RAN Slicing in Multi-MVNO Environment Under Dynamic Channel Conditions

Darshan A. Ravi, Vijay K. Shah, Chengzhang Li, *Graduate Student Member, IEEE*, Y. Thomas Hou, and Jeffrey H. Reed, *Fellow, IEEE*

***Abstract*—With the increasing diversity in the requirement of wireless services with guaranteed Quality of Service (QoS), radio access network (RAN) slicing becomes an important aspect in implementation of next-generation wireless systems (5G). RAN slicing involves the division of network resources into many logical segments where each segment has specific QoS and can serve users of the mobile virtual network operator (MVNO) with these requirements. This allows the network operator (NO) to provide service to multiple MVNOs each with different service requirements. Efficient allocation of the available resources to slices becomes vital in determining the number of users and therefore, the number of MVNOs that a NO can support. In this work, we study the problem of the modulation and coding scheme (MCS)-aware RAN slicing (MaRS) in the context of a wireless system having MVNOs which have users with minimum data rate requirement. Channel quality indicator (CQI) report sent from each user in the network determines the MCS selected, which in turn determines the achievable data rate. But the channel conditions might not remain the same for the entire duration of a user being served. For this reason, we consider the channel conditions to be dynamic where the choice of the MCS level varies at each time instant. We model the MaRS problem as a NonLinear Programming problem and show that it is NP-Hard. Next, we propose a solution based on the greedy algorithm paradigm. We then develop an upper performance bound for this problem and finally evaluate the performance of the proposed solution by comparing it against the upper bound under various channel and network configurations.**

***Index Terms*—5G and beyond networks, dynamic channel conditions, performance bound, radio access network (RAN) slicing.**

I. Introduction

WITH the advent of the Internet of Things (IoT), the number of devices accessing the Internet has been increasing exponentially. Ericsson has estimated that about 5 billion IoT devices will be connected to the Internet and about 2.6 billion 5G subscriptions by the end of 2025 [1]. Efficient utilization of available spectrum resources becomes vital to accommodate this growth. Adding to this requirement is the complexity of users having varied Quality of Service (QoS) requirements.

To address this complexity, the radio access network (RAN) slicing technology has been widely adopted by several industrial communities [2], [3]. With the help of RAN slicing, operators can perform service customization, isolation, and multitenancy support on common physical network infrastructure by enabling logical as well as physical separation of network resources [4]. This multitenancy support enables network operators (NOs) to support multiple mobile virtual NOs (MVNOs) in the form of a slice. The Third-Generation Partnership Project (3GPP) has identified network slicing as one of the key technologies to achieve varied performance requirements—such as high throughput, high-security goals in 5G networks [5].

One of the key features of RAN slicing is that—MVNOs are assigned slices that are independent of one another [6]. That is, the allocation of the radio resources is up to the NO who can allocate them at will, based on the QoS requirement while ensuring complete isolation between slices. The NO we consider is based on software defined-RAN (SD-RAN) controller architecture comprising of a Slice Manager and MVNO specific scheduler. The NO architecture is formally introduced in Section III.

Once the QoS requirements for each MVNO are collected by NO, the core problem lies in the allocation of scarce spectrum resources such that each MVNO's QoS requirement is met for all its users. We consider spectrum resources as resource block (RB). This is a difficult problem because over-provisioning of RB for a user, will result in wastage, and under-provisioning might not meet the QoS requirements. Therefore, the design of an efficient slicing algorithm to meet each MVNO user's requirement is a key for optimal usage of RB. Also, from a business standpoint, optimal usage of RB which will result in the increased number of users served in a time slot and thereby increased number of MVNOs supported by a fixed number of RB is of great interest.

One of the factors which influence the slicing decision is the channel condition experienced by the RB during its path toward the users. To convey the channel information, each user in the network sends a channel quality indicator (CQI) report back to the NO. Often in real-world scenarios, the channel conditions do not remain the same. They keep varying with respect to time and frequency. In order to take into account this dynamic channel condition, the users send the CQI report in

Manuscript received March 1, 2021; revised July 10, 2021; accepted August 19, 2021. Date of publication August 27, 2021; date of current version March 7, 2022. This work was supported in part by DARPA under Grant HR0011-19-C-0096; in part by the Virginia Commonwealth Cyber Initiative (CCI); and in part by the Virginia Tech Institute for Critical Technology and Applied Science (ICTAS). *(Corresponding author: Darshan A. Ravi.)*

Darshan A. Ravi, Chengzhang Li, Y. Thomas Hou, and Jeffrey H. Reed are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: darshan19@vt.edu; licz17@vt.edu; thou@vt.edu; reedjh@vt.edu).

Vijay K. Shah is with the Cybersecurity Engineering Department, George Mason University, Fairfax, VA 22030 USA (e-mail: vshah22@gmu.edu).

Digital Object Identifier 10.1109/JIOT.2021.3108145

regular intervals with its periodicity determined by the NO and in between this interval, the channel conditions are assumed to remain same [7]. To remain close to reality, we consider dynamic channel conditions in our work.

We illustrate the problem of RAN slicing under dynamic channel conditions by considering the minimum data rate per time slot for each user as a specification by MVNOs. Calculation of data rate for a user at a given time depends on the modulation and coding scheme (MCS) level chosen for the user by the NO at that time. Choice of MCS level in turn depends on the CQI report sent from the users of MVNO. Now, the problem we are addressing in this article is, how do we create a channel conditions aware slice for each MVNO such that, the maximum number of MVNO user's minimum data rate requirement is met.

From the IoT applications standpoint, this problem is crucial to support the scalability and diverse requirement of IoT devices under a resource crunch situation. We can envision a unique slice created in the RAN, specific for IoT to support a specific IoT application. In this article, we have shown how the slice can be created when an IoT application requirement is the minimum data rate.

Even though the RAN resource allocation issue has been studied extensively in the recent past [6], [8], [9], the problem of resource allocation to MVNOs under dynamic channel conditions is relatively new. This is discussed more in Section II. The design of efficient resource allocation/slicing enforcement algorithm is not trivial and is met with unique challenges.

1) *Users Maximization:* Meeting the minimum data rate requirement for the maximum number of MVNO users in the slice time slot. This can be achieved by choosing the optimal number of RB and the MCS level for each user.
2) *Orthogonality:* Each RB should be allocated to only one user across all MVNOs at a given time slot to avoid interference [10]–[12].

This work aims to design, analyze, and validate the MCS-aware RAN slicing (MaRS) algorithm that takes into consideration the challenges mentioned above. To summarize, this work makes the following contributions.

1) We formulate the MaRS problem as a Nonlinear Programming Optimization problem in Section V using the model developed in Section III. We will also prove the NP Hardness of the MaRS problem.
2) We propose a solution for this problem using the greedy algorithm paradigm in Section VI.
3) We develop an upper performance bound for the MaRS algorithm in Section VII.
4) We provide an implementation of the proposed solution and carry out an exhaustive evaluation in Section VIII.

II. Related Works

There has been significant work to address the problem of RAN slicing, especially in the recent past. There have been many excellent surveys on this topic [4], [10], [12]–[14]. The authors in these surveys provide comprehensive information regarding the work being done on this topic. Additionally, a book has been published on the topic of RAN slicing where many slicing algorithms have been proposed [15]. Specifically [13] covers the advancements in RAN slicing which is based on the software-defined network (SDN) architecture. The architecture considered in our work loosely follows the work covered in [13].

