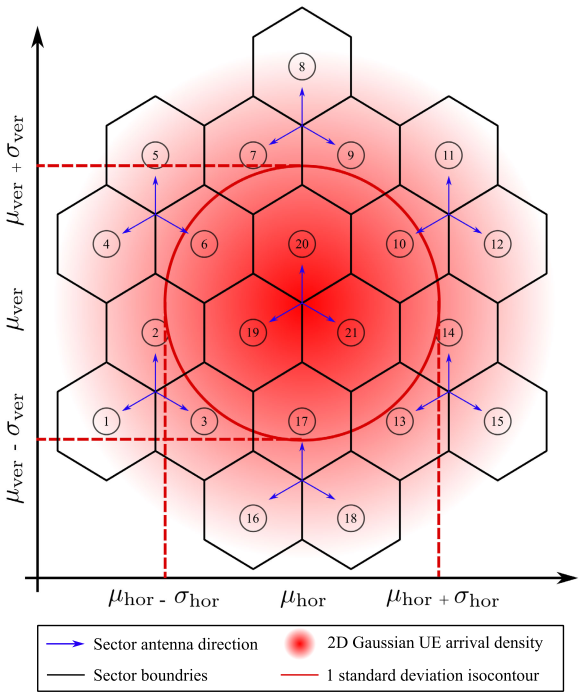

$\mu_{\text{ver}} + \sigma_{\text{ver}}$
μ_{ver}
μ_{ver} - σ_{ver}
μ_{hor} - σ_{hor}
μ_{hor}
$\mu_{\text{hor}} + \sigma_{\text{hor}}$
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
Sector antenna direction
2D Gaussian UE arrival density
Sector boundries
1 standard deviation isocontour

FIGURE 2. Cellular network layout with 21 sectors. The non-uniform spatial distribution is considered to be a truncated 2D Gaussian.

these slice types as the slice templates to be instantiated every time a new slice is added to the network. The slices that belong to the same slice type have the same KPIs as requirements. However, the slices can have their own specific target values. The introduced framework is not limited by the introduced service types here and slice types with different KPIs or different RRM mechanisms can be added.

- *Best Effort (BE)*:
  The users belonging to this slice type do not have any requirements on their instantaneous throughput. Services like eMBB and applications like web-browsing are reasonable examples of this slice type, as they are not very sensitive to instantaneous bit rate. However, the long-term average of the users' throughputs (T_{BE}) and the fifth-percentile throughput (F_{BE}) KPIs must be above the targets that have been declared in the SLA, i.e., $\overline{T}_{\text{BE}}$ and $\overline{F}_{\text{BE}}$, respectively. Moreover, to prevent the number of active best effort (BE) users in the network to grow indefinitely, we assume that the users of a BE slice are dropped from the network if they remain active longer than a time threshold θ_D. This mechanism ensures that even under very congested conditions, the number of users cannot grow indefinitely. Although this mechanism ensures network stability, it is not in the interest of BE slices to have its users dropped frequently. Therefore, the dropping rate (D_{BE}) should be below a target defined in the SLA, which we denote as $\overline{D}_{\text{BE}}$. For convenience, we use the KPI $1 - D_{\text{BE}}$ and acceptable dropping rates are achieved above $1 - \overline{D}_{\text{BE}}$. Finally, we assume that AC admits all BE users.
- *Constant Bit Rate (CBR)*:
  The admitted users of a constant bit rate (CBR) slice are guaranteed to have a constant throughput, regardless of the user's channel conditions. URLLC services can be considered as a CBR slice type, since for such services the required payload is constant but it is crucial that the bit rate remains constant. Since the throughput is constant for all users, the only KPI that is associated with this slice is the admission rate (A_{CBR}) which has to be above the target in the SLA, i.e., $\overline{A}_{\text{CBR}}$.
- *Minimum Bit Rate (MBR)*:
  Similar to BE users, the minimum bit rate (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 acceptable quality. This slice type can represent both of the eMBB and URLLC services. The average throughput of MBR users (T_{MBR}) and the admission rate (A_{MBR}) are the considered KPIs for this slice type. These KPIs should be above the targets in the SLA, i.e., $\overline{T}_{\text{MBR}}$ and $\overline{A}_{\text{MBR}}$. 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 of the admitted users.

We define $\mathcal{S}_{\text{BE}}$, $\mathcal{S}_{\text{CBR}}$ and $\mathcal{S}_{\text{MBR}}$ to be the sets of all BE, CBR and MBR slices, respectively. Table 1 summarizes the different slice types. The traffic parameters for each of these slice types are arrival rate λ_s, which represents the load that slice s is introducing; file size f_s, which determines the amount of data downloaded in each session; and the constant/minimum guaranteed bit rate $\overline{G}_s$. These traffic parameters are agreed upon in the SLA and the network owner and the slice owner should fulfill their responsibilities, or else suffer the penalties defined in the contract. The control parameters refer to the respective RRM mechanisms, which are described in Subsection III-C. The output KPIs are the live measurements of the KPIs in the network and KPI targets are agreed upon in the SLAs.

In this work, we do not introduce slices that are not demanding in terms of throughput or delay, but the challenge is the number of users, i.e., in order of millions of devices. mMTC services could be instances such slices. From a system-level view of RAN slicing, the relaxed requirements of such slices in terms of throughput or delay means that the RRM entity can assign resources to the users of such slices, when the demand from other slices is not high, e.g., in the midnight times. For other types of services however, the coexistence in the temporal and spatial domain is inevitable.

C. RADIO RESOURCE MANAGEMENT MECHANISMS

In legacy networks, the requirements of the users are satisfied via RRM mechanisms that are QoS-aware. However, in a

TABLE 1. Slice types with diverse requirements.

Slice Type	Best Effort (BE)	Constant Bit Rate (CBR)	Minimum Bit Rate (MBR)
Example service	eMBB	URLLC	eMBB and URLLC
Traffic parameters	• Arrival rate λ_s • File size f_s	• Arrival rate λ_s • Constant bit rate $\overline{G}_s$ • File size f_s	• Arrival rate λ_s • Minimum bit rate $\overline{G}_s$ • File size f_s
Control parameters	• Scheduler weights ξ_s	• Admission threshold η_s	• Scheduler weights ξ_s • Admission threshold η_s
Output KPIs	• Average throughput T_s • Fifth percentile throughput F_s • 1 - Dropping rate $1 - D_s$	• Admission rate A_s	• Average throughput T_s • Admission rate A_s
KPI targets	• $\overline{T}_s = 80$ Mbps • $\overline{F}_s = 3$ Mbps • $1 - \overline{D}_s = 97$ %	• $\overline{A}_s = 97.5$ %	• $\overline{T}_s = 90$ Mbps • $\overline{A}_s = 98$ %

sliced mobile network, not only the QoS of individual users should be satisfied, but also the requirements of different slices should be fulfilled. Therefore, the RRM should be slice-aware, such that the collective performance of the users of the different slices can be controlled. To enable slice-aware RRM, we introduce slice-specific control parameters that control the PS and the AC processes.

1) PACKET SCHEDULER (PS)

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

$$R^i_{s,c} = \omega^i_{s,c} \cdot B \cdot \log_2(1 + \Psi^i_{s,c}), \tag{1}$$

where $\omega^i_{s,c} \in [0, 1]$ is the normalized resource share of the user i in slice s and cell c, $\Psi^i_{s,c}$ is the ith user signal-to-interference-plus-noise-ratio (SINR) of and B is the total system bandwidth. We focus on system-level aspects of RAN slicing and make assumptions and abstractions for the physical layer mechanisms. The main outcome of this work is not affected by these assumptions. We assume that instead of physical resource block (PRB) assignment, fractional resources can be assigned to the users. For calculating the signal-to-interference-plus-noise-ratio (SINR), first, we assume the worst-case scenario that the interference is always present from surrounding cells. This way, the interference could be calculated without implementing specific multiple access schemes [40]. Next, in order to make the simulation of longer intervals (in order of hours) feasible, the mobility aspect of the users is ignored. Due to this assignments, the SINR values are constant during the download session.

For CBR users, a constant throughput is guaranteed and specified in the SLA as $\overline{G}_s$. Consequently, the amount of resources needed to achieve the target throughput for every user belonging to slice $s \in \mathcal{S}_{\text{CBR}}$ and in cell c is given by

$$\omega^i_{s,c} = \frac{\overline{G}_s}{B \cdot \log_2(1 + \Psi^i_{s,c})}. \tag{2}$$

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

$$\Omega_{\text{CBR, c}} = \sum_{s \in \mathcal{S}_{\text{CBR}}} \sum_{i=1}^{N_{s,c}} \omega^i_{s,c}. \tag{3}$$

The remaining, i.e., $1 - \Omega_{\text{CBR, c}}$, are shared among MBR and BE users.

