Dynamic partitioning of radio resources based on 5G RAN Slicing

Massimiliano Maule*, Prodromos-Vasileios Mekikis*,
Kostas Ramantas*, John Vardakas*, and Christos Verikoukis†
*Iquadrat Informatica S.L., Barcelona, Spain
†Telecommunications Technological Center of Catalonia (CTTC/CERCA), Castelldefels, Spain

***Abstract*—Network Slicing (NS) represents a key technology enabler for advanced connectivity and data processing tailored to customers' specific requirements. While significant progress has already been achieved for Core NS, Radio Access Network (RAN) slicing still presents limitations in terms of sharing infrastructure, Service Level Agreement (SLA) guarantees, isolation, resource scheduling and allocation. In this context, this paper firstly introduces a novel slices configuration framework for the 5G New Radio (5G NR) infrastructure able to dynamically migrates the radio resources among the slices, while preserving the Quality of Service (QoS) of the served users. Our solution is detailed illustrated and tested on top of a real case 5G scenario, using a software-based simulator. Finally, this paper investigates the flexibility, scalability, and real-time properties of the proposed method, as required in the future 5G cloud-based architectures.**

***Index Terms*—5G network,5G NR, RAN slicing, Software-defined Network, virtualization**

I. Introduction

The Fifth-Generation (5G) mobile network targets novel use-cases and business models expected to globally convert the role of telecommunications technology in the society. The definition of a service-oriented architecture combined with enhanced computing power dislocated in the network defines an ecosystem involving vertical markets such as automotive, energy, food and agriculture, city management, government, healthcare, manufacturing, public transportation, and many more. This solution will serve a larger portfolio of applications with a corresponding multiplicity of requirements ranging from high reliability to ultra-low latency going through high bandwidth and mobility [1].

As consequence of this service-oriented vision, the Service Providers (SPs) share infrastructures to deliver mobile services to end users, following two modalities: Passive Sharing (PS) consists of sharing network infrastructure such as masts, sites, cabinet, power, cooling, and Active Sharing (AS) for the sharing of Radio Access Network (RAN) elements such as antennas and controllers.

According with the National Regulatory Authority (NRA), the infrastructure sharing is translated in cost saving for the operators [2]. Even though it is complex to define an estimation cost model, some NRAs provided figures of cost saving, as illustrated in Fig. 1:

- Cat 1. Passive sharing cost savings
- Cat 2. Active sharing (excl. spectrum) cost savings
- Cat 3. Active sharing (incl. spectrum) cost savings

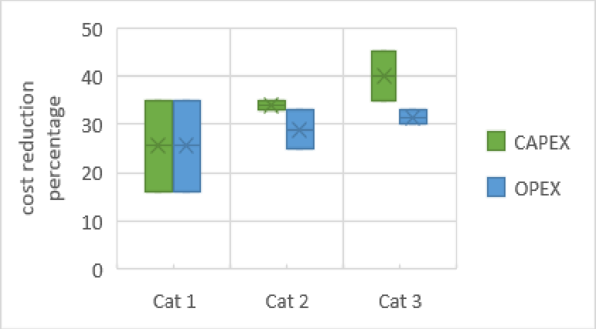


Fig. 1. 5G CAPEX and OPEX estimation

The 5G system has the ambition of responding to the widest range of services and applications in the history of mobile and wireless communications categorized in (i) enhanced mobile broadband (eMBB), (ii) ultra-reliable and low-latency communications (URLLC) and (iii) massive machine-type communications (mMTC).

As early 5G investigation phase, different organizations have been formed to establish research requirements and set the path for the definition of the next generation of mobile solutions. In particular, 3rd Generation Partnership Project (3GPP), during the fourth quarter of 2015, approved a 5G study titled "Study on Architecture for Next Generation Systems", introducing the concept of NS as a promising future-proof framework adhering to the technological and business needs of different industries [3].

NS enables business customers to deploy connectivity and data processing following ad-hoc requirements defined through the operator Service Level Agreement (SLA). Each operator orchestrates its slices as an end-to-end logical network on top of a shared physical infrastructure, spanning over different technology domains (e.g., core, transport and access networks) and administrative domains (e.g., different mobile network operators), while this technology is transparent to the business customer.

While the previous standards were characterized by monolithic network infrastructures of inelastic elements, software, and functionalities, the 5G architecture and NS introduce a novel resource abstraction technique able to overcome the previous legacy systems restrictions. For instance, Network

Function Virtualization (NFV) techniques permit the softwarisation of multiplexing and multitasking radio features, while Software Defined Networking (SDN) enables the sharing of network elements (i.e., Commercial Off-The-Shelf (COTS) equipment) among different tenants.