In the recent works, the RAN slicing problem [16]–[18] has been dealt by designing solutions using various theoretical means optimization [19], [20], game theory [21]. There have also been many advancements where several machine learning approaches have been used to address the RAN slicing problem—{Reinforcement Learning [22], Deep Learning [23]–[25]}. These machine learning approaches are not suitable for deployment due to their huge data requirements for training and the time it requires to do so. Moreover, accurate predictions of the channel conditions are required to make the slicing algorithm effective.

One of the key limitations of these works is that it does not show the actual deployment of RAN slices on top of a physical network. Although D'Oro *et al.* [9] discussed RAN slicing policies and enforcement problems by considering fine-grained control of resources, it falls short when we bring in dynamic channel conditions. Moreover, the problem formulation considers slice as allocation of a certain percentage of RBs from a given pool without considering the underlying requirement for these slices.

One of the works which closely focuses on addressing the RB allocation problem is [11]. The authors propose an RB partitioning algorithm that focuses on allocating RB to every MVNO by simultaneously maximizing the percentage of satisfied MVNOs while allocating the minimum amount of RB. However, the problem in [11] does not take into account the dynamic channel conditions.

Our work can be closely compared to [8]. Papa *et al.* [8] addressed the problem of RAN slicing by considering dynamic channel conditions in an SD-RAN-based architecture. One of the key architectural differences between our work and [8] is the flexibility offered to the MVNO in the SD-RAN architecture. Papa *et al.* [8] considered individual Slice Managers for each slice but a common scheduler for all the users. This provides very little flexibility for MVNOs. In our work, we consider an independent scheduler for all MVNOs. This allows MVNO, the option of choosing its users for scheduling at each time interval. Section III discusses this in detail.

Korrai *et al.* [26] addressed the RAN slicing problem for multiplexing eMBB and URLLC slices. Although this article [26], considers the MCS selection in the design of the slicing algorithm, it again falls short in providing MVNO the flexibility in scheduling as the architecture considered is completely different.

This article [27], [28] present a framework for LTE virtualization. The authors propose an architecture for virtualizing the LTE base stations (called eNodeB in LTE architecture) with the objective of having different operators sharing the same physical resources. The solution is based on a hypervisor (as in CPU virtualization), which hosts different virtual nodes, allocates the resources, and is responsible for spectrum sharing and data multiplexing. In [29], the framework from [27]

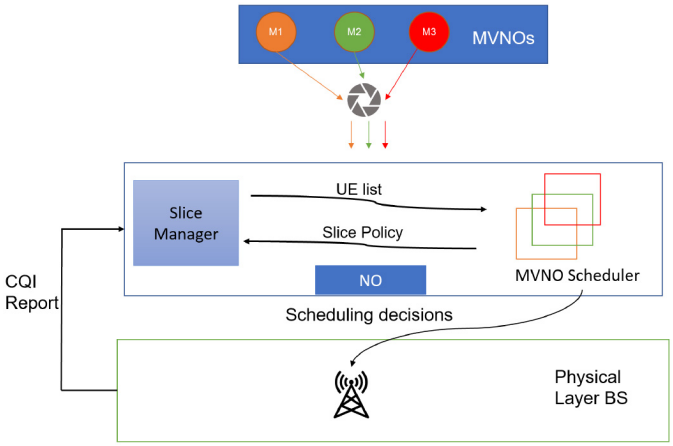

Fig. 1. SD-RAN slicing architecture.

and [28] is used to present an algorithm for scheduling physical RB for the virtual nodes. The main idea of this algorithm is that if an eNodeB is overloaded and a neighbor eNodeB has available resources, a user is selected to be migrated to the unloaded eNodeB. Although the concept of centralized control is similar to our work, the problem statement is completely different. In our work, we are addressing the problem of RAN slicing in a multi-MVNO environment as opposed to resource sharing.

In summary, our work addresses the shortcomings of these papers by providing more flexibility to the MVNOs, developing an efficient slicing algorithm with dynamic channel conditions, and carrying out a thorough validation.

III. System Model

We consider an NO administering a single 5G RAN base station B and set of $\mathcal{M} = \{1, 2, \ldots, \mathrm{M}\}$ MVNOs as depicted in Fig. 1. The NO serves the MVNOs by creating virtual RAN slices built on top of the underlying physical network B. We split NO functionally into Slice Manager and MVNO scheduler. This architecture lies in line with 5G RAN concepts, where the management and orchestration are implemented as an SDN. We adopt the architecture principle similar to [8]–[30], and include additional features to aid the proposed slicing procedure.

Once the NO collects the minimum data rate slice request from all MVNOs, it creates an instance of MVNO scheduler for each MVNO in the network. We define Λ_m^i as the minimum data rate requirement for each user i of MVNO $\forall i \in m, m \in \mathcal{M}\ \forall m$. MVNO Scheduler for all $m \in \mathcal{M}$ provides a scheduling order of users belonging to m, $\mathcal{U}_m$, to the Slice Manager. The Slice Manager, which has the CQI information for each user in the network, dynamically assigns the resources on B to each MVNO slice based on this scheduling order sent by the MVNO. The advantage of this architecture is that it leaves the choice of scheduler implementation, up to the MVNO. Each MVNO may employ a unique scheduling algorithm.

Since the BS follows the 5G cellular technology, spectrum resources are organized as grids of RB, that span across both time and frequency domains [31]. Each RB represents the minimum Spatio-temporal scheduling unit. Considering N_{RB} and T as the number of available subcarriers and temporal slots, respectively, the set of available RB is $|R_b| = N_{RB} \times T$ in the physical RAN network for a certain bandwidth.

Implication of Time Slot T: Theoretically, the time slot T can range from 1 TTI(t) to 1000's of TTIs, depending upon how dynamically the Slice Manager wishes to operate resource slicing policy. Under realistic consideration and in lieu with next-generation O-RAN architecture [32], it is expected that the slicing manager will either reside in nonreal-time ran intelligent controller (RIC) or near-real-time RIC, which are, respectively, in order of $>$ 1s and (10–1000 ms) time scales [33]. Thus, in our work, we consider that T will be a large value, in the range of several milliseconds. Further, we consider the user's minimum data rate requirement is defined per time slot T.

Dynamic Channel Conditions: We consider the channel conditions to be dynamic in nature and may vary in frequency and time, but remain consistent within the time slot T. This is similar to aperiodic CSI reporting [7]. Depending on the channel condition obtained from CQI reports for users of the MVNOs being served, the Slice Manager determines a suitable MCS for transmission depending on each MVNO user's minimum data rate requirement, out of 29 MCS levels as per 5G 3GPP specification [31]. Let $\mathcal{C}$ denote the set of available MCS, i.e., $\mathcal{C} = \{0, 1, \ldots, 28\}$. The MCS determines how much information (in bits) is modulated and coded in each RB by the BS. The higher the MCS is, the higher the modulation and coding rate is. That means, the maximum amount of information that can be transmitted on one RB also depends on the channel conditions. If the channel condition is poor and the NO uses a high MCS, then the information carried in the RB will not be successfully received and decoded. Therefore, the achievable data rate by an RB depends on both the MCS level chosen by the NO as well as the channel condition for this RB.

