SlicedRAN: Service-Aware Network Slicing Framework for 5G Radio Access Networks

Behnam Ojaghi, Ferran Adelantado
Wireless Networks Research Lab (WINE)
Universitat Oberta de Catalunya (UOC)
Castelldefels, Spain
{bojaghi, ferranadelantado}@uoc.edu

Angelos Antonopoulos, Christos Verikoukis
Centre Tecnologic de Telecomunicacions de Catalunya (CTTC/CERCA)
Castelldefels, Spain
{aantonopoulos, cveri}@cttc.es

***Abstract*—5G mobile networks are envisioned to substantiate new vertical services with diverse performance requirements. Slicing in the Radio Access Network (RAN) promises an efficient solution for these diversified needs of 5G networks, which foresees the separation of the Base Station (BS) functionality between the Central Unit (CU) and the distributed Remote Radio Heads (RRHs). In this paper, we formulate a Mixed Integer Programming (MIP) framework that maximizes the throughput by jointly selecting the optimal Functional Split (FS) and the routing path from a connected User Equipment (UE) to the CU, while satisfying the agreed Service Level Agreements (SLAs) of each service. Furthermore, we propose an effective heuristic, SlicedRAN, which creates isolated RAN slices premised on the service requirements connected through a Fronthaul/Backhaul (FH/BH) network and obtains near-optimal solutions in a short computing time compared to the MIP framework. Our results show that there is a trade-off between the architecture of the FH/BH network and the minimum SLA of each slice, which provides a solution to efficiently design a virtualized network infrastructure. According to the results, the SlicedRAN outperforms existing State-of-the-Art (SoA) up to 112% gain in throughput. Results are shown close to the optimal results, with a loss below 5%.**

***Index Terms*—5G, Functional split, Crosshaul, RAN slicing, Network virtualization**

I. Introduction

The fifth-generation (5G) of mobile communications is designed to serve various types of demanding services with extremely different Quality of Service (QoS) requirements. The International Telecommunication Union (ITU) categorizes 5G mobile network services into three main types [1]: i) Enhanced Mobile Broadband (eMBB) is the set of services that need higher-bandwidth, such as High Definition (HD) videos, Virtual Reality (VR), and Augmented Reality (AR), ii) ultra-Reliable and Low Latency Communications (uRLLC) characterizes the range of services demanding low latency and more reliable mobile services, such as industrial Internet, remote surgery, assisted or automated driving, and iii) massive Machine Type Communications (mMTC) is designated for the services that require high connection density though with relaxed latency and throughput requirements, such as smart city and smart agriculture applications. It has been largely proven in the literature that the traditional *one-size-fits-all* approach to mobile network infrastructure is unable to deal with the expected wide range of services and the extremely different QoS requirements of 5G [2]. In order to be able to serve this traffic, virtualization emerges as an essential component at the network edge, namely the virtual partitioning of the mobile Radio Access Network (RAN).

Through virtualization, Mobile Network Operators (MNOs) will be able to create on-demand isolated slices on top of the physical network to support various use cases, such as the Internet of Things (IoT), automated cars, streaming video, remote health care, etc [3], [4]. To fully meet these application demands, RAN architecture must be flexible enough to adapt the network to such diverse requirements. Centralized/Cloud RAN (C-RAN) has emerged as a flexible architecture to improve performance thanks to its ability to coordinate between access nodes, while it is cost-efficient due to resource pooling. In 5G, C-RAN will be composed of a Central/Cloud Unit (CU) and a set of geographically distributed Remote Radio Heads (RRHs) connected through a packet-based network (i.e., integrated Fronthaul/Backhaul (FH/BH)) as proposed by 3GPP [5]. More recently, a flexible design approach is suggested for C-RAN, where the optimal distribution of BS functions between the CU and the RRHs, known as Functional Split (FS), is challenging [6]–[8]. This architecture determines the amount of functions left locally at the RRHs, and the amount of functions centralized at a high processing CU. A proper choice of FS depends on the capacity of the FH/BH network, as the centralization of the RAN functions imposes strict capacity requirements in the FH/BH network. This renders the design of the FH/BH network even more complicated due to the virtualization and capability of having multiple split choices per RRH.

II. Related Work

Given the importance of Software Defined Networking (SDN) as an enabler for both virtualization [9] and slicing [10], some recent works have focused on RAN virtualization platforms and slicing designs. Authors in [11]–[14] studied network slicing for C-RAN resources, yet FS and FH/BH network are missing. Foukas et al. propose FlexRAN [15], a flexible and programmable Software-Defined RAN (SD-RAN) platform, composed of a centralized controller and one agent per eNB that separates control and data planes and allows a flexible control plane

design. However, despite making a step forward in the direction of the virtualization of the RAN, the proposal still lacks a slicing design. The same authors have also proposed Orion [16], which is a RAN slicing design running on the FlexRAN platform that guarantees the functional isolation among slices. Isolation of functions among slices is of paramount importance since it allows a slice-custom FS within a single shared eNB, i.e., different slices sharing the same physical node can be configured with different FSs. For instance, in a given eNB, a slice serving a high-speed UE better suits a centralized FS, so that the coordination among neighboring cells is tighter and the handover performance can be simplified. Conversely, the slice serving a low latency UE would require a decentralized FS to reduce the Hybrid Automatic Repeat Request (HARQ) delay. This is the main weakness of [17]–[20], where for simplicity, either no slicing is considered or all slices adopt the same FS. Two notable recent works propose Wizhaul [21] and FluidRAN [22] to address the FS optimization. More specifically, WizHaul [21] formulates a joint routing and FS optimization to maximize the Centralization Degree (CD) of the network, i.e., the network functions placed at the CU, according to the availability of the network resources. Similarly, FluidRAN [22] follows the same rationale but targeting at the monetary cost minimization. However, despite their insightful conclusions, the slicing option in the RAN is neglected in both of these works. Given the complex and diverse set of QoS requirements of services that 5G will have to serve, the approach proposed in [21] that optimizes the CD, tends to prioritize high throughput services. In order to overcome this aspect, in this work, we introduce the Service Level Agreement (SLA) per slice, defined as the percentage of UEs of a specific slice served with the required QoS (throughput, latency, etc). In that sense, different SLAs have been investigated per slice. The considered SLAs in this work, are focused on keeping the percentage of UEs under control. Thus, guaranteeing the SLA associated with each service accepted by the network.

In this context, our recent work in [23] proposes a joint routing (from UE to CU) and FS optimization while considering different slices, and shows that there is a trade-off between CD (i.e. the allocation of network functions in the CU or in the RRH) and the throughput in the network.

In this work, we extend our previous work and develop a Mixed Integer Programming (MIP) framework, which maximizes the throughput, covering the optimization of routing and FS selection, and setting minimum SLA thresholds for each service to solve the prioritization problem in [21] and meet the diverse set of QoS requirements of services. We further propose a heuristic method, named SlicedRAN: a service-aware network slicing framework for 5G RAN, which finds near-optimal solutions in a short time. We elucidate how to solve the problem of providing isolated and tailored slices for different services with customized FSs per slice when CU and RRHs are connected through a FH/BH network. The work not only proposes a joint slicing, FS and routing solution, but it is also intended to gain insight into how RAN should be designed, and the interweaving of the different RAN aspects.

The remainder of this paper is structured as follows: Section III introduces the system model. In Section IV, we formulate the MIP problem, then propose the heuristic SlicedRAN. Section V provides a performance evaluation of our framework and its results. Finally, Section VI concludes the paper with suggestions for future work.

III. System Model

In this section, we model the traffic and the RAN, including the CU, the RRHs, and the integrated FH/BH network connecting them. Likewise, the FSs and the traffic routing in the network are described, and finally, the associated constraints are defined.

A. Radio Access Network

The initial C-RAN concept is to apply a single direct FH link to connect each RRH to the BBU pool (equivalently, CU). However, due to concerns to scalability, CAPEX, and multiplexing, it is expected that the FH will evolve towards more complex and shared topologies which have been comprehensively explored in [18], [24]. In this work, we focus our discussion around a fully connected network topology, and presents a simple but a realistic deployment of the C-RAN network topology which is composed of a CU, a set of RRHs, and an integrated packet-based FH/BH network (often known as crosshaul [25]), which is a set of forwarding nodes (i.e., routers) connecting CU and RRHs, as introduced in [5]. However, our framework can be applied to different network topologies by modifying the capacity of specific links in the network.