To model PS of the 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 $\boldsymbol{\xi}_c = [\xi_{1,c}, \xi_{2,c}, \cdots, \xi_{|\mathcal{S}_{\text{BE}} \cup \mathcal{S}_{\text{MBR}}|,c}]$, where $\mathcal{S}_{\text{BE}} \cup \mathcal{S}_{\text{MBR}}$ constitutes all the BE and MBR slices. The resource share of user $i = 1, 2, \cdots, N_{s,c}$ belonging to slice $s \in \mathcal{S}_{\text{BE}} \cup \mathcal{S}_{\text{MBR}}$ is defined as

$$\omega^i_{s,c}(\boldsymbol{\xi}_c) = \frac{\xi_{s,c} \cdot (1 - \Omega_{\text{CBR, c}})}{\sum\limits_{s' \in \mathcal{S}_{\text{BE}}} N_{s',c} \cdot \xi_{s',c} + \sum\limits_{s'' \in \mathcal{S}_{\text{MBR}}} N_{s'',c} \cdot \xi_{s'',c}}, \tag{4}$$

where

$$\sum_{s' \in \mathcal{S}_{\text{BE}}} \xi_{s',c} + \sum_{s'' \in \mathcal{S}_{\text{MBR}}} \xi_{s'',c} = 1. \tag{5}$$

If we only use (4) for the MBR and BE users, there might be some MBR users that are not assigned enough resources to achieve their minimum throughput. To simultaneously use (4) and fulfill the MBR requirement, we propose an iterative scheduling solution. First, the resources are shared based on (4). If any of the MBR users has lower throughput than the minimum bit rate, as in (2), the minimum required resources are determined and assigned accordingly. Let $\breve{N}_{s''}$ be the number of users that have received this special treatment,

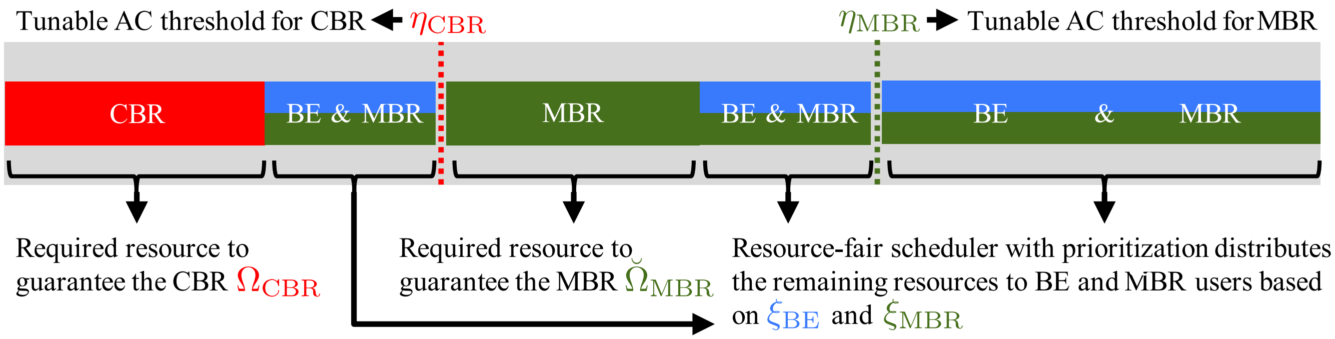

Tunable AC threshold for CBR
η_{CBR}
η_{MBR}
Tunable AC threshold for MBR
CBR
BE & MBR
MBR
BE & MBR
BE & MBR
Required resource to guarantee the CBR Ω_{CBR}
Required resource to guarantee the MBR $\breve{\Omega}_{\mathrm{MBR}}$
Resource-fair scheduler with prioritization distributes the remaining resources to BE and MBR users based on ξ_{BE} and ξ_{MBR}

FIGURE 3. Radio resource utilization of BE, CBR and MBR slices in the slice-aware radio resource management scheme. The control parameters of the AC (η_{CBR} and η_{MBR}) and the PS (ξ_{MBR} and ξ_{BE}) should be adjusted dynamically so that the slices' SLAs are fulfilled.

where $s'' \in \mathcal{S}_{\mathrm{MBR}}$. The collective resource consumption of the users of these slices is

$$\breve{\Omega}_{\mathrm{MBR},c} = \sum_{s'' \in \mathcal{S}_{\mathrm{MBR}}} \sum_{i=1}^{\breve{N}_{s'',c}} \omega^{i}_{s'',c}. \tag{6}$$

After this special treatment of some MBR users, the resource share of users of slices s in $\mathcal{S}_{\mathrm{BE}} \cup \mathcal{S}_{\mathrm{MBR}}$ is defined as

$$\omega^{i}_{s,c}(\boldsymbol{\xi}_c) = \frac{\xi_{s,c} \cdot (1 - \Omega_{\mathrm{CBR},c} - \breve{\Omega}_{\mathrm{MBR},c})}{\sum\limits_{s' \in \mathcal{S}_{\mathrm{BE}}} N_{s',c} \cdot \xi_{s',c} + \sum\limits_{s'' \in \mathcal{S}_{\mathrm{MBR}}} \hat{N}_{s'',c} \cdot \xi_{s'',c}}, \tag{7}$$

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 (7), there might be some MBR users whose resource share is not sufficient. Therefore, the iteration is repeated until all the MBR users are satisfied. The process is guaranteed to terminate, since the number of admitted users of MBR slices is limited via AC.

2) ADMISSION CONTROL (AC)

The role of AC in the network is to regulate the incoming traffic. Tenants want the admission rate to be as high as possible. However, by admitting more users, other network KPIs are affected. AC is especially crucial in sliced networks, since too many users from one slice might negatively impact the other slices. We define slice-specific resource thresholds. For all MBR and CBR slices $\in \mathcal{S}_{\mathrm{CBR}} \cup \mathcal{S}_{\mathrm{MBR}}$, when a user appears in slice s, the admission policy is

$$\begin{cases} \text{If } \Omega_{s,c} \leq \eta_{s,c} & \text{grant admission} \\ \text{If } \Omega_{s,c} > \eta_{s,c} & \text{deny admission,} \end{cases} \tag{8}$$

where $\eta_{s,c}$ is the resource threshold for slice s and $\Omega_{s,c} = \sum_{i=1}^{N_{s,c}} \overline{G}_s/B \cdot \log_2(1 + \Psi^{i}_{s,c})$ is the minimum amount of resources required to satisfy the MBR or CBR slice. In other words, if a new user increases $\Omega_{s,c}$ above $\eta_{s,c}$, admission is denied.

Since the total amount of resources is normalized to one, the sum of the resource thresholds may not exceed one. On the other hand, it should be remembered that BE slices should not be sacrificed for CBR and MBR slices. Therefore, it is necessary to limit the sum of the resource thresholds (that are only used by CBR and MBR slices) to less than one. Here we define the *reserve threshold* $\eta_{\mathrm{res},c}$ and the following holds true for the resource thresholds

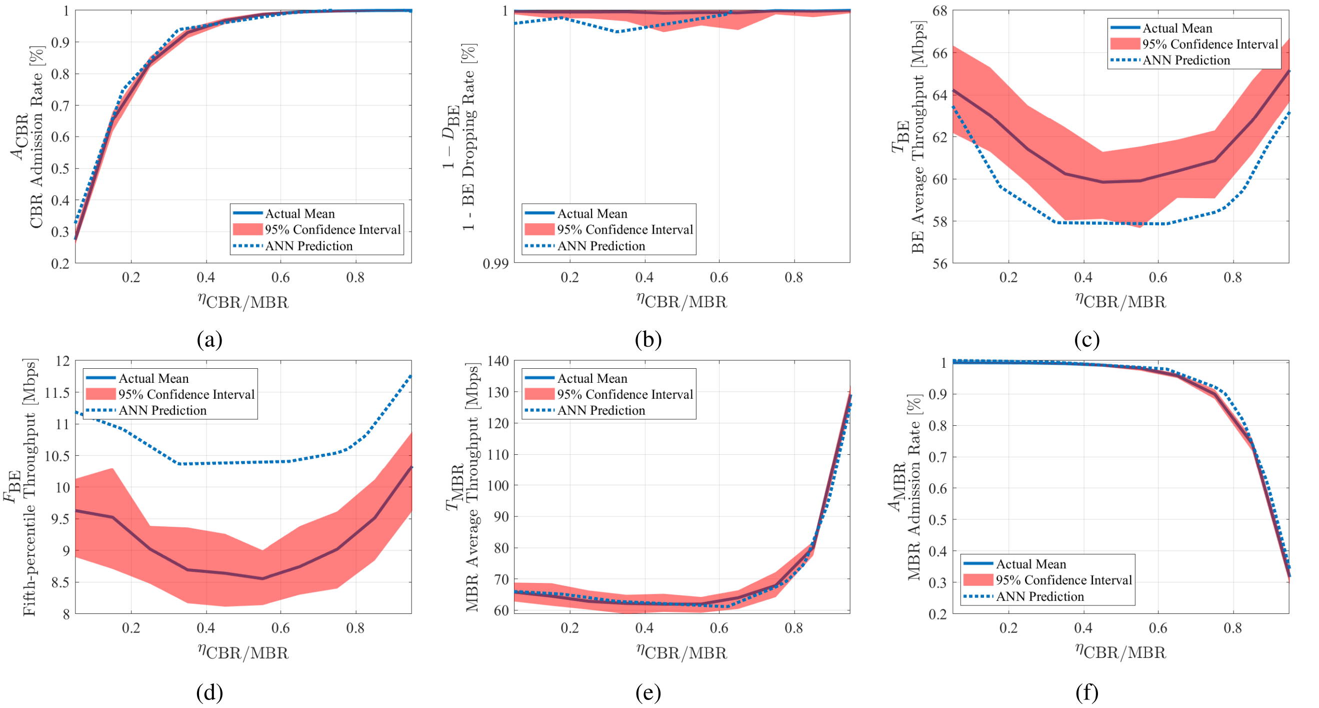
$$\sum_{s \in \mathcal{S}_{\mathrm{CBR}} \cup \mathcal{S}_{\mathrm{MBR}}} \eta_{s,c} = 1 - \eta_{\mathrm{res},c}. \tag{9}$$

Fig. 3 exemplifies how the resources are dynamically allocated among the slices with the proposed slice-aware RRM. Without loss of generality, we assume one instance of each slice types in the remainder of this paper. $\eta_{\mathrm{res},c}$ and the slice-specific control parameters η_{MBR}, η_{CBR}, ξ_{MBR} and ξ_{BE} control the AC and PS processes in each cell. Tuning these parameters effects the KPIs of the slices. With three slices, there are three degree of freedom (DoF) for RRM. Determining ξ_{MBR} implicitly sets ξ_{BE} to $1 - \xi_{\mathrm{MBR}}$ (see (5)). Also, note that the $\eta_{\mathrm{MBR}} + \eta_{\mathrm{CBR}} + \eta_{\mathrm{res},c} = 1$, which means that determining two of them implicitly sets the other one to make use of all resources. To unify the treatment of all control parameters, we project these control parameters to three parameters $\in [0, 1]$. With this transformation, the 3D control parameter space is a unit cube-shaped volume that occupies $[0, 1]$ in each dimension.