Let $q_{u_m^i}^{r,t}$ denotes the maximum MCS that can be used for a certain RB r to serve a user $u_m^i \in \mathcal{U}_m$ such that the information carried in RB can be successfully received by the user at TTI $t \in T$

$$1 \leq q_{u_m^i}^{r,t} \leq |C|.$$

Let $v^{c,t}$ denote the modulation and coding rate for an RB under MCS $c \in \mathcal{C}$ and $d_{u_m^i}^{r,c,t}$ denote the maximum achievable data rate by RB r for the user u_m^i under MCS $c \in \mathcal{C}$ at time $t \in T$. If $c \leq q_{u_m^i}^{r,t}$, the transmission would be successful and the achievable data rate is $v^{c,t}$. Otherwise, the transmission would be unsuccessful and the data would be lost. That is

$$d_{u_m^i}^{r,c,t} = \begin{cases} v^{c,t}, & \text{if } c \leq q_{u_m^i}^{r,t} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Finally, the NO also imposes a restriction on maximum throughput allowed per MVNO slice, namely, $\overline{\Lambda_m}$, depending on channel conditions or business requirements [34]. This restriction prevents any individual MVNO slice from overloading the network. An intuitive way of selecting Λ_m^i may

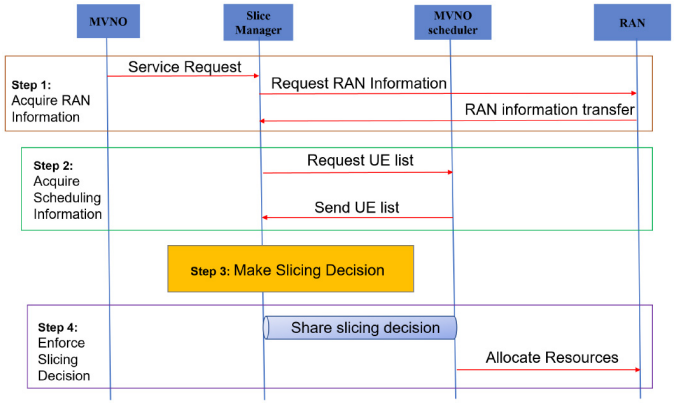

Fig. 2. SD-RAN slicing architecture workflow diagram.

be from a pure business perspective, i.e., whichever MVNO pays the most will get higher throughput. However, in general, the choice of maximum throughput in a multi-MVNO, limited resources environment introduces a new problem that is out of the scope for this work.

IV. SD-RAN Workflow

Before we proceed with designing the slicing algorithm, it is important to understand how different components in the SD-RAN architecture interact with each other to serve users of MVNO. In this section, we present the workflow for our SD-RAN architecture in Fig. 1.

The NO communicates with several components before it assigns resources to a specific user of the MVNO. After an MVNO requests services from an NO, the task of allocating resources can be broadly divided into four steps.

Step 1 (Acquiring RAN Information): After an MVNO submits a request for a service to the Slice Manager of the NO, the Slice Manager acquires the RAN information. This RAN information contains the number of users in the network for the MVNO, the channel conditions experienced by each user, and the available resources in terms of RB in the network to serve the MVNO. In Fig. 2, the base station and the users in the network are represented by a single RAN block.

Step 2 (Acquiring Scheduling Information): An instance of the scheduler is created in the NO for each MVNO that requests a service. It is up to the MVNO on how the scheduling algorithm is implemented. For example, a particular MVNO may use round-robin and other MVNO might opt for priority scheduling. It is one of the novelties in our work where we provide the MVNO, the flexibility of choosing the scheduling algorithm. To make a slicing decision, the Slice Manager interacts with the instance of the MVNO scheduler to acquire the scheduling list which is a list of users and its unique minimum data rate requirement that is generated through the MVNO specific scheduling algorithm.

Step 3 (Making Slicing Decision): After step 2, the Slice Manager has all the required information to make a slicing decision. It has the list of users that it needs to serve with its minimum data rate requirement, its channel conditions, and the available resources in the network to serve them. Now, the Slice Manager limits the number of users that can be served for an MVNO by imposing an upper bound of maximum throughput allowed per slice. Using the MCS-aware RAN slicing Algorithm, the Slice Manager makes a slicing decision by assigning resources to the users of the multiple MVNOs across time T.

Step 4 (Enforcing Slicing Decision): After the Slice Manager makes the slicing decision for time slot T, it is conveyed to the MVNO scheduler and enforced on RAN. The MVNO scheduler can use this slicing decision as an input to generate the scheduling list for the next time slot T.

V. Problem Formulation

In this section, we formulate the MaRS problem as a Nonlinear Optimization problem. The problem aims at determining the optimal set of RBs to be allocated to each MVNO $m \in \mathcal{M}$ in time slot T, such that the maximum number of users can be served across MVNOs in T, by considering—1) MVNO's minimum bit rate requirement is met for each of its users; 2) each MVNO scheduler's unique user scheduling order is ensured; and 3) the total throughput per slice does not exceed the maximum allowable throughput set by the NO for that MVNO slice.

Notation: Let set $\mathcal{U}_m = \{u_m^1, \ldots, u_m^j, \ldots, u_k^{|U_m|}\}$ denote the scheduling order of all users belonging to MVNO $m \in \mathcal{M}$.

Decision Variables: Let u_m^i denotes whether a user i belonging to MVNO m can be served by the Slice Manager. Let $x_m^{r,i,t}$ denote the whether a certain RB $r \in R$ is allocated to any user u_m^i in MVNO m at TTI $t \in T$. Let $y_{i,m}^{c,t}$ denote whether an MCS level c is chosen by a user u_m^i at TTI t

$$P1: \max_{x_m^{r,i,t}, y_{i,m}^{c,t}} \sum_{m \in M} \sum_{u_m^i \in \mathcal{U}_m} u_m^i \tag{2}$$

$$\sum_{u_m^i \in U_m} x_m^{r,i,t} \leq 1 \quad \forall r, t \tag{3}$$

$$c \times y_{i,m}^{c,t} \leq c_{max}^{r,i,t} x_m^{r,i,t} \quad \forall t, i, m, r, c \tag{4}$$

$$\sum_{c} y_{i,m}^{c,t} \leq 1 \quad \forall m, i, t \tag{5}$$

$$u_m^i \geq u_m^j \quad \forall i < j \quad \forall m \tag{6}$$

$$\sum_{t \in T} \sum_{r \in R} x_m^{r,i,t} y_{i,m}^{c,t} d_{u_m^i}^{r,c,t} \geq u_m^i \Lambda_m^i \quad \forall m, i \tag{7}$$

$$\sum_{t \in T} \sum_{r \in R} \sum_{u_m^i \in U_m} x_m^{r,i,t} y_{i,m}^{c,t} d_{u_m^i}^{r,c,t} \leq \overline{\Lambda_m} \quad \forall m \tag{8}$$

$$x_m^{r,i,t}, y_{i,m}^{c,t}, u_m^i \in \{0, 1\}. \tag{9}$$

The maximization problem given in (2) targets to accommodate the maximum number of users to satisfy the constraints. Constraint (3) indicates that an RB can be allocated to one UE at any given time. Equation (4) indicates that MCS chosen for a user cannot be greater than maximum MCS supported by any RB r at that time. Moreover, (5), ensures that a single MCS level is chosen for a user at time t. Constraint (6) ensures that scheduling order determined by the MVNO scheduler is maintained in allocating resources. Equation (7) meets the minimum data rate requirement for each user belonging to an MVNO. Equation (8) indicated the maximum data rate