We define a C-RAN architecture as a graph topology $G = (\mathcal{I}, \mathcal{Q}, \mathcal{L})$, where $\mathcal{I}$ is the superset of CU (node 0) and the set of R RRHs, $\mathcal{Q}$ is the set of forwarding nodes (i.e., routers), and $\mathcal{L}$ is the set of links $\mathcal{L} = \{l_{i,j} : i, j \in \mathcal{I} \cup \mathcal{Q}\}$ connecting these elements whose vertices can be divided into two disjoint sets $\mathcal{I}$ and $\mathcal{Q}$, that is, $\mathcal{I}$ and $\mathcal{Q}$ are each independent sets such that every edge connects a vertex in $\mathcal{I}$ to one in $\mathcal{Q}$ (see Fig. 1 (a)). Vertex sets $\mathcal{I}$ and $\mathcal{Q}$ are often known as bipartite [26] sets. Accordingly, the set of forwarding nodes is defined as $\mathcal{Q} = \{1, \ldots, Q\}$; and the set of RRHs, $\mathcal{R} = \{Q+1, \ldots, Q+R\}$. Each link $l_{i,j} \in \mathcal{L}$ has a capacity equal to $\omega_{i,j} \geq 0$ (in b/s). Fig. 1 (a) shows the layout of the RAN, where forwarding nodes are organized as a matrix of m rows and Q/m columns. The connection between the CU and an RRH can be realized through multiple paths [22], each path including several links. Given that there might exist multiple paths between CU and RRH r, the set of all possible paths from CU to RRH r is denoted by $\mathcal{P}^r$.

The computational capacity of the CU and the RRHs is limited and expressed as κ_r for $\forall r \in \mathcal{R}$ and κ_0 for CU. As for the bandwidth allocated to RRHs, we define ρ^r as the number of Physical Resource Blocks (PRB) allocated to RRH r (See Table II). For the sake of simplicity, as used in [27], in the following, we assume equal transmitted power

per PRB with a distance-dependent path-loss model, as for the Signal-to-Noise Ratio (SNR) and for the Modulation and Coding Scheme (MCS), we adopt the models used in [28].

B. Traffic model

In our system model, we focus on the DownLink (DL) traffic, however, our study could be extended to include UpLink (UL). The set of UEs is denoted by $\mathcal{U}$, and the cardinality of the set is expressed by U. Each UE demands a service type $s \in \mathcal{S}$, which is mainly characterized by a required data rate. Thus, the data rate required by UE u with service s is denoted by λ_u^s. We also denote the total number of UEs with service of s as η^s. These demands at each RRH create an aggregate flow emanating from the CU routed to RRH. Hence, the RAN operation can be modeled as a multi-commodity flow problem where the flows rely on the FS at each RRH.

C. Functional Splits

The protocol stack in an eNB consists of several layers, each one responsible for a specific function or a set of functions [29]. In this context, the FS can be defined as the distribution of functions/layers between the CU and the RRH. 3GPP has proposed in [5] a wide range of possible granularities for the FS, from the coarsest granularity (the FS is determined based on the computational capacity of the RRH and the CU, as well as on the FH/BH network capacity) to the finest granularity (the FS is decided on a UE, bearer or slice basis).

Without precluding any of the granularity levels proposed by 3GPP, in our work, we focus on the slice-based FS, assuming that one slice is created for each service[1]. As shown in [5], the network layers Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC) (high and low sublayers), Medium Access Control (MAC) (high and low sublayers), and Physical Layer (PHY) (high and low sublayers) can be allocated either in the CU or in the RRH. Accordingly, each FS will be defined by the set of functions allocated in the CU and the set of functions allocated in the RRH.

In the sequel, we will assume a set of four network functions, denoted as $\mathcal{F} = \{f_0, f_1, f_2, f_3\}$, where f_0 is the low layer network function (RF, signal and analog processing, etc.), which is always placed in the RRH; f_1 serves all PHY functions except for function f_0; f_2 corresponds to RLC and MAC; and f_3 is the high layer network function (e.g., PDCP and above layers). Depending on the FS, these functions will be allocated either at the RRH or at the CU, and thus defining the FH/BH bandwidth requirements between the CU and RRHs. Note that we focused on the main types of FS options, which are the key splits as discussed in [29]. Thus, the addition of other FS options in our work would not affect the system model. Table I includes a summary of the allocation of functions and the associated FH/BH bandwidth requirements for each split [22]. In principle, regardless of the adopted FS, f_0 is always placed in RRH and f_3 is in CU, thus generating three different FS options, namely split 1, split 2, and split 3. Split 1 is a completely decentralized FS that accommodates all functions except f_3 at the RRH. That is, all layers below PDCP run in the RRH. Given the allocation of functions, this split does not have traffic overhead and the required FH/BH capacity can be approximated by the aggregate UEs' traffic. In split 2, f_2 is moved from the RRH to the CU, thus leaving only f_0 and f_1 in the RRH (see Fig. 1 (b)). This allows a higher degree of coordination among eNBs sharing the same CU, thus enabling better utilization of resources with techniques such as Coordinated MultiPoint (CoMP), frame alignment, and centralized HARQ. However, split 2 allocation imposes higher traffic overhead than split 1. Finally, in split 3, only the RF function is located at the RRH, while the rest of functions are moved to CU (complete centralization), thus transmitting In-Phase and Quadrature (IQ) samples through the FH/BH. In this case, samples are usually encapsulated with Common Public Radio Interface (CPRI) [31] and the required fronthaul capacity depends on the bandwidth allocated to the eNB, the number of antennas, etc. That is, fronthaul capacity requirement does not depend on the UEs' traffic for split 3. The main advantage of split 3 is that the centralization achieves the highest coordination degree among eNBs. Note that processing functions has a cost and needs Central Processing Unit (CPU) processing resources. We use c_1 and c_2 as the CPU computational costs for f_1 and f_2 (CPU reference core per Gb/s).

TABLE I: Functions' allocation and FH/BH bandwidth requirements for a traffic denoted by λ_u^s, for UE u and service s, with 20 MHz bandwidth; Downlink: MCS index 28, 2x2 MIMO replicated from [22].

Split Type	Traffic Load (b/s)	Functions at CU	Functions at RRH
1	λ_u^s	f_3	f_0, f_1, f_2
2	$1.02\lambda_u^s + 1.5 \cdot 10^6$	f_2, f_3	f_0, f_1
3	$2.5 \cdot 10^9$	f_1, f_2, f_3	f_0

Given the described scenario, the network creates different slices on top of the physical RAN to serve the traffic. Fig. 1 (b) conveys an example of a slice created to serve a UE u with service s. The slice is created across the FH/BH network (through one or several paths) and an RRH r, from the CU to the UE. Depending on the service and the required QoS, the FS will be 1, 2, or 3. Note, however, that the set of forwarding nodes, the links, and the RRH can be shared with other slices.

D. Traffic Routing in the RAN

Given that the traffic served by RRH r can be forwarded through any of the paths in $\mathcal{P}^r$, we define the traffic over one of these paths as t_p^r, where $p \in \mathcal{P}^r$. Therefore, the total traffic served by RRH r can be expressed as $\sum_{p \in \mathcal{P}^r} t_p^r$. Similarly, as discussed in subsection III-C, the traffic that traverses the FH/BH network depends not only

[1]Given that slicing will be done on a service type basis as highlighted by 3GPP [30], hereafter service and slice concepts will be interchangeable.

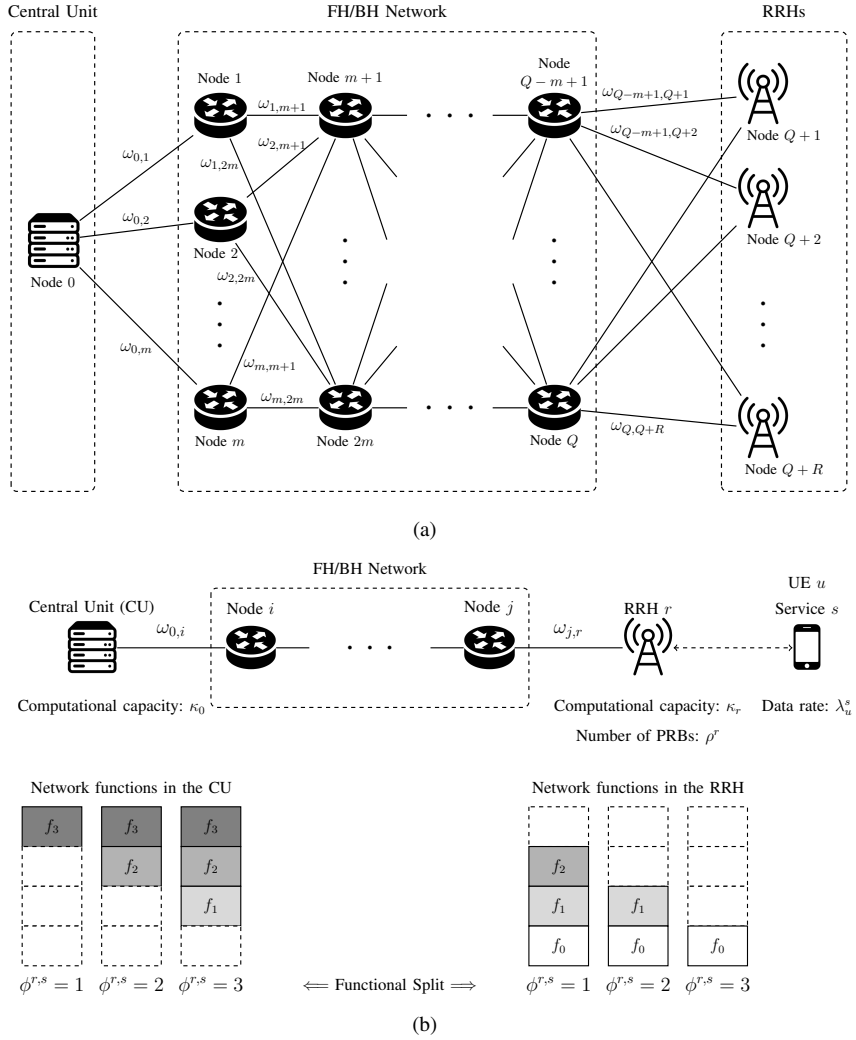

Fig. 1: (a) Radio Access Network model; (b) Scheme of a slice created over a path accross the FH/BH network from the CU to the UE u with service s.