The first DoF is η_{res}, which describes the reserved resources for non-guaranteed bit rate (GBR) slices and is in the interval $[0, 1]$. The ratio between η_{CBR} and η_{MBR} is another degree of freedom, which we denote as $\eta_{\mathrm{CBR/MBR}}$ and is also situated in the interval $[0, 1]$. The resource threshold for the CBR is $\eta_{\mathrm{CBR}} = (1 - \eta_{\mathrm{res}})\eta_{\mathrm{CBR/MBR}}$ and the resource threshold for MBR is $\eta_{\mathrm{MBR}} = (1 - \eta_{\mathrm{res}})(1 - \eta_{\mathrm{CBR/MBR}})$. The final degree of freedom is the scheduler weight $\xi_{\mathrm{BE/MBR}}$, which is also in the interval $[0, 1]$. The scheduler weight for the BE slice is $\xi_{\mathrm{BE}} = \xi_{\mathrm{BE/MBR}}$ and the scheduler weight for the MBR slice is $\xi_{\mathrm{MBR}} = 1 - \xi_{\mathrm{BE/MBR}}$.

D. PROBLEM FORMULATION

The aim of the slice-aware RRM is that, by tuning the slice-specific control parameters of PS and AC, the KPI targets defined in the SLAs are achieved by the network. In our system model, the scheduler weights $\xi^{t}_{s,c}$ $\forall s \in \mathcal{S}_{\mathrm{BE}} \cup \mathcal{S}_{\mathrm{MBR}}$



(a) (b) (c)

(d) (e) (f)

FIGURE 4. Change of KPIs in response to changes in one of the control parameters. The $\eta_{\mathrm{res}} = 0.05$ and $\xi_{\mathrm{BE/MBR}} = 0.5$ are kept constant. The actual mean and confidence interval are based on independent simulations and the ANN predictions are based on the trained ANN.

and admission thresholds $\eta^t_{s,c}$ $\forall s \in \mathcal{S}_{\mathrm{CBR}} \cup \mathcal{S}_{\mathrm{MBR}}$, in each cell $c = 1, \cdots, C$ are the control parameters to steer. Note that we have also included t as an index for the time interval. We consider an interval $[(t-1)\tau, t\tau]$, where τ is the duration of each interval. During each interval t, the KPIs are measured, based on which, the control parameters are updated at interval $t+1$. At each cell c the relationship between the local KPIs and both the local control parameters and local conditions can be stated as

$$\mathbf{y}^t_c = f_c(\mathbf{x}^t_c | \mathbf{z}^t_c), \tag{10}$$

where $\mathbf{y}^t_c$ is the vector containing the local KPIs and the local control parameter vector is defined as

$$\mathbf{x}^t_c = [\xi^t_{\mathrm{BE/MBR},c}, \eta^t_{\mathrm{res},c}, \eta^t_{\mathrm{CBR/MBR},c}]^\top. \tag{11}$$

The network condition vector is defined as

$$\mathbf{z}^t_c = [\lambda^t_{\mathrm{CBR},c}, \lambda^t_{\mathrm{BE},c}, \lambda^t_{\mathrm{MBR},c}, \psi^t_{\mathrm{CBR},c}, \psi^t_{\mathrm{BE},c}, \psi^t_{\mathrm{MBR},c}]^\top, \tag{12}$$

where $\lambda^t_{s,c}$ is the load of slice s and $\psi^t_{s,c}$ is the average SINR of the users of slice s, defined as

$$\psi^t_{s,c} = \frac{1}{N_{s,c}} \sum_{i=1}^{N_{s,c}} \Psi^i_{s,c}. \tag{13}$$

Similarly, we define the relationship of global KPIs to all of the control parameters and network conditions as

$$\mathbf{y}^t_g = f_g(\mathbf{X}^t_g | \mathbf{Z}^t_g), \tag{14}$$

where $\mathbf{y}^t_g$ is the vector containing the global KPIs, $\mathbf{X}^t_g = [\mathbf{x}^t_1, \mathbf{x}^t_2, \cdots, \mathbf{x}^t_C]$ is a matrix containing the slice-specific control parameters for all cells and $\mathbf{Z}^t_g = [\mathbf{z}^t_1, \mathbf{z}^t_2, \cdots, \mathbf{z}^t_C]$ represents all the relevant network conditions that play a role in determining the KPIs. These conditions cannot be influenced by the RRM and the slice manager can only tune $\mathbf{X}^t_g$. The goal is to find a proper set of control parameters so that all of the KPIs meet or exceed their target values. One way to approach this problem is to solve an optimization problem, where the error function (or cost function)

$$E = \left\| \mathbf{y}^t_g - \bar{\mathbf{y}} \right\|_2 = \left\| f_g(\mathbf{X}^t_g | \mathbf{Z}^t_g) - \bar{\mathbf{y}} \right\|_2, \tag{15}$$

should be minimized [41]. However, to solve this optimization problem, a model of the RAN is required, i.e., $f_g(\cdot)$ needs to be expressed analytically. Since all inter-dependencies in $\mathbf{Z}^t_g$ and $\mathbf{X}^t_g$ are not easily available, $f_g(\cdot)$ is not available. Fig. 4 illustrates the effects of changing one of the control parameters on the KPIs of the slices. Moreover, the KPIs have different units and thus create costs on different scales. Hence, their summation to a single term in (15) implicitly and unfairly discriminates between KPIs. For instance, the unit of the admission rates are percentages and the unit of average throughput is throughput per time and there is no simple way of directly comparing the two. To avoid this implicit discrimination of the KPIs, we aim to achieve the target values for all KPIs and do not compare them with each other. We consider the problem to be a multi-objective optimization, with a binary fulfillment criteria for each KPI. In the following section, an adaptive and iterative framework is introduced for maximizing the number of achieved KPI targets.

IV. SLICE MANAGEMENT ALGORITHM

In this section we introduce an adaptive algorithm that iteratively changes the RRM control parameters such that the multi-objective optimization that is introduced in Subsection III-D is solved, i.e., the number of fulfilled KPIs is maximized. To do so, we first take a look at the Jacobian matrix of (10), which is noted as $\mathbf{J}$ and has the dimension $K \times P$, where K is the total number of KPIs and P is the total number of slice-specific control parameters in each cell. The element $[j_{i,j}]$ represents the first-order derivative of the i-th KPI with regard to the j-th control parameter, i.e.,

$$\mathbf{J} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_P} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_K}{\partial x_1} & \cdots & \frac{\partial y_K}{\partial x_P} \end{bmatrix}. \tag{16}$$

Thereafter, we define the *reaction matrix* $\hat{\mathbf{J}}$, which is a quantization of $\mathbf{J}$. The element in the i-th row and j-th column is

$$\hat{j}_{i,j} = \begin{cases} 0 & \text{increase in } j \text{ does not affect KPI } i \\ +1 & \text{increase in } j \text{ increases KPI } i \\ -1 & \text{increase in } j \text{ decreases KPI } i. \end{cases} \tag{17}$$

Note that $\hat{\mathbf{J}}$ is quantizing the $\mathbf{J}$ by taking only the signs of the first-order derivative. Two methods for determining this matrix are explained in IV-A. With this coarse knowledge about the relationship between the KPIs and the control parameters, we propose an algorithm for slice management in RAN. The idea is that whenever there is a violation in any of the KPIs, the matrix $\hat{\mathbf{J}}$ points out which parameters to increase, decrease, or do not change.

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

$$\mathbf{v}_g^t[k] = H(\bar{\mathbf{y}}[k] - \mathbf{y}_g^t[k]) = \begin{cases} 1 & \text{if } \mathbf{y}_g^t[k] < \bar{\mathbf{y}}[k] \\ 0 & \text{if } \mathbf{y}_g^t[k] > \bar{\mathbf{y}}[k], \end{cases} \tag{18}$$

where $\mathbf{v}_g^t[k]$, $\mathbf{y}_g^t[k]$ and $\bar{\mathbf{y}}[k]$ are the k-th KPI in the violation, global KPI and the target KPI vector, respectively. Note that the step function $H(\cdot)$ is used in (18) rather than the conventional mean square error metric. As mentioned in Subsection III-D, the reason is that the KPIs have different units (e.g., admission rate is measured in percentages and average throughput in 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, i.e., the binary fulfillment criteria.

So far we have assumed that the KPI reports $\mathbf{y}_g^t$ are collected over the whole network, referring to global KPIs. However, we also need to define local KPI reports in cell c as $\mathbf{y}_c^t$ and the associated local violation vector as $\mathbf{v}_c^t$ (18). The combined violation vector in cell c is then defined as

$$\tilde{\mathbf{v}}_c^t = \mathbf{v}_g^t \odot \mathbf{v}_c^t \tag{19}$$

where $\odot$ is element-wise multiplication (Hadamard product). The reason for combining local and global violation vectors is that the control parameters should only change in the cells where KPI targets are violated and not in the entire network. On the other hand, if a KPI target is not globally violated, there is no need to react locally. In this way, violations are only registered if there is a local and global violation.