The decoupling between the virtualised and the physical infrastructure enables scaling and flexibility of the slices, defining an environment where the resources are adapted on demand, introducing new NS challenges in terms of data isolation among multi-tenancy solutions, management of different end-to-end QoS within a slice, network functions optimisation for automatic selection of network resources and functions, monitoring the NS behaviour in a multi-domain scenario, and capability exposure for NS API for slice configuration and interaction.

Different slicing models have been addressed in the 5G architecture of different research projects and publications. In [4], the authors present a functional framework for the management of slicing for a New Generation - Radio Access Network (NG-RAN) infrastructure, based on a dynamic RAN slicing approach. In [5], a dynamic RAN slicing approach for LTE-based systems is presented, and three representative use cases are utilized for isolation, sharing, and customization capabilities.

As novel approach, the use of Reinforcement Learning (RL) methods for NS control and management has gained interest due to their promising performance. In [6], a RL RAN slicing admission control system is presented, where the system learns the services with the potential to bring high profit (i.e., high revenue with low degradation penalty), and hence to be accepted. In [7], the authors designed a slice admission block based on traffic prediction, where the forecasted load is adjusted based on measured deviations.

Even though the aforementioned methods improve the performance of dynamic RAN slicing, there are significant differences between a real case scenario and a simulation environment. First, the acquisition time of network statistics is a critical feature for the dynamism of the solution: a suitable trade-off must be defined between the GET/POST calls towards the controller, and the traffic control plane overloading. Second, only a few NS lifecycle process exists, which are able to embrace multiple scenarios, meaning that the optimal performance is achieved with specific customization of the network. Third, due to customer data privacy policies, only few real data traffic database for training RL and traffic forecasting models are available. As a consequence, the accuracy of the RL model strictly depends from the training set, and unexpected network behaviour, which is common in the Access Part (AP) of the network, is not correctly treated.

This paper presents a novel slicing resources stratification methodology where the degree of sharing for each slice is evaluated according to the type of services, maximum slice capacity, QoS requirements, SLA, and ad-hoc isolation policies. This solution aims to optimize the resource management in the RAN part of the network, create a new level of flexibility where the slice settings are dynamically configured according to the real-time traffic, and an intelligent control of critical traffic peaks through a tiny flexible resource over-provisioning functionality.

The rest of this paper is organized as follows: Section II presents a literature review of the 3GPP specifications of NS and its integration with the 5G architecture. Section III explains in details the proposed resource sharing algorithm. Section IV illustrates the test environment and results. Finally, Section V, provides some concluding remarks.

II. 3GPP Network Slicing and Architecture Integration

A. System Aspects

NS is the embodiment of the concept of running multiple logical networks as virtually independent business operations on a common physical infrastructure in an efficient and economical way [8].

Starting from release 15, various working groups jointly collaborate for a comprehensive slice concept standardization, in particular SA1 (service requirements), SA2 (architecture), SA3 (security) and SA5 (network management).

A NS is a logical end-to-end network that can be dynamically instantiated, on top of the physical architecture. A given User Equipment (UE) is first authenticated through the Access Mobility Function (AMF), and successively accesses different slices linked with the same Access Network (AN), as detailed explain in [9]. Following the standardization specs, three are the basic features for the slice deployment:

- A Network Function (NF), which is a processing function in the network, equipped with 3GPP interfaces.
- A logical connection among the architecture entities with specific network capabilities and characteristics.
- A Network Slice Instance (NSI), which represents a set of NF instances and hardware resources.

The selection of the NSI for the UE is triggered from the AMF during the registration phase, through a cross examination among the UE subscription parameters (S-NSSAIs, PLMN ID, etc.) and the Network Slice Selection Function (NSSF). At this point, the AMF receives a Session Management (SM) message from the UE with the selected S-NSSAIs necessary for the establishment of the Protocol Data Unit (PDU) session. With these parameters, the AMF chooses the Session Management Functions (SMF) necessary for the definition of the User Plane Function (UPF).

As last step, the Network Repository Function (NRF) is used for identify the required NFs using the NSI. The data transmission starts after a PDU session is established with a Data Network (DN) in a NS.

B. Compliant Architecture

This section describes the integration of our solution with the aforementioned slice service-based architecture, as illustrated in Fig. 2.