TABLE II: Summary of Notations

Symbol	Description
Sets	
$\mathcal{R}$	Set of RRHs
$\mathcal{Q}$	Set of forwarding nodes
$\mathcal{U}$	Set of UEs
$\mathcal{F}$	Set of network functions
Parameters	
$\omega_{i,j}$	Total bit-rate capacity of link $l_{i,j}$ (b/s)
ρ^r	Available physical resource block (PRB) at RRH_r
$\rho_u^{r,s}$	Required PRB of UEs to connect to RRH r
λ_u^s	Transmission rate of UE u with type s (b/s)
$T^{r,s}$	Traffic served by slice s of RRH r
c_1	CPU consumption to compile f_1 (RCs per Gb/s)
c_2	CPU consumption to compile f_2 (RCs per Gb/s)
κ_r	Computation capacity of each RRH (RCs per Gb/s)
κ_0	Computation capacity of CU (RCs per Gb/s)
τ^s	Proportion of UEs (i.e., SLA) for slice/service type s
η^s	Number of UEs with slice/service type s
w	Bandwidth of a PRB (KHz)
Variables	
$x_u^{r,s}$	Binary variable to associate UE u with type s to RRH r
t_p^r	The variable to show the traffic routing from CU to RRH r
$f_n^{r,s}$	The variable to indicate the placement of functions
$y_{i,j}^p$	The variable to indicate if the path p includes link $l_{i,j}$

on the traffic received/transmitted by/from the UEs but also on the FS. Thus, a UE served by RRH r generating a traffic λ_u^s causes a traffic through the FH/BH network equal to $T_u^{r,s} = \alpha_{\phi^{r,s}}\lambda_u^s + \beta_{\phi^{r,s}}$, where λ_u^s is the traffic generated by UE u with service s and $\phi^{r,s}$ is the FS used in RRH r for service s. In general, $\phi^{r,s}$ can take values in $\{1,2,3\}$. However, due to the constraints imposed by QoS requirements, each service s can only use a subset of FSs, denoted as Φ^s. Thus, $\phi^{r,s} \in \Phi^s \subseteq \{1,2,3\}$. As for $\alpha_{\phi^{r,s}}$ and $\beta_{\phi^{r,s}}$, they are coefficients and used to properly calculate the traffic load in the FH/BH network, and depend on the FS used in RRH r for service s, i.e. $\phi^{r,s}$. As observed in Table I, when service/slice s uses FS 1, i.e. $\phi^{r,s}=1$, we have $\alpha_1 = 1$ and $\beta_1 = 0$. In the case of split 2, $\alpha_2 = 1.02$ and $\beta_2 = 1.5 \cdot 10^6$ b/s. Finally, in split 3, $\alpha_3 = 0$ and $\beta_3 = 2.5 \cdot 10^9 \cdot \frac{\rho_u^r}{100}$ b/s[2], where ρ_u^r is the bandwidth allocated to UE u at RRH r expressed in

[2]According to literature, the required transmission rate required for 20 MHz bandwidth (i.e. 100 PRBs) is around 2.5 Gb/s. This is the reason why the number of PRBs is normalized with respect to 100 PRBs

number of PRBs. Accordingly, the FH/BH transmission rate of slice 3 depends on the bandwidth allocated in the RRH for this slice.

According to the definitions stated above, the traffic traversing the FH/BH network to serve UEs with service s connected to RRH r is given by

$$T^{r,s} = \sum_{u \in \mathcal{U}} x_u^{r,s} \cdot T_u^{r,s}, \tag{1}$$

where $x_u^{r,s} \in \{0,1\}$ is a binary variable equal to 1 when UE u requires service s and is served by RRH r, and 0 otherwise. Each RRH can run different slices and serve different services[3] simultaneously. Thus, if we define the traffic served by slice s of RRH r as $T^{r,s}$, the total traffic served by RRH r can be expressed as $\sum_{s \in \mathcal{S}} T^{r,s}$. Therefore, it holds that

$$\sum_{p \in \mathcal{P}^r} t_p^r = \sum_{s \in \mathcal{S}} T^{r,s}. \tag{2}$$

The accommodation of functions f_0, f_1, f_2 and f_3 in the RRH r or in the CU depends exclusively on the adopted FS. We define the set of variables $f_n^{r,s} \in \{0,1\}$ for $n = \{0,1,2,3\}$. If function f_n runs in the RRH r for service/slice s, then $f_n^{r,s} = 1$. Conversely, if it runs in the CU, then $f_n^{r,s} = 0$. By inspecting Fig. 1 (b) and Table I, it can be shown that, when service s uses split 1, then $f_0^{r,s} = f_1^{r,s} = f_2^{r,s} = 1$ and $f_3^{r,s} = 0$. If it uses split 2, then $f_0^{r,s} = f_1^{r,s} = 1$ and $f_2^{r,s} = f_3^{r,s} = 0$. Finally, if it uses split 3, then $f_0^{r,s} = 1$ and $f_1^{r,s} = f_2^{r,s} = f_3^{r,s} = 0$. Thus, in general, when service/slice s uses FS $\phi^{r,s}$,

$$f_n^{r,s} = \begin{cases} 1 & \text{if } n \leq 3 - \phi^{r,s} \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where $n = \{0,1,2,3\}$ and $\phi^{r,s} = \{1,2,3\}$. Note that allocating a function either in the CU or in the RRH has a computational cost. In the sequel, as described in Section III-C, the computational cost of function f_n, for $n = \{0,1,2,3\}$, is denoted by c_n.

IV. SlicedRAN: Service-Aware Network Slicing Framework for Base Station

The management and the operation of dynamic FS in the RAN pose significant challenges with several trade-offs. On the one hand, a reduction in the FH/BH network's load can be achieved by locating RAN functions at RRHs, though at the expense of increasing the computational needs in the RRHs. On the other hand, offloading the RAN functions and pooling them at the CU benefits from a reduction of the computational capacity required at the RRHs and offers centralized control that can improve the network's performance, but with higher FH/BH bandwidth requirements. In the same vein, not all FSs meet the requirements of all services, and as proposed by 3GPP [5] it is expected that each slice would have diverse QoS requirements. Regardless of how exactly a slice is implemented within the RAN, different functionality mapping (i.e., FS selection) may be suitable for each slice. The QoS requirements of each service have been taken into account when adapting the FS for each service. For instance, eMBB traffic requires a high degree of coordination among eNBs to achieve high data rates. This suggests a scenario in which eMBB UEs require high bandwidth along with high-speed execution for these bandwidth-intensive applications, processing of a vast amount of data in a cloud (equivalently CU) [32]. This means centralizing network functions towards the CU, (e.g. split 3). Conversely, uRLLC needs fast retransmissions to guarantee low latency and high reliability. In that sense, decentralized FSs are needed (e.g. split 1), which means the experienced delay for this service is minimized since the most of functions are decentralized and located in the RRHs. mMTC, such as IoT applications, is a service with which intermediate splits would work [33].

In this context, a convenient network slicing algorithm provides the network with a higher degree of flexibility, thus enabling the adaptation of the FS of each slice to traffic requirements and network limitations (RRH and/or CU computing capacity, FH/BH network capacity, etc.). Thereby, we propose a novel MIP framework to formulate joint FS and network slicing for future 5G networks, and describe the proposed MIP optimization problem formulation. We next propose an effective heuristic method SlicedRAN which is based on Relaxation Induced Neighborhood Search (RINS) heuristic and then we explain it's performance.

A. MIP Problem Formulation

As already stated, in the following we propose an optimization solution aimed to maximize the throughput of the 5G network by jointly selecting the most convenient and efficient FS and routing per slice.