Using the combined violation vector $\tilde{\mathbf{v}}_c^t$, we know which KPIs are not satisfied and with the reaction matrix $\hat{\mathbf{J}}$, we know which control parameters should be changed. Therefore, the update rule at interval t for control parameters in cell c is defined as

$$\mathbf{x}_c^{t+1} = \mathbf{x}_c^t + \delta \hat{\mathbf{J}}^\top \tilde{\mathbf{v}}_c^t, \tag{20}$$

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

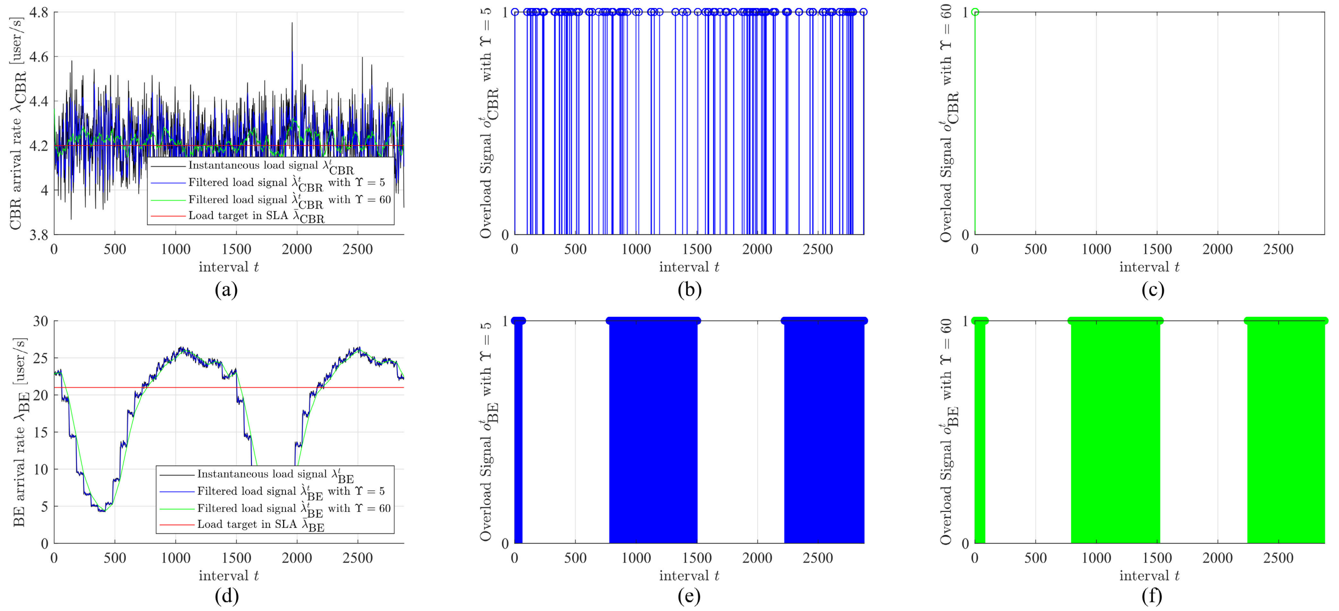
A. APPROXIMATING THE REACTION MATRIX

In the following, we introduce two methods for acquiring the reaction matrix of (17). The first method is based on a static and heuristics-based reaction matrix. This method heavily relies on domain expertise and cannot be generalized effortlessly. For the second method, we make use of a ANN as a candidate function approximator and show that the reaction matrix can be constructed dynamically and without domain expertise.

1) HEURISTICS-BASED REACTION MATRIX

One method for constructing a reaction matrix $\hat{\mathbf{J}}$ is to manually relate the K KPIs' reactions to X control parameters, using heuristics that are based on general domain expertise. The proposed reaction matrix in (21), as shown at the bottom of the next page.

The only KPI of the CBR slice is the admission rate A_{CBR} and it increases if η_{res} is decreased or if the ratio between the CBR and MBR resource threshold, i.e., $\eta_{\text{CBR/MBR}}$, is increased. Moreover, $\xi_{\text{BE/MBR}}$ does not affect the performance of the CBR slice, since it is not affected by the scheduler weights. The KPIs of the BE slice only increases if the scheduler prioritizes it over the MBR, i.e., increase $\xi_{\text{BE/MBR}}$. Besides, if the resource thresholds of the CBR and MBR (η_{res}) slices are decreased, there will be less users from those slices to compete with the BE users. Finally, if the CBR resource threshold is larger than the MBR resource threshold (increase in $\eta_{\text{CBR/MBR}}$), there will be less MBR users that compete with BE users. Regarding the KPIs of the MBR slice, if the η_{res} is increased, the number of admitted MBR users (A_{MBR}) decreases, but at the same time, there will be less users and thus the average user throughput (T_{MBR}) increases. By increasing the ratio of scheduler weights of BE and MBR slices, i.e. $\xi_{\text{BE/MBR}}$, MBR users will obviously suffer. Similar to increases of η_{res}, increasing $\eta_{\text{CBR/MBR}}$, decreases the



(a) (b) (c)

(d) (e) (f)

FIGURE 5. Load signals (λ_{CBR} and λ_{BE}) and violation signals (o^t_{CBR} and o^t_{BE}). CBR slice has a nominal load (top-row) and BE has a load based on a daily traffic profile (bottom-row). With increasing Υ the load estimation precision is increased and the overload signals are less prone to error.

number of admitted MBR users while increasing the average user throughput.

Determination of the reaction matrix $\hat{\mathbf{J}}$ requires heuristics and domain expertise, which might not be easily available. Moreover, the inaccuracies of an heuristics-based approximation could inevitably drive the control parameters to a space that might not be optimal. Furthermore, a static reaction matrix is not a good representation for all control parameters ($\mathbf{X}^t_g$) or network conditions ($\mathbf{Z}^t_g$), because of their interdependencies. For instance, as Fig. 4 illustrates, T_{BE} and F_{BE} are not monotonic with regard to the $\eta_{\text{CBR/MBR}}$ and a static reaction matrix cannot capture this effect. Hence, we must conclude that such a static approach also requires a good starting point for the control parameters, where the reaction matrix is most valid.

2) ANN-BASED REACTION MATRIX

To avoid the drawbacks of the previous approach, we introduce a method for dynamically estimating the reaction matrix $\hat{\mathbf{J}}$. Firstly, we intend to approximate (10). We do so by means of a supervised-learning approach, in which an ANN is trained. The ANN has three fully-connected (dense) layers with 50 nodes each and with Rectified Linear Unit (ReLU) activation functions. For more in depth study of the fundamentals of ANNs refer to [42]. Fig. 6 depicts the architecture of this ANN. The input of this ANN is the local control parameters vector $\mathbf{x}^t_c$ and local conditions vector $\mathbf{z}^t_c$ and the output is the local KPIs vector $\mathbf{y}^t_c$. After collecting diverse data of the inputs and outputs the training is performed. This diverse data can be recycled from the history of the heuristics-based approach. As shown in Fig. 4, the ANN can capture a coarse estimation of the network behaviour, i.e., (10) can be replaced by

$$\hat{\mathbf{y}}^t_c = \hat{f}_c(\mathbf{x}^t_c|\mathbf{z}^t_c), \tag{22}$$

where $\hat{\mathbf{y}}^t_c$ is the output of the ANN and $\hat{f}_c(\cdot)$ represents the ANN. To derive a reaction matrix from the trained ANN, we approximate the first-order derivatives in (16) with finite difference:

$$\frac{\partial y_i}{\partial x_j} \approx \frac{\partial \hat{y}_i}{\partial x_j} \approx \frac{\hat{y}_i(x_j+\delta) - \hat{y}_i(x_j-\delta)}{2\delta}, \tag{23}$$

where y_i is the actual i-th KPI, $\hat{y}_i$ is the corresponding estimation from the ANN, δ is the adaptation step size and x_j

$$\hat{\mathbf{J}}^\top = \begin{array}{cccccccc} & \rotatebox{90}{A_{CBR}} & \rotatebox{90}{$1-D_{\text{BE}}$} & \rotatebox{90}{T_{BE}} & \rotatebox{90}{F_{BE}} & \rotatebox{90}{T_{MBR}} & \rotatebox{90}{A_{MBR}} & \\ \left[\right. & -1 & +1 & +1 & +1 & +1 & -1 \left.\right] & \eta_{\text{res}} \\ & 0 & +1 & +1 & +1 & -1 & -1 & \xi_{\text{BE/MBR}} \\ & +1 & +1 & +1 & +1 & +1 & -1 & \eta_{\text{CBR/MBR}} \end{array}. \tag{21}$$