TABLE I
NOTATION TABLE

Symbol	Definition
$\mathcal{M}$	A set of MVNOs requesting slices from NO.
Λ_m^i	Minimum data rate req. for user i in MVNO $m \in \mathcal{M}$.
$\overline{\Lambda_m}$	Maximum allowable throughput for a slice $m \in \mathcal{M}$.
$\mathcal{U}_m$	Scheduling List for MVNO $m \in \mathcal{M}$.
$\mathcal{C}$	A set of possible MCS values as per 3GPP specifications.
u_m^i	Represents a UE i belonging to MVNO $m \in \mathcal{M}$.
$v^{c,t}$	The modulation and coding rate for an RB under MCS $c \in \mathcal{C}$ at time $t \in T$.
$q_{u_m^i}^{r,t}$	The maximum MCS that can be used for a certain RB r to user belonging to MVNO $m \in \mathcal{M}$ at time $t \in T$.
$d_{u_m^i}^{r,c,t}$	The maximum achievable data rate by RB r for a UE u_m^i under MCS $c \in \mathcal{C}$ at time $t \in T$.
$c_{max}^{r,i,t}$	Maximum mcs that can be selected for a RB r for user i at time $t \in T$.
$\mathcal{L}^T$	Slicing List - List of users to be scheduled across MVNOs at time T.
C_{max}	Maximum MCS that can be selected for a user at any TTI $t \in T$.
h^t	Maximum achievable data rate for a user at each tti t.
$\tilde{c}$	MCS used to achieve maximum data rate at each TTI h^t.
A^t	List to hold maximum data rate for each user for every TTI $t \in T$.
R_{tot}	Total available RB in the network.
$\hat{R}$	RB that have been already allocated in T.
R'	RB that contribute to achieve maximum data rate h^t at each TTI $t \in T$.
c'	MCS value for R' that achieve the maximum data rate h^t.
$R*$	Total RBs used to meet data rate requirements all the users in time slot T.
$C*$	MCS used for all the RB in $R*$.
U	Users served in time slot T.

achieved by the resources allocated to MVNO is under the maximum allowable throughput for MVNO. Table I highlights all the notations used in this paper.

Theorem 1: The MaRS problem is NP-Hard.

Proof: In order to prove the NP Hardness, consider the optimization problem defined in (2) for a single MVNO and for a single time slot $t \in T$. Therefore, we drop the m and t notation. Further, we consider channel condition is the same across all base stations (and RBs), then the MCS level for all the RBs will be the same, $c \in \mathcal{C}$. This affects (7) and (8). Therefore, we can rewrite the optimization problem and the constraints as follow:

$$P2: \max_{x^{r,i}} \sum_{u^i \in \mathcal{U}} u^i \tag{10}$$

$$\sum_{r \in R} x^{r,i} \geq u^i \tag{11}$$

$$\sum_{r \in R} x^{r,i} \leq \overline{\Lambda_m} \tag{12}$$

$$x^{r,i}, u^i \in \{0, 1\}. \tag{13}$$

Notice that P2 is a maximum coverage problem, which is a classic NP-Hard problem [35]. Since the MaRS problem can be modeled as a maximum coverage problem, the MaRS problem is also an NP-Hard problem. ■

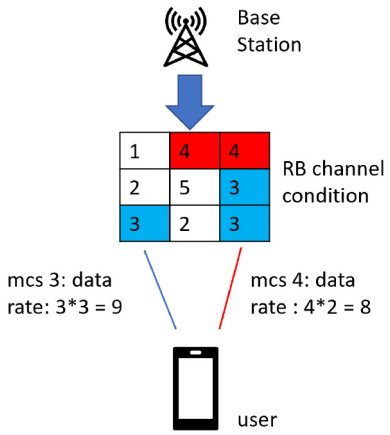

Fig. 3. Example for MCS selection.

VI. MCS-AWARE RAN SLICING ALGORITHM

In this section, we develop the MCS-aware RAN slicing Algorithm based on a greedy paradigm.

A. Key Intuitions Behind the Proposed Algorithm

The design of the MaRS algorithm is based on the following key intuitions.

Intuition 1: From the MaRS problem objective function [See (2)], it is obvious that we need to maximize the number of users that can be served in T for each MVNO $m \in \mathcal{M}$. We consider the minimum data rate requirement to be per time slot T and we say that a user u_m^i is served only if it is allocated sufficient RBs such that its minimum data rate Λ_m^i is met in T. Based on this observation, we should minimize the number of RBs utilized to serve each user.

Intuition 2: We sort the users across MVNOs in increasing order based on their minimum data rate requirement Λ_m^i. We call it as Slicing List $\mathcal{L}^T$. Even though each user can have its own minimum data rate requirement, we must follow the scheduling order defined by the MVNO (6). That is, for some MVNO m if the scheduling order is u_m^1, u_m^2, we must always allocate resources to u_m^1 first even if $\Lambda_m^1 > \Lambda_m^2$. This ensures that in a case of insufficient RBs to support all users in T, the user which is first in the scheduling order gets higher priority than other users. However, we do not maintain any scheduling order across MVNOs. That is, for any MVNOs m, n, the Slicing List can be $\mathcal{L}^T = \{\Lambda_m^1, \Lambda_m^2, \Lambda_n^1\}$ if $\Lambda_m^2 < \Lambda_n^1$.

Intuition 3: To incorporate the channel conditions in the slicing decision, we must consider the effect of the MCS selection on RB. In Fig. 3, we use an example to show the dependency between MCS selection and the number of RB. Suppose a user is requesting a data rate of eight from a base station which has nine RBs. The channel conditions for each of the RB is denoted in their respective grid position. If the BS chooses MCS 3 for transmission, three RBs are required to meet the user's data rate requirement as $3 \times 3 = 9$. If BS chooses MCS 4 for transmission, the user's data rate requirement can be met by just two RB as $2 \times 4 = 8$. Therefore, choosing the higher MCS reduces the RB utilization to meet the data rate requirement. From the previous ideas, we know that we must

Algorithm 1 Slicing List

1: Collect scheduling order and minimum data rate requirement Λ_m^i for each user.
2: Generate a tuple for each user which contains MVNO id, scheduling order, minimum data rate $< m, i, \Lambda_m^i >, \forall i, m$.
3: Add all users to the list $\mathcal{L}^T = [< m, i, \Lambda_m^i >], \forall i, m$.
4: **Sort** $\mathcal{L}^T$ based on Λ_m^i.
5: **Sort** $\mathcal{L}^T$ based on i.
6: return $\mathcal{L}^T$

use the least amount of resources to serve users to maximize the number of users served. This implies we must choose the maximum MCS for each user at any given time.

Intuition 4: The slicing decision is an iterative approach wherein, we allocate the subset of unallocated RBs based on its MCS level to a user of MVNO at each iteration. The slicing decision is controlled by two main factors, the minimum data rate requirement for each user Λ_m^i and the maximum allowable throughput decided by the NO for each MVNO, $\overline{\Lambda_m}$. Eventually, the algorithm exits when all the users have been served or when all RB are allocated.

B. Algorithm Details

In this section, we discuss how we utilize the MCS levels on the RB in making the slicing decisions.