Accordingly, the objective of the solution is to maximize the throughput by determining i) the UE association to the RRH and ii) the path through which each type of traffic is forwarded. In parallel, the solution decides the most appropriate FS for each service based on the constraints of the network, such as the capacity of the links and the computational capacity of the CU and the RRHs.

Hence, the maximization of the network throughput, which is our objective function, can be written as in (4). The constraints of the optimization model are defined in (5) - (13).

$$\text{max.} \sum_{r \in R} \sum_{s \in S} \sum_{u \in U} \lambda_u^s \cdot x_u^{r,s} \tag{4}$$

Subject to:

$$\sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}} \sum_{n=0}^{3} x_u^{r,s} \cdot \lambda_u^s \cdot c_n \cdot f_n^{r,s} \leq \kappa_r, \forall r \in \mathcal{R} \tag{5}$$

$$\sum_{r \in R} \sum_{s \in S} \sum_{u \in U} \sum_{n=0}^{3} \lambda_u^s \cdot x_u^{r,s} \cdot c_n (1 - f_n^{r,s}) \leq \kappa_0 \tag{6}$$

[3]Please, recall that *service* and *slice* are used interchangeably, and we assume that the network creates a slice per each service.

$$\sum_{p \in P^r} t_p^r = \sum_{s \in S} T^{r,s}, \forall r \in \mathcal{R} \quad (7)$$

$$\sum_{r \in R} \sum_{p \in P^r} t_p^r . y_{i,j}^p \leq \omega_{i,j}, \forall j \neq i \in \mathcal{Q} \quad (8)$$

$$\sum_{s \in S} \sum_{u \in U} x_u^{r,s} . \rho_u^{r,s} \leq \rho^r, \forall r \in \mathcal{R} \quad (9)$$

$$\sum_{r \in R} x_u^{r,s} = 1, \forall u \in \mathcal{U}, \forall s \in \mathcal{S} \quad (10)$$

$$\sum_{u \in U} \sum_{r \in R} x_u^{r,s} \geq \tau^s . \eta^s, \forall s \in \mathcal{S} \quad (11)$$

$$x_u^{r,s} \in \{0,1\}, \forall r \in \mathcal{R}, \forall s \in \mathcal{S} \quad (12)$$

$$\phi^{r,s} \in \Phi^s, \forall r \in \mathcal{R}, \forall s \in \mathcal{S} \quad (13)$$

First, (5) ensures that the computational cost of the functions run in each RRH does not exceed the RRH computational capacity κ_r. Similarly, constraint (6) is used to bound the maximum computational capacity supported by the CU (κ_0). Constraint (7) guarantees that all the traffic served by a RRH, regardless of the slice to which the traffic belongs, equals the traffic forwarded over the paths from the CU to the RRH, as shown in (2). Constraint (8) states that the flow from each RRH r to CU is bounded by the capacity of the links of the paths, denoted as $\omega_{i,j}$ (b/s). The $y_{i,j}^p \in \{0,1\}$ indicates if the path p includes link $l_{i,j}$ or not. As for the number of PRBs allocated by each RRH, constraint (9) ensures that the number of PRB allocated to UEs served by the RRH can not exceed the maximum number of PRBs. Moreover, UEs are not served by more than a single RRH simultaneously with constraint (10). We impose a minimum SLA for each slice in constraint (11) to guarantee the SLA of a particular slice. Specifically, $\tau^s \in [0,1]$ stands for the minimum proportion of UEs with service type s that must be served with the required QoS and η^s is the total number of UEs with service type s. Thus, the minimum number of UEs that should meet the required QoS is $\tau^s \cdot \eta^s$. Indeed, this constraint solves the problem of prioritization for the services with higher bandwidth requirements and creates a balance for the diversity of accepted services/slices, where its impact is analysed in Section V. Constraint (12) defines $x_u^{r,s}$ as a binary variable. Finally, (13) defines the set of allowed FSs for a slice/service. For illustrative purpose, let us assume that a given service s must be served always with FS 1. In that case, the set of allowed FSs, denoted in (13) as Φ^s, would be $\Phi^s = \{1\}$. Thus, (13) forces all RRHs to serve service s with FS $\phi^{r,s} = 1$. Instead, if we assume that a service can be delivered with splits 2 and 3, then $\Phi^s = \{2,3\}$. In this case, the FS for service s can take values $\phi^{r,s} = \{2,3\}$ in the different RRHs. It is worth noting that the value of $\phi^{r,s}$ determines the value of $f_n^{r,s}$ for $n = \{0,1,2,3\}$ according to (3).

Theorem 1. *The optimization model turns out to be a MIP problem, which is an NP-complete problem, and has a complexity of* $\mathcal{O}(2^N)$.

Proof. The optimization model that maximizes the network throughput is a MIP problem, for which commercial and free solvers can be used. Generally, a MIP problem is known to be an NP-complete problem, and as the computation time for NP-complete problems is high [34], the number of nodes (UEs and RRHs) in the network negatively affects the computation time due to the increment of the search space of the variables. We solved the MIP optimization using IBM ILOG CPLEX Optimization Studio [35]. This optimizer has a high-performance solver which uses algorithms such as branch-and-bound, branch-and-cut, etc. Jeroslow [36] proved that the complexity of branch-and-bound for a MIP problem is $\mathcal{O}(2^N)$, where N is the number of variables in the optimization. In our optimization problem, we have binary ($x_u^{r,s}$) and continuous (t_p^r) variables that require the branch-and-bound method.

We next propose a solution algorithm for the MIP problem. Note that our MIP optimization model and the provided solution algorithm are generic and can be easily extended for various scenarios where the routing and computation cost functions are strictly convex and linear on the UEs' traffic load.

B. Solution Method: RINS Heuristic for SlicedRAN

The computational complexity of the MIP problem increases substantially for large scale of networks, and the number of nodes in the network negatively affects this computation time. To overcome this issue, one solution could be using Reinforcement Learning (RL) algorithms or developing a heuristic approach. Indeed there is a trade-off between selecting the heuristic approach and RL algorithms. In particular, the training time in RL algorithms is high whereas in heuristic approach is null. Conversely, once trained, the computational time is lower for RL than for the heuristic method. The heuristic approach is able to face changes better than RL as long as it can be executed fast enough (i.e., reduced computational complexity). To this aim, we develop a heuristic solution, SlicedRAN, using a heuristic method to handle it in a shorter computing time. One of these heuristic methods is known as RINS [37] that separates a MIP problem into sub-problems and explores a neighborhood of the current incumbent solution and solves reduced problems at some nodes of a branch-and-cut tree and obtains a good solution among the incumbent solution. The proposed solution algorithm is summarized in Algorithm 1.

According to this algorithm, we take a network topology, the set of all UEs, and the proportion of UEs $\tau^s \in [0,1]$ as an input and then initialize the SNR and $\rho_u^{r,s}$ of each UE to the list of RRHs. Next, for all UEs belong to each slice we solve SlicedRAN based on RINS (line 1-5), Then, we check the constraint (11) to check SLA threshold for each service; if the network guarantees the current SLA (i.e., τ^s) (line 6), then the feasible solution is sought for the current slice s, where the solution is stored in $Sol.s$ (line 7). Otherwise, the solution is infeasible due to insufficient resources in the network (line 10). Finally, we store the final solution of the current slice s in Sol (line

13) and return this solution as an output of our algorithm (line 15).

Algorithm 1: SlicedRAN

Input: $G = (\mathcal{I}, \mathcal{Q}, \mathcal{L})$: a network topology graph
$\mathcal{U}$: the set of UEs
τ^s: the proportion of UEs (i.e., SLA %)
Initialize: Compute SNR and create candidate list of RRHs for each UE
Compute $\rho_u^{r,s}$ for each UE and create candidate list of RRHs based on $\rho_u^{r,s}$
Output: Sol: the solution for all UEs of each slice

1 **repeat**
2 | $\forall u \in \mathcal{U}$
3 **foreach** $s \in S$ **do**
4 | **for** τ^s = *0 to 100* **do**
5 | | $tmp \leftarrow$ Solve SlicedRAN based on RINS heuristic in sub-problems
6 | | **if** *SLA constraint holds* **then** ▷ (11)
7 | | | $Sol.s \leftarrow tmp$ ▷ Feasible solution
8 | | **end**
9 | | **else**
10 | | | $Sol.s \leftarrow Inf.$ ▷ Infeasible solution due to lack of resources
11 | | **end**
12 | **end**
13 | $Sol \leftarrow Sol.s$
14 **end**
15 **return** Sol

Theorem 2. *The run-time complexity of SlicedRAN is* $\mathcal{O}\big((|S|.|U|.|T|).(2^{|S|.|U|.|R|+|P|.|R|})\big)$.