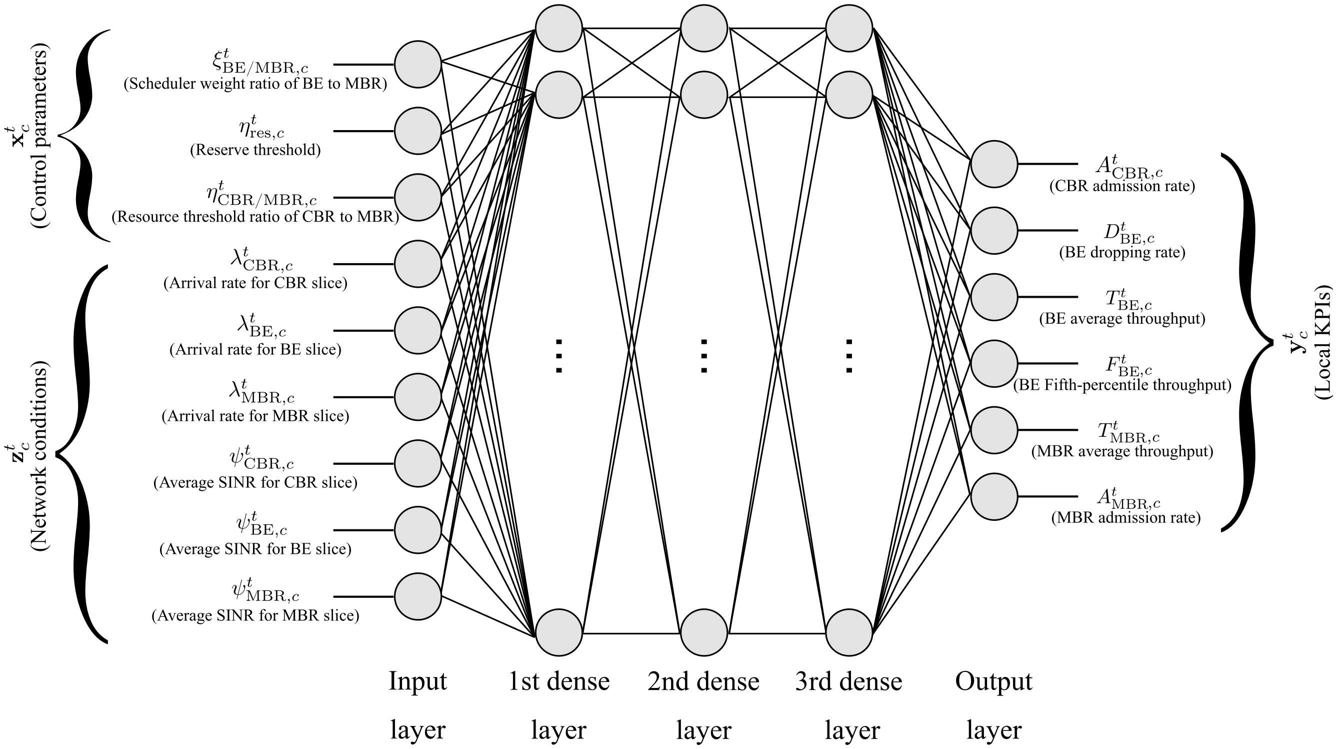

$\mathbf{x}_c^t$ (Control parameters)
$\xi^t_{\mathrm{BE/MBR},c}$ (Scheduler weight ratio of BE to MBR)
$\eta^t_{\mathrm{res},c}$ (Reserve threshold)
$\eta^t_{\mathrm{CBR/MBR},c}$ (Resource threshold ratio of CBR to MBR)
$\mathbf{z}_c^t$ (Network conditions)
$\lambda^t_{\mathrm{CBR},c}$ (Arrival rate for CBR slice)
$\lambda^t_{\mathrm{BE},c}$ (Arrival rate for BE slice)
$\lambda^t_{\mathrm{MBR},c}$ (Arrival rate for MBR slice)
$\psi^t_{\mathrm{CBR},c}$ (Average SINR for CBR slice)
$\psi^t_{\mathrm{BE},c}$ (Average SINR for BE slice)
$\psi^t_{\mathrm{MBR},c}$ (Average SINR for MBR slice)
$A^t_{\mathrm{CBR},c}$ (CBR admission rate)
$D^t_{\mathrm{BE},c}$ (BE dropping rate)
$T^t_{\mathrm{BE},c}$ (BE average throughput)
$F^t_{\mathrm{BE},c}$ (BE Fifth-percentile throughput)
$T^t_{\mathrm{MBR},c}$ (MBR average throughput)
$A^t_{\mathrm{MBR},c}$ (MBR admission rate)
$\mathbf{y}_c^t$ (Local KPIs)
Input layer
1st dense layer
2nd dense layer
3rd dense layer
Output layer

FIGURE 6. Architecture of the ANN. The ANN maps the control parameters and the network condition vectors to the local KPIs, approximating (10).

is the j-th control parameter. Since in our definition of the reaction matrix $\hat{\mathbf{J}}$ all elements consist of 0, +1 and −1, then the element in the i-th row and j-th column of $\hat{\mathbf{J}}$ is defined as

$$\hat{j}_{i,j} = \begin{cases} 0 & -\epsilon < \dfrac{\partial y_i}{\partial x_j}/\bar{y}_i < \epsilon \\ +1 & \dfrac{\partial y_i}{\partial x_j}/\bar{y}_i > \epsilon \\ -1 & \dfrac{\partial y_i}{\partial x_j}/\bar{y}_i < -\epsilon, \end{cases} \tag{24}$$

where $\bar{y}_i$ is the target for the i-th KPI. ϵ is a threshold for determining how significant is the impact of the control parameter on the KPI. In this work we use $\epsilon = 0.05$. Note that we have normalized the approximate first derivative to the corresponding KPI target value so that the elements of $\hat{\mathbf{J}}$ are not affected by the amplitude of the approximate first derivative. With this method, at each interval t and in each cell c the reaction matrix $\hat{\mathbf{J}}^t_c$ can be dynamically created and used in (20) to adapt the control parameters. After training the neural network, we can approximately find the computational complexity of a forward pass of the neural network by counting how many multiplications, additions, and activations are performed. Given that there are 9 input parameters, three hidden layers with 50 nodes and 6 output nodes, the total number of multiplications or additions are $9\times 50+50\times 50+50\times 50+50\times 6 = 5750$. The number of ReLU activations is equivalent to the total number of nodes, i.e., $3 \times 50 + 6$. For each element of the ANN-based reaction matrix in each cell, two passes of the ANN is required. Therefore, the total number of ANN forward passes is $2 \times C \times K \times P = 2 \times 21 \times 6 \times 3 = 756$. Since the adaptation update interval (τ) is in order of minutes, the computational complexity should not hinder the implementation of the algorithm.

B. LOAD VIOLATIONS AND PRIORITIZATION

So far we have assumed that the algorithm should react to all violations from all slices. However, the responsibilities of the slice owner should be taken into account as well, i.e., if the load introduced by one of the slices is above the target load defined in the SLA, the network owner can discriminate against that slice. The reason behind this approach is that the radio resources are scarce and if one of the slices is overloading, the other slices are negatively impacted. This discrimination only needs to happen if the non-overloading slices do not meet their KPI targets. Else, if the non-overloading slices have their targets met, there is no reason to discriminate against an overloading slice.

Similar to the KPIs, there are two types of overload; local overload and global overload. At interval t, we define the global overload signal as $\mathbf{o}^t_g$ and the local overload signal in cell c as $\mathbf{o}^t_c$. These vectors are of size K. Note that there are S slices in the network and in $\mathbf{o}^t_g$ and $\mathbf{o}^t_c$ we repeat the overload signals to match the dimension of the violation vectors, i.e., K. To issue the overload signal, we rely on the network measurements regarding user arrivals, i.e., the estimated arrival rates of the Poisson processes. However, as shown in Fig. 5, instantaneous load measurements can be very noisy and unreliable for use in the control algorithm. Therefore, here we suggest using a moving average filter to cancel out noise. The filtered global and local (in cell c) load signals for slice s at time interval t are defined as

$$\begin{aligned}\grave{\lambda}^{t}_{s,g} &= \frac{1}{\Upsilon}\sum_{t'=t-\Upsilon+1}^{t}\lambda^{t'}_{s,g}\\ \grave{\lambda}^{t}_{s,c} &= \frac{1}{\Upsilon}\sum_{t'=t-\Upsilon+1}^{t}\lambda^{t'}_{s,c},\end{aligned} \tag{25}$$

where, Υ denotes the length of the moving average filter and $\lambda^{t'}_{s,g}$ denotes the instantaneous load signal. The global overload vector and local overload vector in cell c are defined as

$$\begin{aligned}\mathbf{o}^{t}_{g} &= H(\grave{\boldsymbol{\lambda}}^{t}_{g} - \overline{\boldsymbol{\lambda}}_{g}(1+\kappa))\\ \mathbf{o}^{t}_{c} &= H(\grave{\boldsymbol{\lambda}}^{t}_{c} - \overline{\boldsymbol{\lambda}}_{c}(1+\kappa)),\end{aligned} \tag{26}$$

where κ is chosen to be a small value that represents the tolerance of the network towards overload. In this work we use $\kappa = 0.05$. Now, to incorporate the overload signals in our RRM algorithm, we first have to check if KPI targets are violated for any of the non-overloading slices. To acquire the non-overloading signals, we negate (logical NOT operator) the overload signals, i.e., $\neg\mathbf{o}^{t}_{g}$ and $\neg\mathbf{o}^{t}_{c}$. Thereafter, we calculate the inner product of the violation signals and the negated overload signals to determine if non-overloading slices face KPI target violations:

$$\begin{cases}\neg\mathbf{o}^{t}_{g}\cdot\mathbf{v}^{t}_{g} = 0 & \text{No violations from non-overloading slices}\\ \neg\mathbf{o}^{t}_{g}\cdot\mathbf{v}^{t}_{g} \neq 0 & \text{Violations from non-overloading slices.}\end{cases} \tag{27}$$

Similar conditions should be checked for local overloads and violations, i.e., $\neg\mathbf{o}^{t}_{c}$ and $\mathbf{v}^{t}_{c}$. If there are KPI target violations for non-overloading slices, KPI target violations for the overloading slices are neglected, such that the non-overloading slices are prioritized. Using element-wise multiplication of vectors, the global and local violation vectors are given as

$$\begin{aligned}\dot{\mathbf{v}}^{t}_{g} &= \mathbf{v}^{t}_{g}\odot\neg\mathbf{o}^{t}_{g}\\ \dot{\mathbf{v}}^{t}_{c} &= \mathbf{v}^{t}_{c}\odot\neg\mathbf{o}^{t}_{c}.\end{aligned} \tag{28}$$

If there are no KPI target violations for non-overloading slices, we try to combat the KPI target violations in overloading slices, i.e., $\dot{\mathbf{v}}^{t}_{g} = \mathbf{v}^{t}_{g}$ and $\dot{\mathbf{v}}^{t}_{c} = \mathbf{v}^{t}_{c}$. The combined violation vector in (19) changes into

$$\tilde{\mathbf{v}}^{t}_{c} = \dot{\mathbf{v}}^{t}_{g}\odot\dot{\mathbf{v}}^{t}_{c}. \tag{29}$$

The rest of the algorithm remains as discussed before. Table 2 summarizes all of the notations introduced in the system model.

TABLE 2. Nomenclature.