The system consists in two slices, one eMBB and one mMTC, managed by a single AMF entity, in a shared Access Point (AP). Nevertheless, the solution can be extended to a

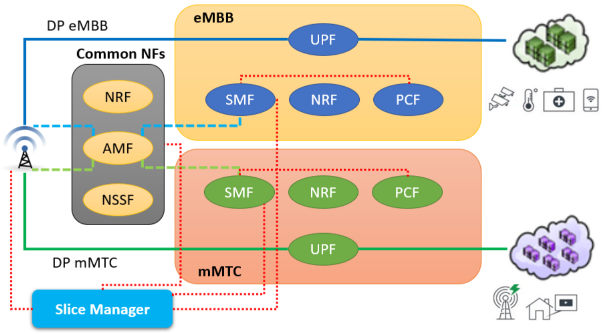


Fig. 2. 5G Compliant System Architecture

higher number of slices, and multiple AMFs can perform handover operations among different AN.

For a better isolation, in addition to the AMF, only the NRF and NSSF are common NFs among the slices, while the SMF, NRF, PCF, and UPF NFs are implemented within each slice. This implementation choice improves decoupling capabilities between the data and control planes, optimize the slice PDU session control and management, and guarantees a dynamic NFs deployment and scaled-up on demand in a completely automated manner.

Our innovative NF for this architecture is represented by the Slice Manager (SM). With the support of RESTful APIs, the SM acquires real-time network informations of each slice and the RAN, elaborates a new parameterization, and post a new slices configuration in the system, following the 5G 3GPP architecture specs. The acquired system statistics are utilized as input for our novel dynamic RAN slicing solution, which performs the resources migration from one slice to another according to each slice SLA.

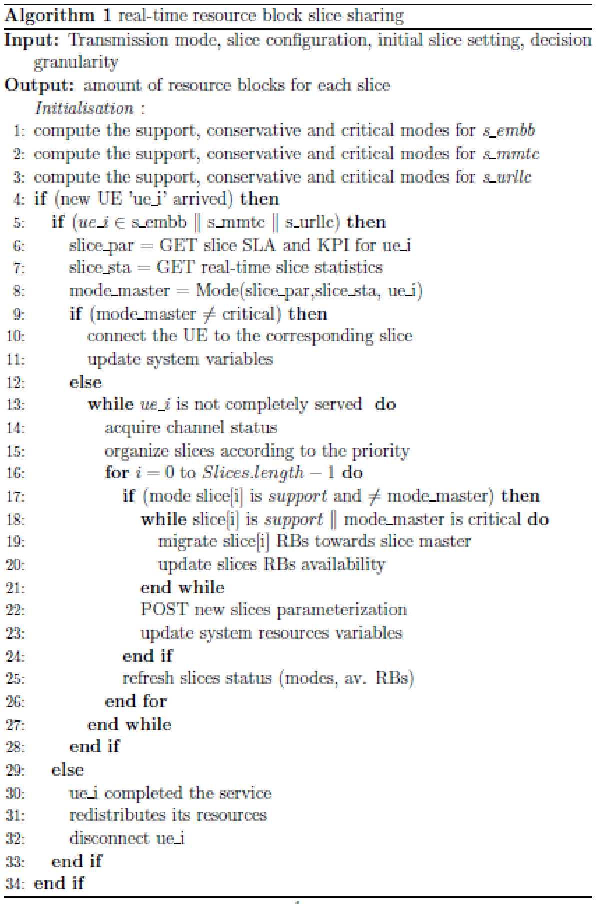
In this paper, the scenario is tested within an simulation environment, equipped with 5G compliant functionalities. The target slice SLA metrics are the average slice data rate and the transmission error probability. Nevertheless, the degree of flexibility of the SM allows the Service Provider (SP) to evaluates other types of metrics, as for example latency constraints, slice priority, packet error probability, advanced resource isolation paradigms, etc.

In the following sections, the paper will use the term *Master* slice for the slice which acquires radio resources (Resource Blocks (RBs)), and *Slave* slice the one which lends RBs.

III. Proposed Algorithm

The proposed algorithm for real-time RBs sharing among the slices is presented in Fig. 3. As initial input parameters, the algorithm receives the Transmission Mode (TM), a set of thresholds (th1, th2) for each slice to compute the *Support*, *Conservative*, and *Critical* modes, the initial bandwidth partition among the slices, and the slice granularity value. Fig. 4 illustrates the initial modes configuration for each slice of the tested scenario.