Proof. SlicedRAN starts at line 2 and ends at 15. In this loop, we iterate on the number of UEs $|U|$ and with the number of services $|S|$. For each SLA threshold τ^s, it takes the number of $|T|$ steps. We next run RINS in subproblems, exploring the convex hull of all the feasible solutions for binary variable $x_u^{r,s}$ and continuous variable t_p^r. The binary variable $x_u^{r,s}$ has a maximum number of $|\mathcal{S}|.|\mathcal{U}|.|\mathcal{R}|$, and continuous variable t_p^r has a maximum number of $|\mathcal{P}|.|\mathcal{R}|$ in the network. Thus, the run-time complexity of SlicedRAN is $\mathcal{O}\big((|S|.|U|.|T|).(2^{|S|.|U|.|R|+|P|.|R|})\big)$.

V. Performance Evaluation

In this section, we present the effectiveness of the proposed solution from the overall system and slice viewpoint by investigation of numerical results for the performance of SlicedRAN.

A. Simulation Scenario

In our analysis, we consider three main types of uRLLC, mMTC, and eMBB services with different QoS requirements. Table III summarizes our simulation setup, where we assume a bandwidth of 20 MHz for each BS (i.e, $\rho^r = 100$ PRBs) with 4 forwarding nodes ($Q = 4$ and $m = 2$ in Fig. 1(a)) and a link capacity ranging from $\omega_{i,j}$ = 100 Mb/s to $\omega_{i,j}$ = 25 Gb/s. We study a scenario composed of a single CU connected to a set of RRHs from 4 to 16 (R = 4 to R = 16) and adopt the values of [38]–[40] to define three types of applications, i.e., medical, IoT, and video streaming applications. We consider s = 1 for medical applications (uRLLC) which use split 1 with λ_u^1 = 120 Kb/s, s = 2 for IoT messages (mMTC) which use split 2 with λ_u^2 = 30 Kb/s, and finally s = 3 for video streaming applications (eMBB) which need a higher degree of centralization (i.e., split 3) with λ_u^3 = 20 Mb/s. As for the computational capacity, we utilize the values used in [22], with $\kappa_0 = 100$, $\kappa_r = 1$ CPU reference core per Gb/s. Regarding the computational cost, $c_1 = 3.25$ and $c_2 = 0.75$ CPU reference core per Gb/s. Note that we exclude the computation costs for $f_0^{r,s}$ which is always placed in RRHs and $f_3^{r,s}$ since it is always in the CU. We consider a distance-dependent path-loss model with transmission power 30 dBm and for the MCS calculation, we adopt the values used in [28]. We then explore two configurations for the evaluation of SlicedRAN.

Configuration 1 (C.1): This configuration is a uniform distribution where 33%, 34%, and 33% are set to UEs with the type of medical applications, IoT messages, and video streaming, respectively. This configuration has a balanced number of different UEs with different QoS requirements, totally 1000 UEs (i.e., U = 1000). Indeed, eMBB UEs which need higher bandwidth (i.e., PRB) in RRHs have the same distribution of mMTC applications (i.e., IoT UEs) which require less bandwidth while injecting extra overheads into the FH/BH network.

Configuration 2 (C.2): In general, the distribution of traffic (each type of UEs) is not necessarily uniform, as a massive number of IoT connections is expected. Therefore, we consider a scenario with 6380 UEs (i.e., U = 6380)[4] where 80% of the connections correspond to IoT applications (mMTC), while 15%, and 5% are set to UEs with medical applications (uRLLC) and video streaming (eMBB), respectively. Apparently, this configuration has fewer eMBB UEs, thus less requirements in terms of PRBs in RRHs. On the other hand, it has more IoT applications which adds huge overheads into the network, thus higher requirements in terms of the capacity of FH/BH networks.

We have conducted extensive Monte-Carlo simulations implemented in Java, while the optimization model is built and solved with IBM ILOG CPLEX Optimization Studio [35]. Note that the computing time needed to obtain the optimal solution (i.e., MIP) with a CPU processor of Core i7-8550U, a RAM of 16 GB for a scenario composed of a single CU, $Q = 4$, R = 10 and for C.1, is the matter of hours while the proposed algorithm (i.e., SlicedRAN) is able to obtain the near-optimal solution around 9 seconds.

We obtain results by maximizing the throughput for each configuration for three metrics of interest:

- *Served Traffic:* to identify the portion of traffic served by applying SlicedRAN on the existing network infrastructure;

[4]In order to have a fair comparison, we consider the same offered traffic of 6.65 Gb/s for both C.1 and C.2. This is why for C.1 totally U = 1000 UEs and for C.2 U = 6380 have been chosen.

- *Link Usage:* to explore the minimum capacity required in the FH/BH network for each configuration;
- *Spectrum Usage:* to study a cost-efficient (i.e., the minimum number of RRHs) set-up required in RRHs per configuration.

All results are compared with [21] where the main objective is maximizing CD and no slicing is considered. Therefore, a single FS is allowed to use in each RRH. In the following, the MIP optimization is labeled as *Optimal*, the proposed heuristic as *SlicedRAN*, and the state of the art [21] as *SoA*.

TABLE III: Simulation Setup

System Bandwidth	20 MHz
Number of PRBs (ρ^r)	100
Number of RRHs (R)	4 - 16
Number of Forwarding Nodes (Q)	4
Number of UEs (U)	1000 - 6380
The capacity of links ($\omega_{i,j}$)	0.1 - 25 Gb/s
Transmission rate of uRLLC (Medical apps)	120 Kb/s
Transmission rate of mMTC (IoT msg)	30 Kb/s
Transmission rate of eMBB (Video Streaming)	20 Mb/s
Transmitted power	30 dBm
CPU consumption to compile f_1 (c_1)	3.25 RCs per Gb/s
CPU consumption to compile f_2 (c_2)	0.75 RCs per Gb/s
Computation capacity of each RRH (κ_r)	1 RCs per Gb/s
Computation capacity of CU (κ_0)	100 RCs per Gb/s
Value of SLA for each slice (τ^s)	5% - 100%

Within the simulation, we compare the performance of SlicedRAN and Optimal with the benchmark scheme of SoA ([21]). We analyse the capacity requirements of RRHs in Section V-B, where we only evaluate the network performance without imposing constraints in the FH/BH network; then, we enforce the FH/BH constraints in Section V-C to explore the impact of the limitation on the FH/BH network in order to provide the guidelines on the design of the FH/BH network. As stated before, maximizing the throughput has a great impact on the QoS of those services/slices which have less bandwidth requirements. To this end, we analyse the impact of imposing a minimum SLA for the network performance in Section V-D.

B. Analysis of Capacity of RRHs

As mentioned previously, the aim of investigating the capacities of RRHs is to find a proper set-up of flexible FS along with the minimum capacity of requirements per RRH. We thus explore the network performance where only constraints on the capacities of RRHs are imposed.

We next present the results of our numerical analysis in Fig. 2, where we assume that $\omega_{i,j} = \infty$ and $\tau^s = 0, \forall s \in \mathcal{S}$ for a set of RRHs from 4 to 16 BSs (R = 4 to R = 16) where the offered load is 6.65 Gb/s for C.1.

As can be seen in Fig. 2, it is evident that the increase in the number of RRHs is consistent with the increase in the average of served traffic in Optimal, SlicedRAN, and SoA. The heuristic SlicedRAN has a higher performance compared to SoA while lower than Optimal. However, it performs very close to Optimal with a very lower computation time. For example, when R = 6 SlicedRAN achieves 5.54 Gb/s throughput while SoA can reach up to 2.95 Gb/s in the throughput, which indicates 85% of gain in the performance of SlicedRAN when compared with SoA performance. The main reason for outperforming SlicedRAN is strongly linked to the benefits of virtualization, which allows RRHs to use different FSs to serve diverse traffic demands; while in SoA (without slicing) each RRH is allowed to use only a single FS in order to serve the corresponding service. Furthermore, by analysis of this figure, it can be identified that increasing the number of RRHs has more impacts on SoA performance. We observe that a smaller number of RRHs results in more traffic rejection in SoA and increasing the number of RRHs adds more diversity in terms of FS for SoA, thus supporting plenty of different services. For example, by increasing RRHs from R = 6 to R = 10 (i.e., the addition of four RRHs) in SoA, the mean served traffic increases up to 68% (from 2.95 Gb/s to 4.98 Gb/s). Whereas this increase of traffic in SlicedRAN is $\sim 14\%$ (from 5.54 Gb/s to 6.30 Gb/s). As stated above, there is a clear evidence that having a network with a small number of RRHs (each RRH with a single FS) fails to support plenty of distinct services. Hence, at the beginning the UEs who were not able to be served with nearby RRHs (due to different FS requirements), with increasing the number of RRHs, the diversity of BSs with different FSs allows them to find a BS with the corresponding FS to meet their requirements.