Symbol	Description
C	Number of cells
S	Number of slices
K	Number of KPIs
P	Number of control parameters in each cell
$H(\cdot)$	Step function
$\mathcal{S}$	Set of all slices
λ, $\grave{\lambda}$, $\overline{\lambda}$	Actual, filtered and target arrival rates
μ	Mean of spatial distribution
σ	Std. dev. of spatial distribution
$1/\sigma$	Concentration factor
ξ	Scheduler weight
η	Admission control threshold
T, $\overline{T}$	Measured and target average throughput
A, $\overline{A}$	Measured and target admission rate
F, $\overline{F}$	Measured and target fifth-percentile throughput
D, $\overline{D}$	Measured and target Dropping rate
$\overline{G}$	Guaranteed bit rate
θ_D	Dropping threshold
R	Users' throughput
ω	Users' normalized resource share
B	Total bandwidth
Ψ	Users' average SINR
Ω_{CBR}, $\breve{\Omega}_{\text{MBR}}$	Resource share of all CBR and MBR slices
$\mathbf{y}_c$, $\mathbf{y}_g$, $\overline{\mathbf{y}}$	Local, global and target KPI vectors
$\mathbf{x}_c$, $\mathbf{X}_g$	Local and global control parameters
$\mathbf{z}_c$, $\mathbf{Z}_g$	Local and global network conditions
$\mathbf{J}$, $\hat{\mathbf{J}}$	Jacobian and reaction matrices
$\mathbf{v}_c$, $\mathbf{v}_g$, $\tilde{\mathbf{v}}_c$	Local, global and combined violation vectors
$\dot{\mathbf{v}}_c$, $\dot{\mathbf{v}}_g$	Local, global violation vectors with priorities
δ	Adaptation step size
f_c, f_g, $\hat{f}_c$	Local, global and ANN-based functions
ϵ	Control parameter impact threshold
κ	Tolerance to overload
$\mathbf{o}_c$, $\mathbf{o}_g$	Local and global overload signal
Υ	Overload filtering window length
τ	Adaptation update interval

V. SIMULATIONS AND NUMERICAL EVALUATIONS

Before evaluating the proposed slice management algorithms, we start by specifying our simulation setup. We assume $S = 3$ slices, one from each slice type, i.e. BE, CBR and MBR. To compare the performance of the different slice management algorithms and to see the effects prioritization, we introduce load anomalies from one of the slices and observe to what extent each scheme can react to these anomalies. The MBR and CBR slices are assumed to introduce as much load as they are allowed in the SLA, i.e., $\lambda_{\text{MBR}} = \overline{\lambda}_{\text{MBR}}$ and $\lambda_{\text{CBR}} = \overline{\lambda}_{\text{CBR}}$. Besides, their spatial distribution is uniform, i.e., $\sigma^2_{\text{CBR}} = \sigma^2_{\text{MBR}} = \infty$. However, the BE slice is assumed to have the load anomalies. Its introduced load λ_{BE} is higher than the target load $\overline{\lambda}_{\text{BE}}$ and the concentration factor $1/\sigma_{\text{BE}}$ is non-zero. Under a given slice management method and prioritization, we simulate different loads and different spatial

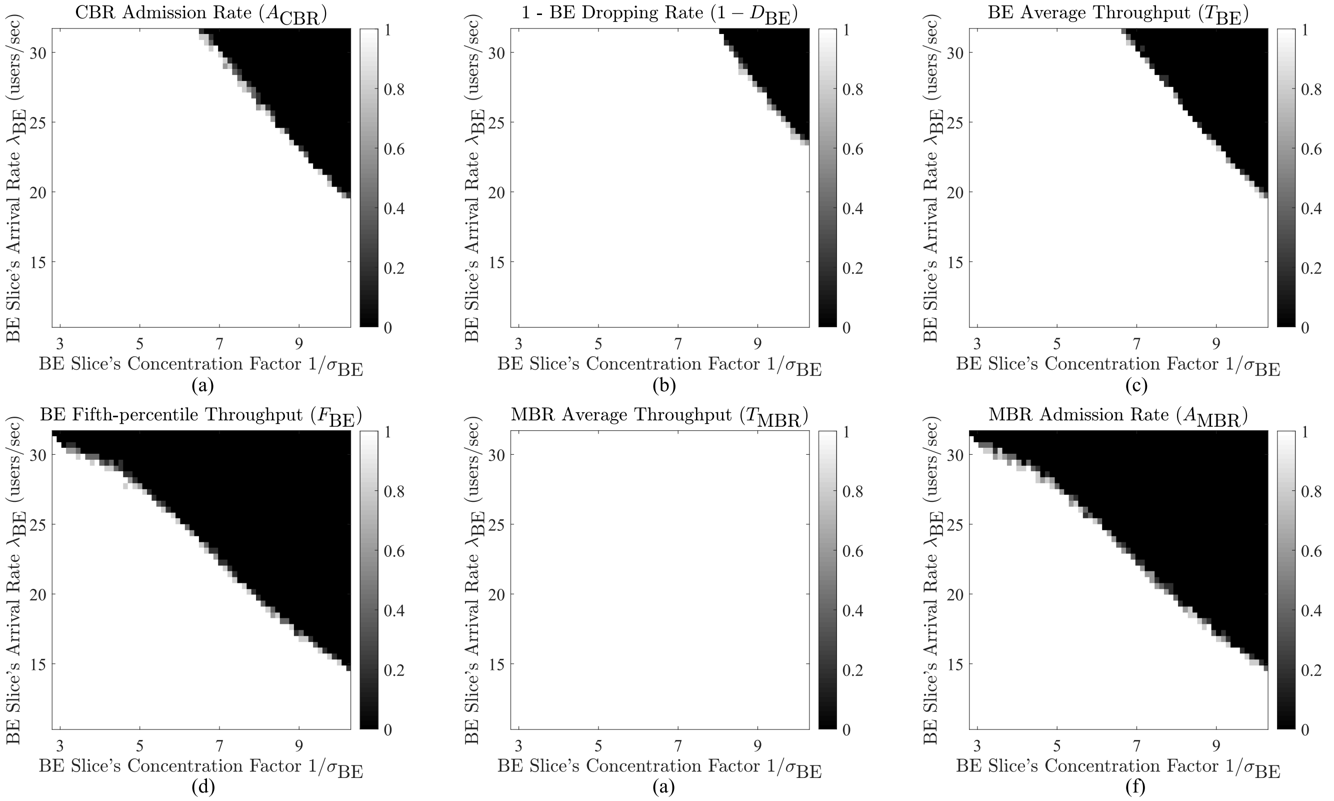


FIGURE 7. Binary fulfillment border for each KPI with heuristics-based reaction matrix and no prioritization.

concentration of the BE slice. The average global KPIs of the slices are then calculated and compared with their targets. The simulation software is MATLAB 2019b and the hardware is a high-performance computing (HPC) cluster. These simulations are done 10 times (with different random realizations) for each pixel in Fig. 7 and Fig. 8. Table 3 lists the relevant simulation parameters, which are mostly based on [38].

A. NUMERICAL EVALUATIONS

The algorithm that fulfills more KPI targets in the face of excess traffic load and high spatial concentration is deemed superior. Fig. 7 illustrates the binary fulfillment of each of the six KPIs, with the heuristics-based method and no prioritization. As the concentration factor or the introduced load of the BE slice increases, i.e., towards the top right of the figure, the KPI targets are increasingly missed. Since the KPI targets are either fulfilled or not, the fulfillment values are ∈ {0, 1}. To summarize the performance of the algorithms, we calculate the sum up of all fulfillment values for the KPI targets. This allows us to generate a fulfillment image that summarizes the performance of the different algorithms. We compare the performance of the heuristics-based and ANN-based reaction matrices. Moreover, the effect of prioritization and different lengths of moving average filters (Υ) is illustrated. The first row of Fig. 8 depicts the fulfillment images for the heuristics-based reaction matrix. Fig. 8 (a) illustrates the case without prioritization of non-overloading slices and treats all KPI target violations equally. In Fig. 8 (b), prioritization is considered, but the length of the moving average filter (Υ) is relatively short, i.e. $\Upsilon = 5$. Therefore, the system is sensitive to variations of the overload signal. However, as shown in Fig. 8 (c), increasing Υ to 60, improves the quality of

TABLE 3. Simulation parameters.

Parameter	Value
File size (f_s)	16 Mb
Adaptation 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 per sector	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 sectors	21
Number of surrounding sectors	36
Cell radius	1 km
Shadowing std. dev.	8 dB
Antenna Model	120 ° sector