Once the initialization phase is completed, the system is ready to accept incoming users. When a new UE connection request arrives and belongs to the SP (row 4), the SM acquires from the RAN and the common NFs set, the corresponding SLA and KPI. If the existing amount of available resources satisfies the users' SLA (rows 9-11), the UE is automatically served and the system parameters updated. Otherwise, the slice is in *Critical* mode, it is labelled *Master*, and the SM activates the sharing procedure (row 13).



```
Algorithm 1 real-time resource block slice sharing
Input: Transmission mode, slice configuration, initial slice setting, decision
    granularity
Output: amount of resource blocks for each slice
    Initialisation :
 1: compute the support, conservative and critical modes for s_embb
 2: compute the support, conservative and critical modes for s_mmtc
 3: compute the support, conservative and critical modes for s_urllc
 4: if (new UE 'ue_i' arrived) then
 5:   if ($ue\_i \in$ s_embb || s_mmtc || s_urllc) then
 6:     slice_par = GET slice SLA and KPI for ue_i
 7:     slice_sta = GET real-time slice statistics
 8:     mode_master = Mode(slice_par,slice_sta, ue_i)
 9:     if (mode_master $\neq$ critical) then
10:       connect the UE to the corresponding slice
11:       update system variables
12:     else
13:       while ue_i is not completely served do
14:         acquire channel status
15:         organize slices according to the priority
16:         for $i = 0$ to $Slices.length - 1$ do
17:           if (mode slice[i] is support and $\neq$ mode_master) then
18:             while slice[i] is support || mode_master is critical do
19:               migrate slice[i] RBs towards slice master
20:               update slices RBs availability
21:             end while
22:             POST new slices parameterization
23:             update system resources variables
24:           end if
25:           refresh slices status (modes, av. RBs)
26:         end for
27:       end while
28:     end if
29:   else
30:     ue_i completed the service
31:     redistributes its resources
32:     disconnect ue_i
33:   end if
34: end if
```

Fig. 3. Proposed solution pseudocode

Until the UE SLA is not guaranteed, for each time interval t (defined through the granularity value), the SM compares the *Master* slice SLA with the current quality system indicators (SINR, CQI, packet error rate, etc.), and estimates the amount of RBs needed to fully accomplish the *Master* slice traffic load request. Foreach other slice, if the amount of utilized resources is less or equal to the corresponding *Support* mode, then the slice is labelled *Slave*, meaning that it is qualified to share part of its RBs with the *Master* slice. Otherwise, the slice is in *Conservative* mode, and its amount of RBs must be preserved to guarantee the QoS of its users served.

Labelled all the slices, the SM performs the migration of RBs from the *Slave* to the *Master*, until the *Master* SLA are guaranteed, or the *Slave* slices do not have more available RBs in *Support* mode (rows 18-21). Completed the migration, the SM posts the new slice parameterization in the RAN (row 22), update the system statistics (row 23), and waits for a new incoming UE (back to row 4).

Every time an UE completes the service, its amount of RBs is carefully released and distributed among the slices (row 31), taking into account the traffic load, priority, and operating mode of each slice.

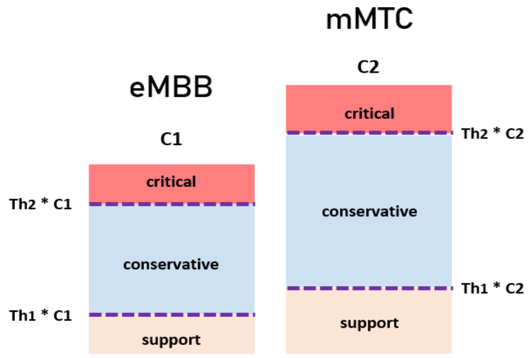


Fig. 4. Initial slice configuration

IV. Performance Evaluation

In the first part of this section, the tools and parameters employed in the two slices scenario are detailed illustrated. Finally, the experimental results are examined, focusing on the flexibility and scalability properties of the proposed solution under appropriate slices configurations.

A. Scenario description

The performance of the slice resource sharing algorithm is evaluated using Matlab R2019a [10]. Our custom-built SM is composed by 4 functions: a *Scenario Manager* where are defined the number of User Equipments (UEs), the SLA for each slice, the number and type of slices, and other settings for external results processing. The *Slice Parameterization* function preserves the status of each slice, compute the slice mode and the amount of RBs at each sharing decision iteration. The *User Class* collects each UE service time duration, required service time, user KPI, and user priority. Finally, the *Physical Downlink Shared Channel (PDSCH) Conformance Test Class* evaluates the optimal slice parameterization taking as input the UEs, slices, and channel characteristics. This class is based on the Matlab 5G-NR Toolbox for the evaluation of the PDSCH channel, following the 3GPP releases [11], [12], [13], [14].