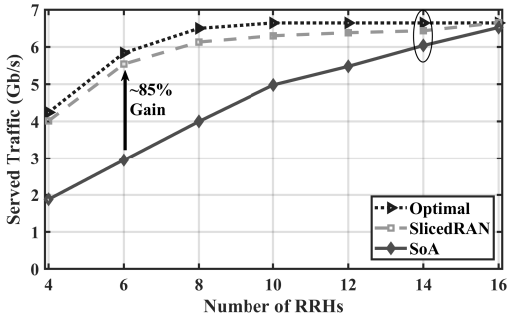

Fig. 2: Average served traffic w.r.t number of RRHs from R = 4 to R = 16 where $\omega_{i,j} = \infty$ with offered load = 6.65 Gb/s for C.1

Similar results were found after evaluating C.2 in Fig. 3, wherein the same settings of Fig. 2 is applied. It is apparent that in all cases SlicedRAN outperforms SoA with a significant difference in the cases of a small number of RRHs. For example, SlicedRAN achieves 5.47 Gb/s in throughput when R = 6, while SoA reaches only 2.57 Gb/s, hence, we have considerably higher gain in the performance of SlicedRAN, that is $\sim 112\%$ gain in throughput.

Comparing Figs. 2 and 3 shows that a significant improvement was obtained in the majority of cases. However, the main inspection of Fig. 3 indicates that SlicedRAN serves almost all traffic when R = 14, while in Fig. 2 in order to reach this performance, two more RRHs are needed (i.e., R = 16), which leads to extra deployment costs for MNOs. This is because in C.1 we have a uniform distribution of traffic i.e., with the same number per each type of UEs, which leads to higher bandwidth

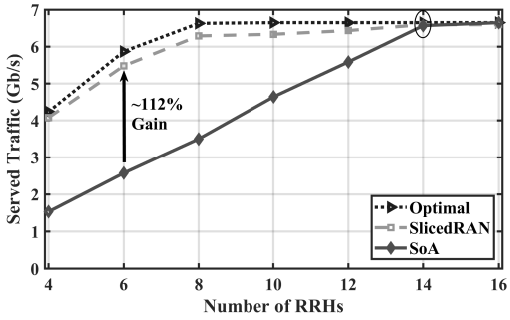

Fig. 3: Average served traffic w.r.t number of RRHs from R = 4 to R = 16 where $\omega_{i,j} = \infty$ with offered load = 6.65 Gb/s for C.2

requirements in RRHs (due to existing more eMBB UEs). In addition, we observe that SoA has a better performance in Fig. 2 which is evaluated for C.1. For example, the average of served traffic in Fig. 2 for R = 8 is higher than 4 Gb/s, while in Fig. 3 (i.e., C.2) the achieved value is less than 3.5 Gb/s. The main reason for this behavior is that the distribution of traffic in C.2 is not uniform, and we have fewer eMBB UEs that need higher bandwidth requirements (i.e., PRB) in RRHs. Indeed, C.2 is mainly composed of 80% of mMTC UEs and 15% of uRLLC UEs (i.e., in total 95 % of all offered traffic), and only 5% of all UEs are eMBB UEs. Hence, the usage of capacities in RRHs (i.e., PRB, computation cost, spectrum) in C.2 is fewer than C.1, where it has a uniform distribution of traffic (i.e., the same number per each type of UEs). Furthermore, SoA in C.2 serves almost all traffic with R = 14 while C.1 needs at least two more RRHs (i.e., R = 16) in order to achieve the same throughput. The main reason for this contradicting behavior is that as we have fewer number of eMBB UEs in C.2, the diversity requirements of RRHs with different FS is lower, which means fewer number of RRHs are needed to meet the requirement of eMBB UEs. On the other hand, in C.1, the uniform distribution entails distinct FSs for each type of service and due to using only a single FS in SoA, this uniform traffic needs more RRHs with different FS in order to serve all traffic.

From these results, we can conclude that with slicing we manage to better use the resources in terms of spectrum usage in RRHs by creating different slices in each RRH. Thus, to achieve the same performance in SoA, MNOs need to deploy more RRHs, which adds more costs for them. This is more important for MNOs to decrease the hardware deployment costs where they are in quest of a cost-efficient design of the BS framework.

C. Analysis of FH/BH Network

The results of subsection V-B showed the gain achieved by slicing in RRHs, when no constraints were imposed to the FH/BH network. However, the design of the FH/BH network has an impact on this gain. Note that the degree of centralization depends on the design and the available capacity in FH/BH networks. Indeed as we increase the capacity of links, higher traffic can be served in the network, and the impact of the FH/BH network on the benefits of slicing can be diminished.

In this regard, we analyse the FH/BH networks for bipartite network topologies by increasing the capacity of links (i.e., $\omega_{i,j}$). First, we explore this analysis for C.2, where more mMTC UEs are deployed. As you see in Fig. 4, the remarkable point is that imposing constraints in the FH/BH networks leads to the loss in the throughput gained in Section V-B. The results show that the heuristic SlicedRAN and Optimal have tight and close results, and, as we increase the capacity of the FH/BH links, more throughput is achieved. However, increasing the capacities of FH/BH networks makes this loss and limitation negligible in SlicedRAN while it remains suffering in SoA. Fig. 4 depicts that only 16% of the offered traffic is actually served when $\omega_{i,j}$ = 100 Mb/s (a loss equal to 84%) for SlicedRAN while this loss is $\sim 97\%$ for SoA. Remarkably, this reduces as we increase the capacities in FH/BH networks. For example, the loss of traffic by SlicedRAN is $\sim 25\%$ when we have $\omega_{i,j}$ = 1 Gb/s while in SoA it is close to 65%, and when we increase the capacity to $\omega_{i,j}$ = 2 Gb/s, the average of traffic served by SlicedRAN is higher (i.e., 6.55 Gb/s) which means under 1% of offered traffic is lost while SoA achieves 4 Gb/s which means 40% of offered traffic is dropped. This effect is pronounced for the utilization of slicing in SlicedRAN, which better uses the resources in the FH/BH network by employing different FSs, which has an impact on the FH/BH network in order to serve different services with various QoS requirements. Whereas in SoA, on one side, only a single FS is used in each RRH to serve corresponding traffic of services. On the other side, SoA suffers from the limitation in the capacities of RRHs since it achieves the same throughput of SlicedRAN when we have more number of RRHs, that is, at least R = 14 RRHs for SoA (See Fig. 3).

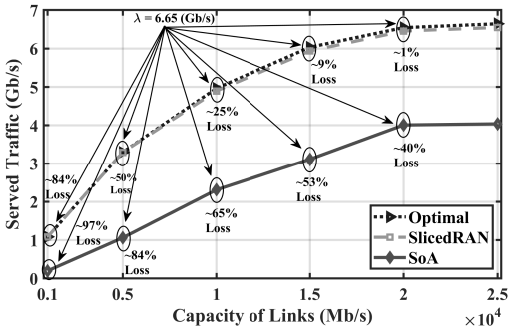

Fig. 4: Average served traffic w.r.t the capacities in FH/BH network (i.e., $\omega_{i,j}$ (in Mb/s)) with offered load = 6.65 Gb/s for C.2

In Fig. 5 which is obtained for C.1, the results demonstrate relatively the same behavior as Fig. 4. Comparing these figures shows that when we have more capacities in FH/BH network, SoA in C.1 performs better when compared to SoA performance in C.2. This is because

TABLE IV: Resource usage for C.2

Optimal (A) / SlicedRAN (B) / SoA (C) performance with 10 RRH									
$\omega_{i,j}$ (Gb/s)	Served Traffic (%)			Link Usage (%)			Spectrum Usage (%)		
	A	B	C	A	B	C	A	B	C
1	16.3	16.3	3	99.9	99.5	99.9	17.3	17.3	7.7
5	49.6	49.6	15.9	99.8	99.5	99.9	62.6	62	27.5
10	74.7	74	34.7	99.7	98	99.9	89.9	95.9	54.2
15	90.8	89.3	46.8	99.8	96.2	99.7	95.1	99	72.6
20	98.5	91.1	60.3	99.3	98.2	91.3	97.1	98.7	85.9
25	100	91.6	60.7	88.1	87.2	73.6	98.5	98.5	85.8

TABLE V: Resource usage for C.1

Optimal (A) / SlicedRAN (B) / SoA (C) performance with 10 RRH									
$\omega_{i,j}$ (Gb/s)	Served Traffic (%)			Link Usage (%)			Spectrum Usage (%)		
	A	B	C	A	B	C	A	B	C
1	15.5	15.4	3.2	99.6	99.8	99.9	11.4	15.3	7.3
5	49.4	48.5	15.6	99.8	99	99.9	56.2	55.6	28.4
10	75.6	70	34.5	99.5	98.2	99.9	81.9	87.4	50.9
15	93.1	82.3	67.7	99.6	93.8	99.9	92.1	91.8	85.5
20	99.9	90.7	74.9	89.1	91.8	78.6	93.4	90.5	90.5
25	100	90.9	75.2	88.7	75.7	74.3	95.2	85.5	90.5

in C.2 we have more deployed eMBB and mMTC UEs that inject more overhead and capacity requirements in FH/BH network; hence, more traffic is rejected.