Recall that the first step in our algorithm is the generation of the Slicing List $\mathcal{L}^T$. This depends on:

1) the minimum data rate requirement for each user Λ_m^i $\forall u_m^i, m$;
2) the scheduling list sent by each MVNO $\mathcal{U}_m$ $\forall m$.

Using this information, the Slice Manager develops $\mathcal{L}^T$ which is valid for time slot T by two-stage sorting, as shown in Algorithm 1.

With the Slicing List $\mathcal{L}^T$ as the input, we present the MCS-aware RAN slicing algorithm in Algorithm 2. The algorithm outputs the least number of RBs and their MCS level in the time slot T such that each user's minimum data rate requirement is met.

As discussed in the previous section, the algorithm uses an iterative approach wherein at each iteration, it serves a user according to $\mathcal{L}^T$. This algorithm consists of two key steps.

Step 1 (Finding the Optimal Number of RBs and Their MCS Level Which Maximizes the Achievable Data Rate at Each TTI $t \in T$): This is addressed by iterating over the MCS values that a user can support, followed by iterating over each TTI. Remember, the achievable data rate at each TTI is directly related to the MCS level chosen for its RBs. Therefore, in our algorithm, we iterate over each TTI, starting with the maximum MCS C_{max}, first to calculate the data rate. We keep track of the maximum achievable data rate by updating h^t after each iteration of MCS $\tilde{c}$.

Step 2 (Greedily Allocate the RBs for Each User Such That Their Requirement Is Met): Once we have the list containing the maximum achievable data rate and the corresponding RBs with the MCS value for each TTI A^t, we now allocate the resources to the user in T. Our key idea is to minimize the number of RBs for each user which will subsequently help us in maximizing served users in T. As discussed in the previous section, to minimize the RB utilization, we need to choose the higher MCS. Following this idea, we follow a greedy approach wherein we choose the RBs with maximum MCS first in A^t to meet the minimum data rate requirement for each user. This enables us to choose the least amount of RBs and corresponding MCS at each TTI $t \in T$, such that each user's requirement Λ_m^i is met for time T.

Algorithm 2 MCS-Aware RAN Slicing Algorithm

Input: Slicing list $\mathcal{L}^T$, Λ_m^i minimum bit rate requirement for each user belonging to mvno m, v^c achievable bit rate with MCS level $c \in C$, maximum allowable throughput for a MVNO $\overline{\Lambda_m}, \forall m$, the maximum mcs that can be supported by a resource block at TTI t, $q^{r,t}$.
Output: Set of allocated RBs R^* and MCS level C^*, for each user in $\mathcal{L}^T$.

1: Initialize $R^* = \phi$ and $C^* = \phi$
2: Initialize already allocated RBs, $\hat{R} = \phi$
3: Total RBs, R_{tot}
4: **for** each user, u_m^i in $\mathcal{L}^T$ **do**
5: current data rate for each user, $d_m^i = \phi$
6: current data rate for each MVNO, $d_m = \phi$
7: **if** $d_m \geq \lambda_m$ **then**
8: **break**
9: **for** each MCS, $c = C_{max}, \ldots, 1$ **do**
10: list to hold each TTI information, $A^t = \phi$
11: **if** $\hat{R} \cap R_{tot} = \phi$ **then**
12: return **"No solution"**
13: **for** $t = 0, \ldots, T$ **do**
14: maximum data rate at tti t $h^t = \phi$
15: **for** $\tilde{c} = C_{max}, \ldots, c$ **do**
16: candidate RB for MCS $\tilde{c}$, $R_{can} = \phi$
17: **for** $r \in R_{tot}$ **do**
18: **if** $r \cap \hat{R} = \phi$ and $q^{r,t} \geq \tilde{c}$ **then**
19: $R_{can} = R_{can} \cup r$
20: **if** $R_{can} \times v^c > h^t$ **then**
21: $h^t = R_{can} \times v^c$
22: $R' = R_{can}, c' = \tilde{c}$
23: add tuple $\delta_i^t =< c', R', h^t >$ to the list A^t
24: **sort** A^t based on decreasing order of c'
25: **for** each tuple δ_i^t in A^t **do**
26: $h_u = h_u + \delta_i^t[h^t]$
27: $R_u = R_u \cup \delta_i^t[R']$
28: $c_u = c_u \cup \delta_i^t[c']$
29: **if** $h_u \geq \Lambda_m^i$ **then**
30: $R^* = R^* \cup R_u$, $\hat{R} = \hat{R} \cup R^*$
31: $C^* = C^* \cup c_u$
32: $U = U + 1$
33: **break**
34: return U, R^* and C^*

C. Time Complexity

We will now discuss the complexity of the MaRS algorithm. To compute the maximum data rate for a user at each TTI, the time complexity is $O(|C||R_{\text{tot}}|)$. In Algorithm 2, this is calculated using the *for* loops on lines 4 and 9. We need to compute this for the each TTI $t \in T$. This is calculated using the *for* loop on line 13. Therefore, the total time complexity to compute maximum data rate for a user in time T is, $O(|T||C||R_{\text{tot}}|)$ (lines: 4–23). After that, we sort maximum

data rate achieved across TTIs. The sorting operation in represented in line 24 which has the complexity of $O(|T||\log T|)$. Now, we iterate over each element in the sorted list to meet the data rate requirement, $O(|T|)$. This is done by the *for* loop on line 25. Then, the MaRS algorithm allocates the optimal resources R_u and chooses its MCS c_u for each user across TTI if possible for the current MCS c (lines: 29–33). Therefore, the total complexity for each iteration of c is $O(|T||C||R_{\text{tot}}|) + O(|T||\log T|) + O(|T|) = O(|T||C||R_{\text{tot}}|)$. Since there are $|C|$ possible values of c, the complexity is $O(|T||C|^2|R_{\text{tot}}|)$. Now, the MaRS algorithm calculates this for every user in $|L^T|$. Therefore, the total time complexity of the MaRS algorithm is $O(|L^T||T||C|^2|R_{\text{tot}}|)$.

Theorem 2: If there exists a feasible solution for any given user in $\mathcal{L}^T$, Algorithm 2 will find it.

Proof: As discussed in the previous section, for each user, Algorithm 2, calculates the maximum achievable data rate for each TTI $t \in T$. The algorithm goes over every possible combination of the RBs and MCS to determine this data rate. Once the algorithm generates A^t, the maximum achievable data rate for a user at each TTI, it proceeds with RB allocation. Now, as long as the sum of the data rates in A^t is greater than the minimum user data rate requirement, the algorithm provides a solution. That is, given a finite set of unallocated RB and a user u_m^i with a minimum data rate requirement of Λ_m^i

$$\sum_{\forall \delta_i^t \in A^t} \delta_i^t\left[h^t\right] \geq \Lambda_m^i \tag{14}$$

the algorithm will find subset of RB and corresponding MCS across T which meet Λ_m^i as long as (14) is met. ■

VII. Performance Bound

As proven in Theorem 1, the MaRS problem is NP-hard and it is not feasible to find a polynomial-time optimal solution. Therefore, it is vital to develop an upper bound for the objective function defined in (2). This upper bound can be used as a benchmark to measure the performance of the scheduling algorithm that we presented in Section V.

When R, the maximum number of RBs in time T is given, our problem aims to find a subset of R for each user. Choice of this subset depends on c, the MCS selected for them. Note that, if we want to maximize (2), we need to find a subset that contains the least amount of RBs.