The results are obtained considering the system of 2 slices illustrated in Section II, sharing the same RAN part inside a common cell. The focus of our experiment is the downlink channel, even if the proposed solution is also fully compatible with the uplink channel.

The eMBB slice has the role of *Master* slice, while the mMTC slice represents the *Slave*. Most of the real cases scenarios barely meet the 5G flexibility in terms of maximum transmission bandwidth configuration, due to the limited number of operators with segments of continuous bandwidth of 100 MHz. For our system, the optimal balance between real implementation and 5G performance is achieved using 40 MHz bandwidth at 30 KHz Sub Carrier Spacing (SCS), achieving 106 RBs in the physical layer. As secondary simulator parameters, the cycling prefix is set *Normal*, the code rate for the transport block size is 490/1024, the number of PDSCH transmission antennas is 8, the number of UE receive antennas is 2, and the PDSCH modulation varies between 16 and 64 QAM, according to the Channel Quality Indicator (CQI) value.

To highlight the RBs sharing principle of our RAN NS solution, at the initial phase, the total amount of available RBs is partitioned as 25% for the eMBB slice, and 75% for the mMTC, without any users connected. The *Support* mode, for both the slices, is set between 0 to 40% of the slice size, the *Conservative* mode from 40 up to 70%, and the remaining 30% represents the *Critical* mode. Progressively, one or more high demand data rate users are applied to the *Master* slice, while the *Slave* slice supports a constant Guarantee Bitrate (GBR) users traffic. With the current injected traffic, the SLA of slice eMBB are not guaranteed, and the slice shifts from *Support* to *Critical* mode. On the other side, the *Slave* slice, equipped with a generous amount of RBs, easily guarantees the SLA of its users, remaining in *Support* mode. The SM activates the slice sharing optimization procedure, and gradually moves RBs among the slices, keeping track of the real-time service traffic demand and the KPI of the ongoing served users for each slice.

The previous mechanism represents the baseline of the three different experiments illustrated in the next section. In the first test, the effectiveness of our approach is presented under optimal channel condition. Then, using the same scenario and initial parameterization, the second test illustrates the correlation between the channel quality and our slice partitioning solution. Driven from the output of the previous test, the third test highlights the flexibility capabilities of the sharing algorithm demonstrating how a slight variation of the initial slice modes parameterization can improve the system performance, without requiring any kind of hardware modification or time-consuming software configuration. For the last test, the impact of the frame granularity during the resource slice assignation phase is shown, and a comparative analysis with the previous experiments is discussed.

B. Results

This section presents an analysis of the simulation results under three representative experiments.

The first test illustrates the migration of RBs from the *Slave* to the *Master* slice, under a granularity of 2 frames, when the sharing procedure is activated. After the initialization phase, two users, respectively requiring an average data rate of 45 and 22 Mbps, send a connection request to the slice eMBB. In parallel, a connection request from a user requiring 15 Mbps average data rate arrives at the mMTC slice. While mMTC remains in *Support* mode even after the user connection, the eMBB slice is located in *Critical* mode, and can not accomplish the slice SLA. Under these conditions, the SM activates the sharing procedure, and the capacity of each slice is updated, as shown in Fig. 5.