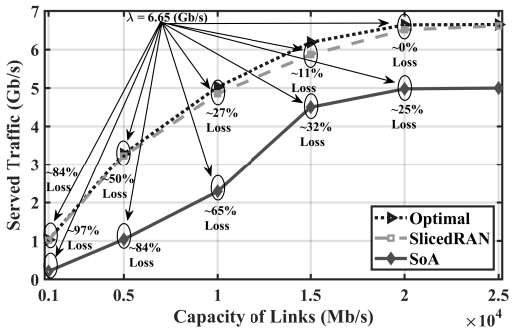

Fig. 5: Average served traffic w.r.t the capacities in FH/BH network (i.e., $\omega_{i,j}$ (in Mb/s)) with offered load = 6.65 Gb/s for C.1

In Table IV, we assess the resource usages of the network in terms of *Served Traffic, Link Usage, Spectrum Usage* for C.2. It must be pointed out that both SlicedRAN and Optimal better use the resources, and more traffic is served with the same resources in the network when compared to SoA. For example, SlicedRAN achieves superior results respectively up to 74% of total traffic when the limitation is imposed into the FH/BH network with $\omega_{i,j}$ = 10 Gb/s whereas with the same capacity in the FH/BH network achieved traffic is about 34.7% in SoA which is less than half of the traffic which is served by SlicedRAN. A similar pattern was obtained when capacity is increased to $\omega_{i,j}$ = 15, 20 Gb/s, where SlicedRAN achieves 89.3%, 91.1%, respectively while SoA reaches only 46.8%, 60.3% sequentially of the total traffic offered.

A similar conclusion was reached by Table V where we have 10 RRHs in the network for C.1. The findings are directly in line with previous findings particularly when the capacity of the FH/BH links is small. From this table, it is evident that the performance of SlicedRAN is substantially better than SoA performance. For instance, when $\omega_{i,j}$ = 1 Gb/s SlicedRAN performs almost five times better than SoA in the percentage of traffic served while using 53% more in the spectrum. It is essential to highlight the fact that in almost all the cases, the usage of the FH/BH links is close to 100%, while in the case of spectrum usage, SlicedRAN performs better when compared to SoA performance.

From these results in tables IV and V, we conclude that dimensioning the FH/BH network, impacts on all explored metrics (*Served Traffic, Link Usage, Spectrum Usage*) of the network. Indeed as much as we increase the capacity of links in the FH/BH networks, it allows us to serve more traffic, especially with leveraging virtualization in SlicedRAN we can have higher gains in terms of *Served Traffic* and efficient resource usage (i.e., *Link Usage, Spectrum Usage*) when compared with SoA.

Having different QoS requirements in 5G especially for three main types of eMBB, uRLLC, and mMTC services need to adopt by MNO's to meet the diversity of these QoS requirements. Indeed, maximizing the throughput is challenging to satisfy the QoS of different services, and could have an impact on the QoS of those services which have less bandwidth requirements on the network. Fig. 6 illustrates this challenge for the proposed SlicedRAN where the objective is maximizing the throughput and all links have the same capacity in the FH/BH network. As can be observed from this figure, SlicedRAN prioritizes the uRLLC and eMBB services, and after it satisfies all of these services (i,e., $\omega_{i,j}$ = 20 Gb/s), then starts to increase serving mMTC services which have less bandwidth requirements. For example, in this figure, almost all eMBB and uRLLC services are served when $\omega_{i,j}$ = 20 Gb/s. On the other hand, served traffic for mMTC services remain under 50% with the same capacity. Indeed, up to 50% of mMTC services are dropped as they have less bandwidth requirements on the FH/BH network. To this end, in the next subsection, we analyse the imposing of different SLAs to guarantee QoS of mMTC services.

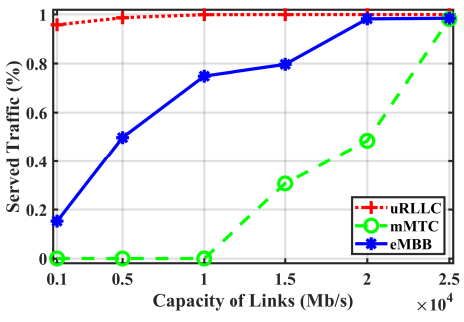

Fig. 6: Served traffic (%) w.r.t the capacities in FH/BH network (i.e., $\omega_{i,j}$ (in Mb/s)) when R = 10 with offered load = 6.65 Gb/s for C.2

D. Analysis of Imposing SLA for each Slices

As observed in Section V-C, the extreme demands with different QoS requirements in 5G show that maximizing

the throughput could have an impact on the QoS of those services which have less bandwidth requirements on the network. To overcome this, a constraint to guarantee a minimum percentage of UEs per slice that meet the required QoS is needed. As explained in Section IV-B, we denote this as SLA. In the following, we analyse the impact of imposing the SLA for each slice. Due to the limitation of resources, it is not always feasible to achieve the SLA. In that case, the solution will be infeasible (as explained in Algorithm 1), we choose the scenarios where the solutions are feasible.

We first assess the analysis of imposing SLA to one slice on the performance of other slices in C.1. Fig. 7 shows the analysis of enforcing SLA for mMTC slice and its impact on other slices. It can be seen in this figure that mMTC slice has a minimum priority to be served in all cases, and the served traffic for this slice is in the minimum percentage comparing with other slices. This behavior is linked to the objective function (i.e., maximizing throughput) which gives higher priority to the slices with higher data rate requirements (i.e., eMBB and uRLLC slices).

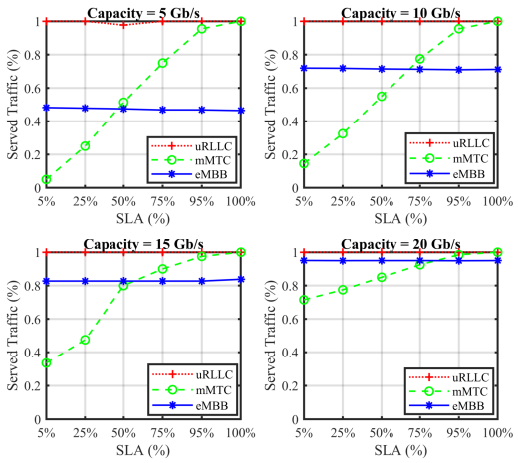

Fig. 7: Served traffic (%) w.r.t the different SLAs imposed for mMTC slice with R = 10 for C.1

We now analyse the impact of enforcing SLA on the performance of eMBB and uRLLC slices in C.2. Fig. 8 shows the analysis of enforcing SLA for mMTC slice and its impact on eMBB and uRLLC slices. As can be seen in this figure, mMTC slice has the same behavior as in C.1. This slice has a lower priority (because of lower data requirements); hence it is considered as the last slice to allocate resources. For instance, when $\omega_{i,j}$ = 15 Gb/s increasing SLA from 5% to 100% for mMTC slice yields a reduction of near to 66% for eMBB slice while an increase up to 95% for mMTC slice. This is mainly because mMTC slice has a huge overhead and costs in the FH/BH network while eMBB slice requires more spectrum resources and has a higher data rate requirement, and uRLLC slice has no overhead in the FH/BH network. Hence, it leads to a drop in the eMBB and uRLLC services. That is, a higher SLA in mMTC slice means rejecting eMBB and uRLLC slices and, thus less throughput in the network.

From the Fig. 7 and Fig. 8, we observe that SlicedRAN mitigates the impact of limits of the network (up to 95% of traffic for mMTC slice) and guarantees on the QoS requirements of those services which have fewer bandwidth requirements (mMTC services) with a sacrifice on the other services which have higher bandwidth requirements (up to 66% of traffic for eMBB services).