Since we want to find an upper bound for objective function in (2), let us consider a fictitious scenario of excellent channel condition for every user in time T. Therefore, every RB $r \in R$, can support the maximum MCS value that a user can support during its allocation. That is

$$q_{u_m^i}^{\max} = \max_{r,t}\ q_{u_m^i}^{r,t}. \tag{15}$$

We further consider that the MVNO scheduler always uses the maximum MCS for each user. That is

$$d_{u_m^i}^{r,c,t} = q_{u_m^i}^{\max}. \tag{16}$$

We then proceed with the allocation of the RBs for each user following the Slicing List, $\mathcal{L}^t$. In this fictitious scenario, the data rate achievable for each user in time T, is directly proportional to the number of RBs allocated to it at time T. Therefore, we can rewrite constraints (7) and (8) as

$$\sum_{t\in T}\sum_{r\in R} x_m^{r,i,t} \times d_{u_m^i}^{r,c,t} \geq u_m^i \Lambda_m^i \quad \forall m, i \tag{17}$$

$$\sum_{t\in T}\sum_{r\in R}\sum_{u_m^i \in U_m} x_m^{r,i,t} \times d_{u_m^i}^{r,c,t} \leq \overline{\Lambda_m} \quad \forall m. \tag{18}$$

We can see that the criterion to meet each user's minimum data rate requirement completely depends on the number of RBs allocated to it. Since we assume the maximum MCS level for each RB, any allocation of the RB with MCS $d_{u_m^i}^{r,c,t}$, for a user to meet Λ_m^i would use the least amount of RBs. Therefore, if we use the least amount of RBs for every user, we can find the maximum users that can be supported by the given set of RBs R for time slot T.

In this section, we developed a very intuitive-based upper performance bound for the MaRS algorithm. In the following section, we perform simulations of this upper bound and compare the performance of the MaRS algorithm with it.

VIII. Performance Evaluation

In this section, we assess the performance of the MaRS algorithm proposed in Section VI. We evaluate the MaRS algorithm in terms of its ability to achieve our objective function of maximizing the users served by varying various 5G network parameters. We use the upper bound developed in Section VII as the benchmark for this purpose.

A. Network Setting

We simulate a 5G NR base station deployed in a certain environment serving $\mathcal{N}$ number of users. This BS and user deployment can be modeled using any standard approaches, such as hexagonal, square lattice, or stochastic geometry-based Poisson point process [36]. We consider this BS to be operating as a frequency division duplexing (FDD) system with a channel bandwidth of 20 MHz, which is divided into 1200 subcarriers organized into R_{tot} =100 RB. while considering subcarrier spacing of 15 Khz. Each PRB represents the minimum scheduling unit and consists of 12 subcarriers and 14 symbols.

For each user in the network, the expected channel condition (in terms of MCS) is randomly chosen. For each MCS c, the modulation and the coding rate $v^{c,t}$ is obtained from [37].

Configurable Parameters: There are many configurable system parameters, such as the time slot T, number of MVNO M, users of each MVNO u_m^i, the minimum data rate for each user λ_m^i, the maximum throughput allocated per MVNO slice by NO Λ_m. We evaluate the performance of the MaRS algorithm under the various combination of these settings.

The number of users served in the time slot T would be the sole performance metric for all our simulation settings.

B. Results

In this section, we evaluate the proposed algorithm against the upper bound against varying parameters.

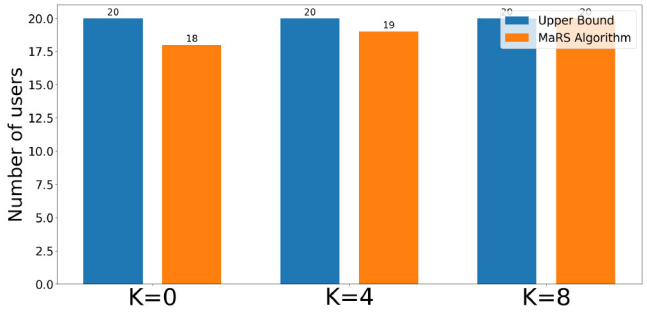

Fig. 4. MaRS algorithm Performance under different Rician factors.

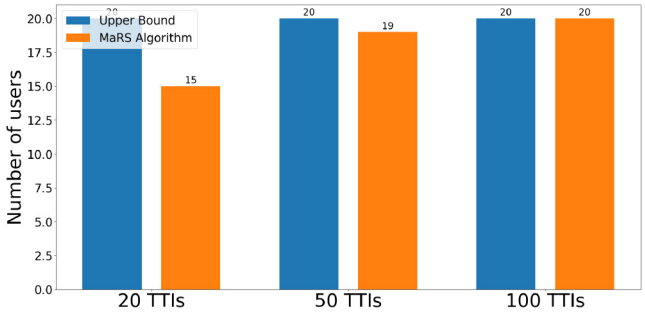

Fig. 5. MaRS algorithm performance under different Time Slot T.

TABLE II
SIMULATION PARAMETERS—VARYING CHANNEL PROPAGATION

Time Slot, T	5
Number of MVNOs, M	2
Number of users per MVNO, u_m^i	10
Minimum Throughput required per user, Λ_m^i	50 Mb/Slot
Maximum allowed throughput per MVNO slice, $\overline{\Lambda_m}$	500 Mb/slot

TABLE III
SIMULATION PARAMETERS—VARYING SLICE SLOT TIME T

T	20,50,100
M	2
u_m^i	10
Λ_m^i	100 Mb/Slot
$\overline{\Lambda_m}$	5 Gb/slot

Varying Channel Propagation: We first evaluate the performance of the MaRS algorithm under channels with varying LOS signal strength. We assume the Rician fading channel with no frequency and time correlation.

Fig. 4 compares the performance of the MaRS algorithm against the upper bound for different Rician factor K. The configuration used for the experiments is listed in Table II.

Under this configuration, we can see that the MaRS algorithm can achieve near-optimal performance. In particular, when the Rician factor $K = 0$ (i.e., the Rayleigh fading), 4 and 8, the number of users served by the MaRS algorithm is within 10% of the respective upper bound. For $K = 8$, the performance of the MaRS algorithm is as good as the upper bound. This can be attributed to the higher availability of resources that can be allocated to the fewer number of users.

Varying Time Slot T: We now evaluate the MaRS algorithm by varying the slice time slot. Increasing the time slot T increases the available resources to meet slice requirements per T. Therefore, for this experiment, we increase the minimum data rate requirement for each user while also increasing the maximum available throughput. We consider the Rayleigh fading to model the channel and generate MCS values for each user. Table III shows the configuration used.

Fig. 5 shows the performance of the MaRS algorithm in comparison with the upper bound. Clearly increasing the number of available resources, increases the performance of the MaRS algorithm. This is evident in Fig. 5 for $T = 100$ TTIs, where the MaRS algorithm catches up with the upper bound in terms of the number of users served across MVNOs.