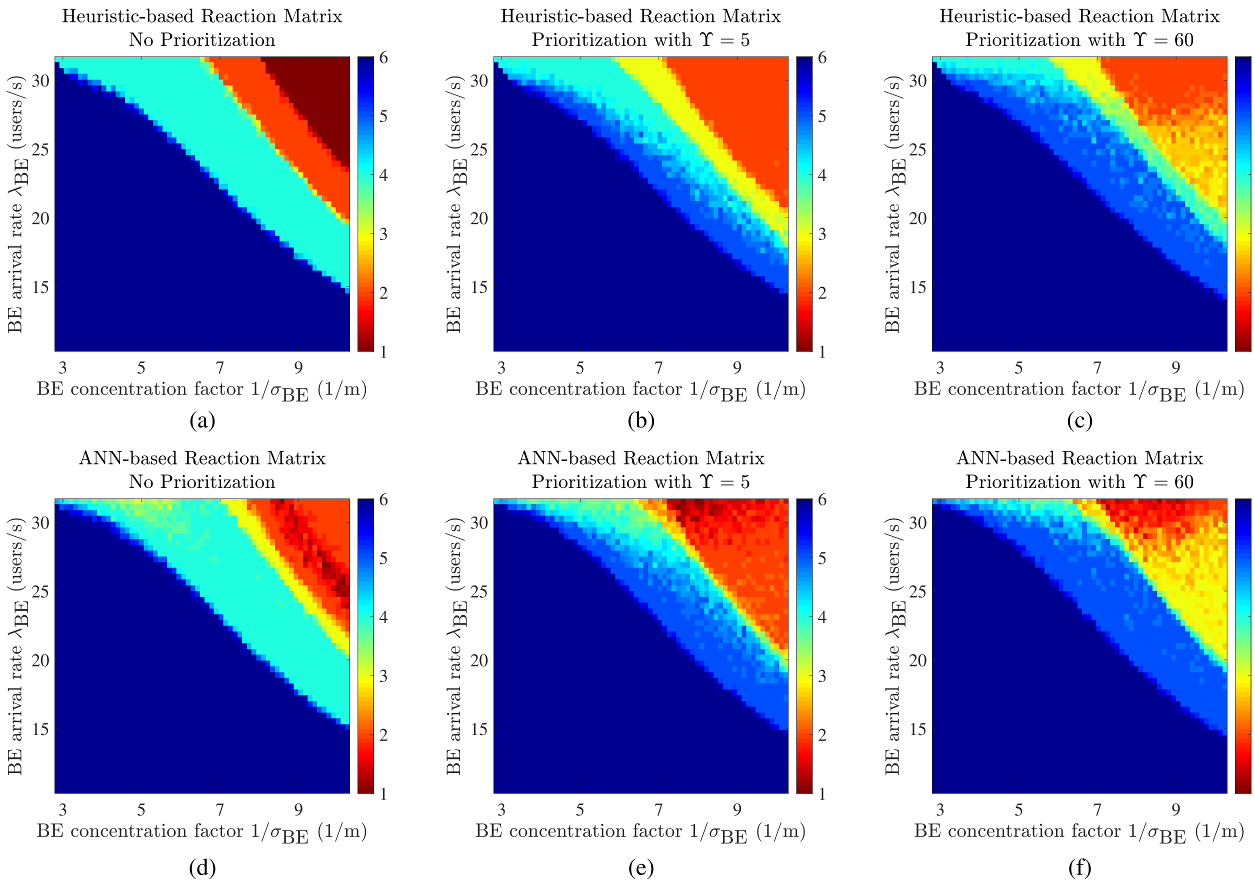
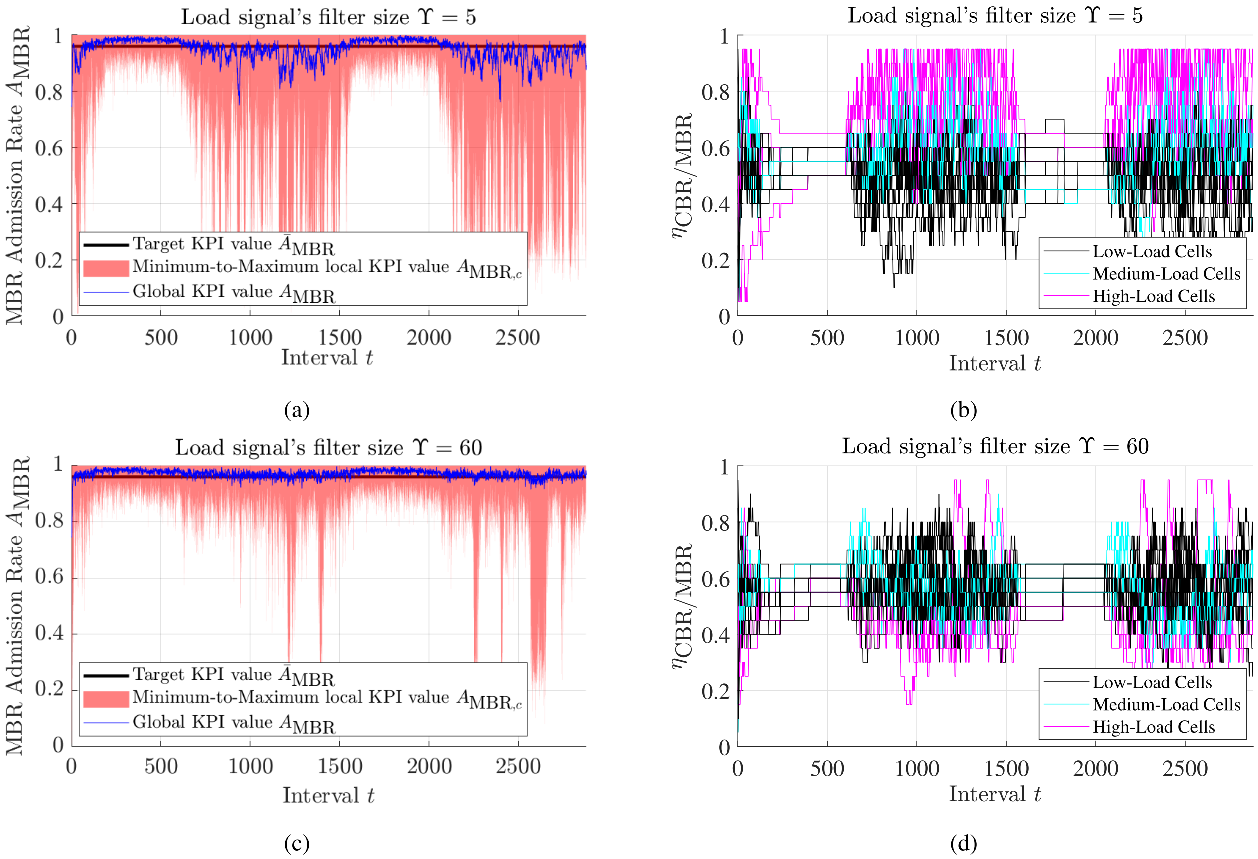


FIGURE 8. **Fulfillment border of different reaction matrices and prioritization. Top three figures (a), (b) and (c) illustrate the performance of heuristics-based reaction matrix and bottom three figures (d), (e) and (f) illustrate the performance of ANN-base reaction matrix. In figures (a) and (d) no prioritization is assumed. In (b) and (e) the prioritization mechanism is active with window length of $\Upsilon = 5$. In (c) and (f) the window length is $\Upsilon = 60$.**

the overload signal. Thus, non-overloading slices are detected more reliably and prioritized accordingly.

The second row of Fig. 8 illustrated the fulfillment border for the ANN-based reaction matrix. Compared to the heuristics-based reaction matrix, the fulfilled region (dark blue) is very similar, but within the unfulfilled region, where at least one of the KPIs cannot be fulfilled, the performance of the ANN-based method is superior, i.e., fewer KPI targets are violated on average. This is because in the feasible region, where it is possible to fulfill all of the KPIs, the violations are sparse and the control parameters are easily adjustable in response. Hence, both the heuristics- and ANN-based reaction matrices yield similar performance. However, beyond the fulfillment border, i.e. in the infeasible region, the superiority of the ANN-based method is clearly visible. The differences originate from the static vs. dynamic approximation of the Jacobian matrix, with the static approach being less accurate for many possible loads and user concentrations. Moreover, beyond the fulfillment border, violation signals are more numerous and if the reaction matrix is based on heuristics only, the numerous violation may cause an unnecessary deadlock, i.e., the control parameters do not change. This can occur when the violation of one KPI target causes an increase in some control parameters and another KPI target violations cause a decrease in the same control parameters. In such cases, if the elements of the reaction matrix are not chosen well, the corresponding control parameters remain stuck at their initial value. Similar to the heuristics-based reaction matrix, prioritization improves the fulfillment of KPI targets in the infeasible region. Moreover, increasing the length of the moving average filter makes the detection of overloading slice more accurate and the system less prone to noise. Note that in some extreme load and concentration points, the worst-case performance of heuristics-based reaction matrix is better than the ANN-based approach. In these points, the excess of BE users negatively influences the CBR and MBR slices and they issue violations. In heuristics-based matrix, as seen in Eq. (21), most of the violations of the KPIs will increase $\eta_{\text{CBR/MBR}}$ and the CBR slice is always implicitly prioritized. On the other hand, a heuristics-based matrix does not prioritize CBR slice over MBR necessarily. Therefore, based on local and instantaneous KPI violation and overload signals, one of the slices will have higher priority over the other one. However, the global KPI for both of the slices may be below



(a)

(b)

(c)

(d)

FIGURE 9. Admission rate of the MBR slice (A_{MBR}) and CBR to MBR threshold ratio ($\eta_{\mathrm{CBR/MBR}}$) in presence of daily traffic profile from the BE slice with smaller moving average window in the top row ($\Upsilon = 5$) and larger moving average window in bottom row ($\Upsilon = 60$).

the target value, since not all of the cells are changing the control parameters uniformly. Increasing the Υ further can reduce this artifact.

To better understand the dynamics of the iterative adaptive procedures and to observe the effects of more accurate prioritization, we now consider the case where the BE slice's load is changing dynamically based on the daily traffic profile in Fig. 5 (c) and the user concentration of that same slice is raised by setting the concentration factor to $1/\sigma_{\mathrm{BE}} = 8$. In Fig. 9 we show the range (over all cells) of the local MBR admission rate A_{MBR}, along with its global value over time (left column). Additionally, we depict the control parameter $\eta_{\mathrm{CBR/MBR}}$ in different cells with different levels of load (right column). The simulations are done for two values of $\Upsilon = 5$ (top row) and $\Upsilon = 60$ (bottom row). Note that there are five more KPIs and two more control parameters that we have left out for the sake of a focused discussion. The goal is to keep the global A_{MBR} above the target, even during the times when the BE slice is overloading. As we can see in Fig. 9 (a), there are some cells, in which the KPI is much below the target and consequently, the global KPI is also below its target. The reason for this behavior is observable in Fig. 9 (b), where the MBR slice threshold is decreased heavily, i.e., $\eta_{\mathrm{CBR/MBR}}$ is at its maximum in high-load cells. This is caused by the inaccuracies of the overload signal with a small filter size $\Upsilon = 5$. However, for $\Upsilon = 60$, we observe that in none of the cells $\eta_{\mathrm{CBR/MBR}}$ is set to its maximum. This highlights again that increased filtering length leads to more accurate identification of the non-overloading slices. Therefore, in the times when one of the other slices is overloading, the negative influences are not propagated to the non-overloading slices.

VI. CONCLUSION

In this paper, we have proposed a flexible framework to define and co-locate different slices with diverse KPI requirements. Since these KPIs are often different in nature, we have devised a flexible multi-objective adaptation method within the RAN slice orchestrator entity. This entity reacts to the violations of these KPIs and adapts the control parameters. This adaptation requires the knowledge of the relationship between the control parameters and the KPIs, which have been provided with a heuristics-based and ANN-based reaction matrix, respectively. Moreover, we have introduced a protection mechanism that ignores the violations of the overloading slices and prioritizes the non-overloading slices. We have shown that with the proposed algorithms and mechanisms, we can endure more load anomalies by identifying those slices that introduce traffic load conforming with the SLAs and prioritizing them.

ACKNOWLEDGEMENT

The authors would like to thank the Center for Information Services and High-Performance Computing (ZIH) at TU Dresden for generous allocations of computer time. They would also like to thank Mr. Fabian Diehm for constructive criticism of the manuscript.