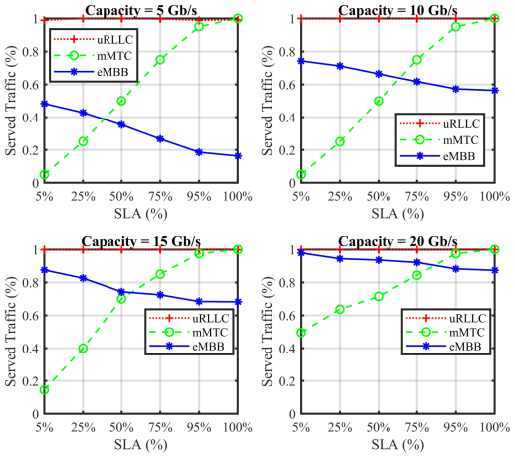

Fig. 8: Served traffic (%) w.r.t the different SLAs imposed for mMTC slice with R = 10 for C.2

VI. Conclusions

RAN slicing needs to cover two main and significant aspects namely, performing a dynamic FS of RAN and creating isolated and efficient slices based on the QoS requirements. In this paper, we proposed SlicedRAN: service-aware network slicing framework for 5G RAN, which creates isolated RAN slices based on the service requirements with customized FSs per slice on top of a network composed of a CU, a FH/BH network, and a set of RRHs. We first formulate a MIP framework, which maximizes the throughput by jointly selecting the optimal routing paths from a connected UE to CU, and FS while satisfying the QoS requirements. We further provide an effective heuristic method, SlicedRAN, that computes near-optimal solutions in a short computing time compared to the optimal one (i.e., MIP). Our framework considers the bottlenecks in the capacity of RRHs, FH/BH network capacity along with a minimum level of SLA for each slice imposed by the different service types. The broad implication of the present research demonstrates a strong trade-off between SLA and the FH/BH network between CU and RRHs which provide a basis for designing a virtualized network infrastructure with a cost-efficient FH/BH network whilst guaranteeing SLA of different slices.

Acknowledgment

This work has received funding from the European Unions Horizon 2020 research and innovation programme

under grant agreement No. 722788 (5G-SPOTLIGHT), 5GROUTES (951867), 5GSolutions (856691), SPOT5G (TEC2017-87456-P), SPOTS (RTI2018-095438-A-I00) funded by the Spanish Ministry of Science, Innovation and Universities, and from AGAUR (2017 SGR 891), and AGAUR (2017 SGR 60).

REFERENCES

[1] ITU, "ITU-R M.2083-0. IMT Vision—Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond," 2015.

[2] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies and Solutions," *IEEE Communications Surveys Tutorials*, p. 1, 2018.

[3] H. Chien, Y. Lin, C. Lai, and C. Wang, "End-to-end slicing with optimized communication and computing resource allocation in multi-tenant 5g systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2079–2091, 2020.

[4] G. Tseliou, F. Adelantado, and C. Verikoukis, "Netslic: Base station agnostic framework for network slicing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3820–3832, 2019.

[5] 3GPP, "TR 38.801. Technical Specification Group Radio Access Network; Study on new radio access technology: Radio access architecture and interfaces release 14," 2017.

[6] C.-Y. Chang, N. Nikaein, R. Knopp, T. Spyropoulos, and S. S. Kumar, "Flexcran: A flexible functional split framework over ethernet fronthaul in cloud-ran," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–7.

[7] S. Gonzalez-Diaz, A. Garcia-Saavedra, A. De La Oliva, X. Costa-Perez, R. Gazda, A. Mourad, T. Deiss, J. Mangues-Bafalluy, P. Iovanna, S. Stracca *et al.*, "Integrating fronthaul and backhaul networks: Transport challenges and feasibility results," *IEEE Transactions on Mobile Computing*, 2019.

[8] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D. J. Leith, "Joint optimization of edge computing architectures and radio access networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2433–2443, 2018.

[9] X. Wang, C. Cavdar, L. Wang, M. Tornatore, H. S. Chung, H. H. Lee, S. M. Park, and B. Mukherjee, "Virtualized Cloud Radio Access Network for 5G Transport," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 202–209, 2017.

[10] R. Ferrus, O. Sallent, J. Perez-Romero, and R. Agusti, "On 5G Radio Access Network Slicing: Radio Interface Protocol Features and Configuration," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 184–192, 2018.

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

[12] A. Aijaz, "Hap – SliceR: A radio resource slicing framework for 5g networks with haptic communications," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2285–2296, 2018.

[13] J. Tang, B. Shim, and T. Q. S. Quek, "Service multiplexing and revenue maximization in sliced c-ran incorporated with urllc and multicast embb," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 4, pp. 881–895, 2019.

[14] J. Tang, B. Shim, T. Chang, and T. Q. S. Quek, "Incorporating urllc and multicast embb in sliced cloud radio access network," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.

[15] 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 *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '16. New York, NY, USA: ACM, 2016, pp. 427–441. [Online]. Available: http://doi.acm.org/10.1145/2999572.2999599

[16] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '17. New York, NY, USA: ACM, 2017, pp. 127–140. [Online]. Available: http://doi.acm.org/10.1145/3117811.3117831

[17] D. Harutyunyan and R. Riggio, "Flexible functional split in 5G networks," in *2017 13th International Conference on Network and Service Management (CNSM)*, nov 2017, pp. 1–9.

[18] C. Y. Chang, R. Schiavi, N. Nikaein, T. Spyropoulos, and C. Bonnet, "Impact of packetization and functional split on C-RAN fronthaul performance," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–7.

[19] A. Maeder, M. Lalam, A. D. Domenico, E. Pateromichelakis, D. Webben, J. Bartelt, R. Fritzsche, and P. Rost, "Towards a flexible functional split for cloud-RAN networks," in *2014 European Conference on Networks and Communications (EuCNC)*, 2014, pp. 1–5.

[20] M. Masoudi, S. S. Lisi, and C. Cavdar, "Cost-effective migration toward virtualized c-ran with scalable fronthaul design," *IEEE Systems Journal*, vol. 14, no. 4, pp. 5100–5110, 2020.

[21] A. Garcia-Saavedra, J. X. Salvat, X. Li, and X. Costa-Perez, "WizHaul: On the Centralization Degree of Cloud RAN Next Generation Fronthaul," *IEEE Transactions on Mobile Computing*, p. 1, 2018.

[22] A. Garcia-Saavedra, X. Costa-Perez, D. Leith, and G. Iosifidis, "FluidRAN: Optimized vRAN/MEC Orchestration," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 1–9.

[23] B. Ojaghi, F. Adelantado, E. Kartsakli, A. Antonopoulos, and C. Verikoukis, "Sliced-ran: Joint slicing and functional split in future 5g radio access networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

[24] 5G-Crosshaul, "H2020 5G-Crosshaul project Grant No. 671598. Detailed analysis of the technologies to be integrated in the XFE based on previous internal reports from WP2/3," 2018.

[25] X. Costa-Perez, A. Garcia-Saavedra, X. Li, T. Deiss, A. De La Oliva, A. Di Giglio, P. Iovanna, and A. Moored, "5g-crosshaul: An sdn/nfv integrated fronthaul/backhaul transport network architecture," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 38–45, 2017.

[26] A. S. Asratian, T. M. J. Denley, and R. Häggkvist, *Bipartite Graphs and their Applications*, ser. Cambridge Tracts in Mathematics. Cambridge University Press, 1998.

[27] *Coordinated Multi-Point in Mobile Communications: From Theory to Practice*. Cambridge University Press, 2011.

[28] A. Mesodiakaki, F. Adelantado, L. Alonso, and C. Verikoukis, "Energy-efficient context-aware user association for outdoor small cell heterogeneous networks," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 1614–1619.

[29] Small Cell Forum, "Small cell virtualization functional splits and use cases," 2016.

[30] 3GPP, "TS 23.501. System Architecture for the 5G System version 15.2.0 (Release 15)," 2018.

[31] A. de la Oliva, J. A. Hernandez, D. Larrabeiti, and A. Azcorra, "An overview of the CPRI specification and its application to C-RAN-based LTE scenarios," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 152–159, 2016.

[32] NGMN Alliance, "NGMN 5G White Paper," Tech. Rep., 2015.

[33] NGMN, "5G Extreme Requirements: Radio Access Network Solutions," Tech. Rep., 2018.

[34] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. USA: W. H. Freeman Co., 1990.

[35] I. I. Cplex, "V12. 1: User's manual for cplex," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.

[36] R. G. Jeroslow, "Trivial integer programs unsolvable by branch-and-bound," *Math. Program.*, vol. 6, no. 1, p. 105–109, Dec. 1974. [Online]. Available: https://doi.org/10.1007/BF01580225

[37] E. Danna, E. Rothberg, and C. Le Pape, "Exploring relaxation induced neighborhoods to improve mip solutions," *Mathematical Programming*, vol. 102, no. 1, pp. 71–90, 2005.

[38] I. Tsompanidis, A. H. Zahran, and C. J. Sreenan, "Mobile network traffic: A user behaviour model," in *2014 7th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, 2014, pp. 1–8.

[39] M. Skorin-Kapov, L.; Matijasevic, "Analysis of qos requirements for e-health services and mapping to evolved packet system qos classes," in *Int. J. Telemed. Appl.*, 2010, pp. 1–18.

[40] Yinan Qi, M. Hunukumbure, M. Nekovee, J. Lorca, and V. Sgardoni, "Quantifying data rate and bandwidth requirements for immersive 5g experience," in *2016 IEEE International Conference on Communications Workshops (ICC)*, May 2016, pp. 455–461.