TABLE IV
SIMULATION PARAMETERS—THREE NETWORK SCENARIOS

	Scenario 1	Scenario 2	Scenario 3
T	50	50	50
M	3	2	3
u_m^i	15	10	5
Λ_m^i	100 Mb/Slot	100 Mb/Slot	50 Mb/Slot
$\overline{\Lambda_m}$	5 Gb/slot	5 Gb/slot	5 Gb/slot

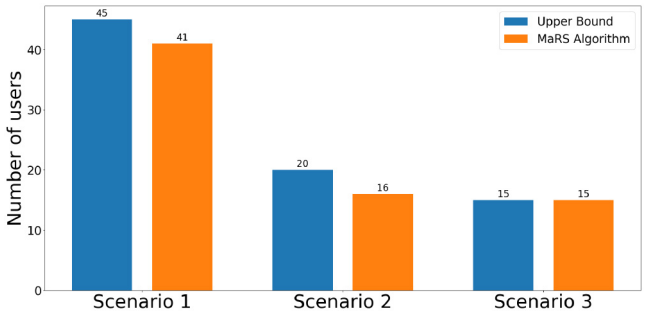

Fig. 6. MaRS algorithm performance comparison for three scenarios.

Varying Other Simulation Parameters: We now vary other system parameters to evaluate the MaRS algorithm performance. We understand the behavior of the MaRS algorithm by considering three scenarios of the network configurations as shown in Table IV for our simulations. We assume Rayleigh fading channels for all the simulations.

In Fig. 6, Scenario 1 represents a network scenario where there are many users with high minimum data rate requirements and few resources to allocate them. Here, we can see that the MaRS algorithm is within 5% of the upper bound. In Scenario 2, we decrease the load on the base station by reducing the number of MVNOs and users. Even in this case, we can see the MaRS algorithm achieves near-optimal performance. Finally, in Scenario 3, where the number of RBs is plenty, we see that the MaRS algorithm performs as well as the upper bound.

Varying User Data Rate Requirement: Further, we varied the data rate requirement for each user in the network under these three scenarios (Fig. 7). We choose a random data rate requirement for each user between 10 and 150 Mb/slot, the results obtained are similar to the previous case where the data rate is fixed.

Fast Changing Channel: Until now, we have considered time correlation for each user in the network where the channel

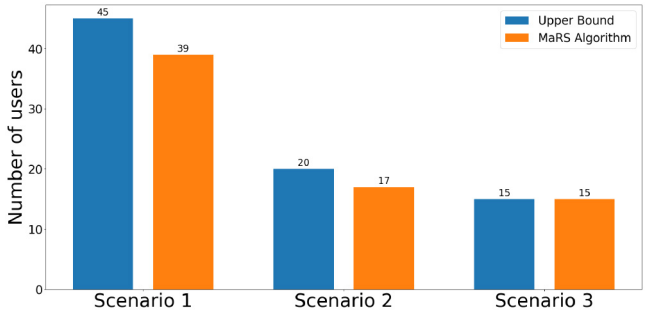

Fig. 7. MaRS algorithm performance comparison for three scenarios with random data rate for each user.

TABLE V
SIMULATION PARAMETERS—FAST CHANGING CHANNEL

T	20,50,100
M	2
u_m^i	30
Λ_m^i	10 Mb/Slot
$\overline{\Lambda_m}$	250 Mb/slot

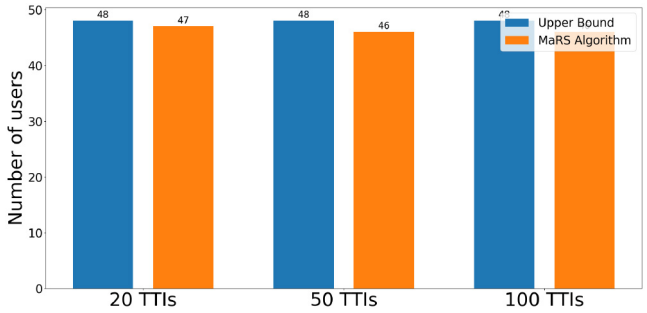

Fig. 8. MaRS algorithm performance for fast changing channels.

conditions remain constant for each user in time slot T. We now consider a network scenario where the channel conditions for each user change at each TTI. We still assume Rayleigh fading channels with no frequency correlation. Table V shows the settings used for this evaluation.

Fig. 8, represents the obtained results. We can see that the MaRS algorithm performance is within 5% of the upper bound. As mentioned in the earlier section, we have developed the MaRS algorithm and evaluated its performance for near real-time and nonreal-time configuration of the RIC in O-RAN architecture. By demonstrating that the MaRS algorithm's performance is near optimal, we can say the MaRS algorithm is a viable option for deployment for nonreal-time and near-real-time RIC.

RB Utilization: Finally, we evaluate the performance of the MaRS algorithm in terms of the number of RB utilized to serve the users across all MVNOs in the networks. We say a user is served when its minimum data rate is met at time slot T. We measure the number of RB utilized to serve users in three scenarios presented earlier under different MCS selection criterion. MCS selection criterion.

1) *Maximum MCS:* We assume that each RB in T for a user can support the maximum MCS.

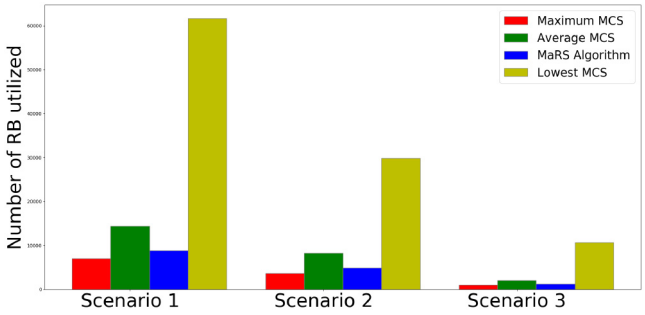

Fig. 9. Comparison of the MaRS algorithm against static allocation algorithms for RB utilization.

2) *Average MCS:* We calculate the average MCS level for a user across T and assume that each RB in T supports this average value.
3) *Lowest MCS:* We calculate the lowest MCS level for a user across T and assume that each RB in T can only support the lowest value.

Fig. 9 shows the obtained results. It is evident that the maximum MCS selection criteria use the least amount of RBs to serve users. This is understandable as we assume the best channel conditions for all RBs. But, there may be significant retransmissions which would increase latency. However, The performance of the MaRS algorithm outperforms the average MCS and lowest MCS selection criteria. There is a significant decrease in the number of RBs used to serve the users using the MaRS algorithm when compared to these criteria. Therefore, using the MaRS algorithm we can serve more users in a time slot T than using the lowest MCS and average MCS static algorithms.

IX. CONCLUSION

In this article, we investigated the problem of RAN slicing in a multi-MVNO environment with varied users having minimum data rate requirements as a specification for the users. First, we discussed the SD-RAN architecture and discussed its operation flow. Then, we formulated the MCS-aware RAN slicing (MaRSP) problem as an optimization problem with an objective function to increase the number of supported users at each time slot T. We proved that the MaRSP problem is NP-Hard. Next, we developed the novel MaRS algorithm where we maximize the data rate for each user at each TTI and assign resources to it based on a greedy paradigm. We also showed that the MaRS algorithm has polynomial time complexity. Following that, we developed an upper performance bound for the MaRS algorithm by considering no frequency and time correlation. Finally, we carry out a thorough evaluation of the MaRS algorithm under various network and channel scenarios. Results conclude that the proposed slicing algorithm achieves near-optimal performance when compared with the upper bound. Through various simulation settings, we have also shown that the MaRS algorithm is easily scalable. In compliance with the O-RAN architecture, we have seen through results that the MaRS algorithm can be applied to nonreal-time and near-real-time RIC deployments. Using the RB utilization as a metric, we have compared the performance of the MaRS algorithm with other static allocation algorithms. We see that

the MaRS algorithm outperforms many static allocation algorithms by using the least amount of RBs to serve the minimum data rate requirement for each user. As promising directions for future works, we can investigate how the upper limit for the MVNO can be identified and verify the applicability of the MaRS algorithm for uplink communication.