REFERENCES

[1] N. Alliance, "NGMN 5G white paper, version 1.0," Next Gener. Mobile Netw. (NGMN), Frankfurt, Germany, Tech. Rep., Feb. 2015.

[2] *Study on Management and Orchestration of Network Slicing for Next Generation Network*, document TR 28.801, 3rd Generation Partnership Project (3GPP), Jan. 2018.

[3] 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 Commun. Mag.*, vol. 55, no. 5, pp. 72–79, May 2017.

[4] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing management & prioritization in 5G mobile systems," in *Proc. 22th Eur. Wireless Conf. Eur. Wireless*, May 2016, pp. 1–6.

[5] X. Zhou, R. Li, T. Chen, and H. Zhang, "Network slicing as a service: Enabling enterprises' own software-defined cellular networks," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 146–153, Jul. 2016.

[6] *IMT-2020 Deliverables*, Int. Telecommun. Union (ITU), Geneva, Switzerland, 2017.

[7] *Study on Architecture for Next Generation System*, document TR 23.799, 3rd Generation Partnership Project (3GPP), Dec. 2016.

[8] N. Alliance, "Description of network slicing concept, version 1.0," Next Generation Mobile Networks (NGMN), Frankfurt, Germany, Tech. Rep., Jan. 2016.

[9] *View on 5G Architecture, Version 2.0*, document 5G Public-Private Partnership, 2016.

[10] P. Rost, A. Banchs, I. Berberana, M. Breitbach, M. Doll, H. Droste, C. Mannweiler, M. A. Puente, K. Samdanis, and B. Sayadi, "Mobile network architecture evolution toward 5G," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 84–91, May 2016.

[11] C.-Y. Chang and N. Nikaein, "RAN runtime slicing system for flexible and dynamic service execution environment," *IEEE Access*, vol. 6, pp. 34018–34042, 2018.

[12] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80–87, May 2017.

[13] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, 3rd Quart., 2018.

[14] V. G. Nguyen and Y. H. Kim, "Slicing the next mobile packet core network," in *Proc. 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2014, pp. 901–904.

[15] T. Taleb, M. Corici, C. Parada, A. Jamakovic, S. Ruffino, G. Karagiannis, and T. Magedanz, "EASE: EPC as a service to ease mobile core network deployment over cloud," *IEEE Netw.*, vol. 29, no. 2, pp. 78–88, Mar. 2015.

[16] Z. A. Qazi, M. Walls, A. Panda, V. Sekar, S. Ratnasamy, and S. Shenker, "A high performance packet core for next generation cellular networks," in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 348–361. [Online]. Available: https://doi.org/10.1145/3098822.3098848

[17] P. Marsch, I. Da Silva, O. Bulakci, M. Tesanovic, S. E. El Ayoubi, T. Rosowski, A. Kaloxylos, and M. Boldi, "5G radio access network architecture: Design guidelines and key considerations," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 24–32, Nov. 2016.

[18] *Study on New Radio Access Technology: Radio Access Architecture and Interfaces*, document TR 38.801, 3rd Generation Partnership Project (3GPP), Mar. 2017.

[19] *Study on New Radio Access Technology Radio Interface Protocol Aspects*, document TR 38.804, 3rd Generation Partnership Project (3GPP), Mar. 2017.

[20] *Policy and Charging Control Architecture*, document TS 23.203, 3rd Generation Partnership Project (3GPP), Dec. 2019.

[21] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, 1st Quart., 2015.

[22] *Network Sharing; Architecture and Functional Description*, document TS 23.251, 3rd Generation Partnership Project (3GPP), Jul. 2020.

[23] A. Khan, W. Kellerer, K. Kozu, and M. Yabusaki, "Network sharing in the next mobile network: TCO reduction, management flexibility, and operational independence," *IEEE Commun. Mag.*, vol. 49, no. 10, pp. 134–142, Oct. 2011.

[24] L. Doyle, J. Kibilda, T. K. Forde, and L. DaSilva, "Spectrum without bounds, networks without borders," *Proc. IEEE*, vol. 102, no. 3, pp. 351–365, Mar. 2014.

[25] M. I. Kamel, L. B. Le, and A. Girard, "LTE wireless network virtualization: Dynamic slicing via flexible scheduling," in *Proc. IEEE 80th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2014, pp. 1–5.

[26] A. Aijaz, "$\mathsf{Hap-SliceR}$: A radio resource slicing framework for 5G networks with haptic communications," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2285–2296, Sep. 2018.

[27] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing in 5G: An auction-based model," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[28] D. Zhang, Z. Chang, and T. Hamalainen, "Reverse combinatorial auction based resource allocation in heterogeneous software defined network with infrastructure sharing," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–6.

[29] O. Narmanlioglu, E. Zeydan, and S. S. Arslan, "Service-aware multiresource allocation in software-defined next generation cellular networks," *IEEE Access*, vol. 6, pp. 20348–20363, 2018.

[30] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Perez, "Network slicing games: Enabling customization in multi-tenant networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[31] A. Ksentini and N. Nikaein, "Toward enforcing network slicing on RAN: Flexibility and resources abstraction," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 102–108, Jun. 2017.

[32] D. Marabissi and R. Fantacci, "Heterogeneous public safety network architecture based on RAN slicing," *IEEE Access*, vol. 5, pp. 24668–24677, 2017.

[33] M. Hu, Y. Chang, Y. Sun, and H. Li, "Dynamic slicing and scheduling for wireless network virtualization in downlink lte system," in *Proc. 19th Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, Nov. 2016, pp. 153–158.

[34] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: A substrate for virtualizing wireless resources in cellular networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1333–1346, Oct. 2012.

[35] D. Marabissi and R. Fantacci, "Highly flexible RAN slicing approach to manage isolation, priority, efficiency," *IEEE Access*, vol. 7, pp. 97130–97142, 2019.

[36] G. Sun, K. Xiong, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and W. Jiang, "Autonomous resource provisioning and resource customization for mixed traffics in virtualized radio access network," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2454–2465, Sep. 2019.

[37] G. Sun, G. T. Zemuy, and K. Xiong, "Dynamic reservation and deep reinforcement learning based autonomous resource management for wireless virtual networks," in *Proc. IEEE 37th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Nov. 2018, pp. 1–4.

[38] *Evolved Universal Terrestrial Radio Access (E-UTRA); Further Advancements for E-UTRA Physical Layer Aspects*, document TR 36.814, 3rd Generation Partnership Project (3GPP), Mar. 2017.

[39] A. Abdel 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 Trans. Veh. Technol.*, vol. 60, no. 4, pp. 1612–1625, May 2011.

[40] M. Castaneda, M. T. Ivrlac, J. A. Nossek, I. Viering, and A. Klein, "On downlink intercell interference in a cellular system," in *Proc. IEEE 18th Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 2007, pp. 1–5.

[41] 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 *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, Jun. 2018, pp. 1–6.

[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

BEHNAM KHODAPANAH (Graduate Student Member, IEEE) received the M.Sc. degree in electrical engineering and information technology from RWTH Aachen, in 2015. He is currently pursuing the Ph.D. degree with the Vodafone Chair Mobile Communications System, Technical University of Dresden. His current research interests are wireless systems, resource management, and application of machine learning in those areas.

AHMAD AWADA (Member, IEEE) received the M.S. degree in communication engineering from the Technical University of Munich, in 2009, and the Ph.D. degree from the Technical University of Darmstadt, Germany, in 2014. He joined Nokia Networks in 2013. Since 2016, he has been working for the Radio Access and Architecture Munich Department, Standardization Research Laboratory, dealing with LTE and 5G standardization research. His research interests include radio transmission schemes, radio resource management and control, and network slicing.

INGO VIERING (Member, IEEE) received the Dr. Ing. degree from the University of Ulm, in 2003, and the Dipl.Ing. degree from the University of Technology Darmstadt, in 1999. He is currently the Co-Founder and CEO of Nomor Research GmbH, Munich, Germany. Furthermore, he is also an honorary professor at the Technical University of Munich. He has filed more than 150 patents, published more than 100 scientific papers, and he is actively contributing to 3GPP.

ANDRÉ NOLL BARRETO (Senior Member, IEEE) received the M.Sc. degree from Catholic University (PUC-Rio), Rio de Janeiro, Brazil, in 1996, and the Ph.D. degree from Technische Universität Dresden, Germany, in 2001, both in electrical engineering. After several positions in academia and industry in Switzerland (IBM Research) and Brazil (Claro, Nokia Technology Institute/INDT, Universidade de Brasília, Ektrum), he joined Barkhausen Institut, Dresden, Germany, in 2018. He was the Chair of the Centro-Norte Brasil Section of IEEE in 2013/2014 and the General Co-Chair of the Brazilian Telecommunications Symposium in 2012. He is currently researching wireless communications for a reliable, resilient, and secure Internet of Things.

MERYEM SIMSEK (Senior Member, IEEE) received the Dipl.Ing. degree in EE and IT and the Ph.D. degree in reinforcement-learning-based ICIC in LTE-advanced HetNets from the University of Duisburg-Essen, in 2008 and 2013, respectively. In 2013, she was a postdoctoral scientist at Florida International University. She has been a research group leader at the Technical University of Dresden, since 2014 and joined the International Computer Science Institute Berkeley, in 2016. Her main research interests include wireless systems and machine learning.

GERHARD FETTWEIS (Fellow, IEEE) received the Ph.D. degree from RWTH Aachen, under the supervision of H. Meyr. After one year at IBM Research, San Jose, CA, USA, he moved to TCSI, Berkeley, CA, USA. Since 1994, he has been a Vodafone Chair Professor at the Technical University of Dresden. Since 2018, he has been heading the Barkhausen Institute. He researches wireless transmission and chip design, coordinates two DFG centers (cfaed and HAEC), the 5GLab Germany, has spun out 16 startups, and is a member of two German academies: (Sciences) "Leopoldina" and (Engineering) "acatech."