REFERENCES

[1] Ericsson Incorporated. (Nov. 2019). *Ericsson Mobility Report*. [Online]. Available: https://www.ericsson.com/4acd7e/assets/local/mobility-report/documents/2019/emr-november-2019.pdf

[2] "Technical report: 5G core vision," Ltd Samsung Electron. Co., Suwon-si, South Korea, White Paper, 2019. [Online]. Available: https://image-us.samsung.com/SamsungUS/samsungbusiness/pdfs/5G_Core_Vision_Technical_Whitepaper.pdf

[3] I. da Silva *et al.*, "Impact of network slicing on 5G radio access networks," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Athens, Greece, 2016, pp. 153–157.

[4] 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.

[5] X. Li *et al.*, "Network slicing for 5G: Challenges and opportunities," *IEEE Internet Comput.*, vol. 21, no. 5, pp. 20–27, Sep. 2017.

[6] S. D'Oro, F. Restuccia, and T. Melodia, "Toward operator-to-waveform 5G radio access network slicing," 2019. [Online]. Available: arXiv:1905.08130.

[7] A. Tchiumento, M. Bennis, C. Desset, L. Van der Perre, and S. Pollin, "Adaptive CSI and feedback estimation in LTE and beyond: A Gaussian process regression approach," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, p. 168, Jun. 2015. [Online]. Available: https://doi.org/10.1186/s13638-015-0388-0

[8] A. Papa, M. Klugel, L. Goratti, T. Rasheed, and W. Kellerer, "Optimizing dynamic RAN slicing in programmable 5G networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, 2019, pp. 1–7.

[9] S. D'Oro, F. Restuccia, A. Talamonti, and T. Melodia, "The slice is served: Enforcing radio access network slicing in virtualized 5G systems," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Paris, France, 2019, pp. 442–450.

[10] A. Kaloxylos, "A survey and an analysis of network slicing in 5G networks," *IEEE Commun. Stand. Mag.*, vol. 2, no. 1, pp. 60–65, Mar. 2018.

[11] C.-Y. Chang, N. Nikaein, and T. Spyropoulos, "Radio access network resource slicing for flexible service execution," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Honolulu, HI, USA, 2018, pp. 668–673.

[12] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 166–174, Oct. 2017.

[13] A. A. Barakabitze, A. Ahmad, A. Hines, and R. Mijumbi, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Comput. Netw.*, vol. 167, Feb. 2020, Art. no. 106984.

[14] M. Chahbar, G. Diaz, A. Dandoush, C. Cérin, and K. Ghoumid, "A comprehensive survey on the E2E 5G network slicing model," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 49–62, Mar. 2021.

[15] R. Wen and G. Feng, *Robust RAN Slicing*. West Sussex, U.K.: Wiley, 2021, pp. 189–208.

[16] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 127–140.

[17] K. Samdanis, S. Wright, A. Banchs, A. Capone, M. Ulema, and K. Obana, "5G network slicing—Part 1: Concepts, principles, and architectures," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 70–71, May 2017.

[18] R. Ferrus, O. Sallent, J. Perez-Romero, and R. Agusti, "On 5G radio access network slicing: Radio interface protocol features and configuration," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 184–192, May 2018.

[19] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017.

[20] T. Ma, Y. Zhang, F. Wang, D. Wang, and D. Guo, "Slicing resource allocation for eMBB and URLLC in 5G RAN," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–11, Jan. 2020.

[21] A. Gudipati, L. E. Li, and S. Katti, "Radiovisor: A slicing plane for radio access networks," in *Proc. Workshop Hot Topics Softw. Defined Netw.*, Aug. 2014, pp. 237–238.

[22] W. Wu *et al.*, "Dynamic RAN slicing for service-oriented vehicular networks via constrained learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2076–2089, Jul. 2021.

[23] M. H. Abidi *et al.*, "Optimal 5G network slicing using machine learning and deep learning concepts," *Comput. Stand. Interfaces*, vol. 76, Jun. 2021, Art. no. 103518.

[24] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard, "DeepSlice: A deep learning approach towards an efficient and reliable network slicing in 5G networks," in *Proc. IEEE 10th Annu. Ubiquitous Comput. Electron. Mobile Commun. Conf. (UEMCON)*, New York, NY, USA, Oct. 2019, pp. 0762–0767.

[25] H. Xiang, S. Yan, and M. Peng, "A realization of fog-ran slicing via deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2515–2527, Apr. 2020.

[26] P. Korrai, E. Lagunas, S. K. Sharma, S. Chatzinotas, A. Bandi, and B. Ottersten, "A RAN resource slicing mechanism for multiplexing of eMBB and URLLC services in OFDMA based 5G wireless networks," *IEEE Access*, vol. 8, pp. 45674–45688, 2020.

[27] Y. Zaki, L. Zhao, C. Goerg, and A. Timm-Giel, "LTE mobile network virtualization," *Mobile Netw. Appl.*, vol. 16, pp. 424–432, Aug. 2011.

[28] Y. Zaki, L. Zhao, C. Goerg, and A. Timm-Giel, "LTE wireless virtualization and spectrum management," in *Proc. WMNC*, Budapest, Hungary, 2010, pp. 1–6.

[29] M. Li *et al.*, "Investigation of network virtualization and load balancing techniques in LTE networks," in *Proc. IEEE 75th Veh. Technol. Conf. (VTC Spring)*, Yokohama, Japan, 2012, pp. 1–5.

[30] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, "FlexRAN: A flexible and programmable platform for software-defined radio access networks," in *Proc. 12th Int. Conf. Emerg. Netw. EXp. Technol.*, 2016, pp. 427–441.

[31] *5G; NR; Physical Channels and Modulation (3GPP TS 38.211 Version 15.4.0 Release 15)*, ETSI Standard TS 38 211, 2018.

[32] *O-RAN: Towards an Open and Smart RAN*, ORAN Alliance, Alfter, Germany, 2018.

[33] "ORAN-WG2.AIML.v01.00 O-RAN working group 2 AI/ML workflow description and requirements," ORAN Alliance, Alfter, Germany, Rep. ORAN-WG2.AIML.v01.00, 2019.

[34] *NG.116 Generic Network Slice Template V5.0*, GSMA, London,U.K., Jun. 2021.

[35] J. R. Rice, "Complexity of computer computations (Raymond E. Miller and James W. Thatcher, Eds.)," *SIAM Rev.*, vol. 16, no. 3, pp. 407–409, 1974.

[36] S. M. Yu and S.-L. Kim, "Downlink capacity and base station density in cellular networks," in *Proc. 11th Int. Symp. Workshops Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, Tsukuba, Japan, 2013, pp. 119–124.

[37] *5G;NR;Physical Layer Procedures for Data; (3GPP TS 38.214 Version 15.3.0 Release 13)*, ETSI Standard TS 138 214, 2018.