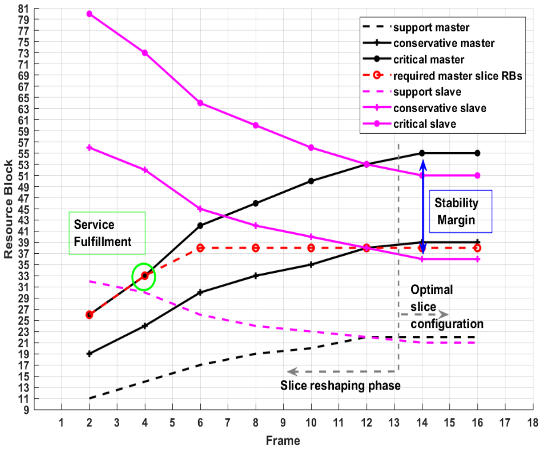


Fig. 5. Slice RBs sharing. Optimal channel condition

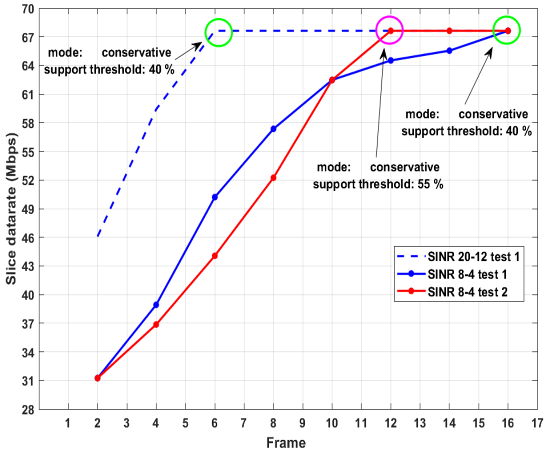


Fig. 6. Performance comparison for different slice parameterization

At each iteration, the available RBs from the *Slave* slice (mMTC) are migrated to the *Master* slice (eMBB). The graph shows how this migration is done gradually, following the granularity of 2 frames. With a SINR of 20 dB, the CQI is optimal, and the system needs only 2 iterations of 2 frames each to balance the slice load (green circle). After this stabilization phase, the SLA is guaranteed and the KPI of the ongoing services are matched. The SM continues to share RBs among the slices until the *Critical* section of the *Master* slice is restored, and the eMBB slice is safely placed within the *Conservative* mode. In this test, the optimal balance of the radio resources is achieved in just 24 frames (240 ms). After this time interval, each slice presents a stable configuration, where the *Support*, *Conservative*, and *Critical* modalities are proportionally re-established.

As introduced with the previous section, test 2 repeats the test 1 for different values of channel SINR classes: 20-12 dB, and 8-4 dB. In Fig. 6, the blue lines illustrate the test 2 behaviour of the eMBB slice (*Master* slice), when the *Support* mode is 40% of the whole slice capacity. Under optimal channel conditions, the system rapidly achieves the stationary point (blue dashed line), while the *Slave* slice is still in *Support* mode after the sharing procedure. For this scenario, a more generous parameterization of the *Slave Support* mode would not improve the whole system capacity and sharing potentials, bringing to a case of resources overprovisioning.

A completely different situation appears when the SINR is between 8-4 dB, and the system parameterization is equal to the test 1 (blue solid line). The stability is not completely achieved due to the little modulation order (16 QAM) and the preserving behaviour of the *Slave* slice towards its served users. To overcome this low-quality parameterization, our solution permits different degrees of configuration of the slices parameters to handle multiple types of scenarios, define ad-hoc flavours of slice isolation, delimit the amount of resources to share, and advanced scheduling capabilities. The test 3, represented with the red line in Fig. 6, shows how our solution responds at low channel quality condition when a more relaxed sharing policy to the *Slave* slice is applied. In this test, the *Slave Support* mode is increased from 40 to 55 % of the whole capacity, raising the amount of RBs that can be shared. With the new parameterization, the eMBB slice reaches the stability condition 40 ms before the test 1, and the slice SLA are completely satisfied, without impacting the KPI of the served *Slave* services.

A bar representation of the slices resources provisioning before and after the execution of the resource optimization algorithm is illustrated in Fig. 7, with the corresponding slice initial and final modes. As previously described, at low SINR the SM must share a higher number of RBs to guarantee the SLA, while under optimal channel conditions, the mMTC slice is still in *Support* mode, meaning that not all its available resources are fully employed.

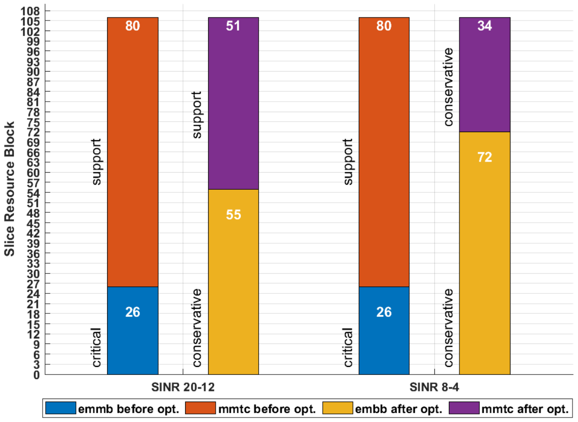


Fig. 7. Slice Resources Partitioning

To conclude the result section, Fig. 8 illustrates another tuning option for the slice sharing algorithm: the *frame granularity*. This parameter defines the window size used by

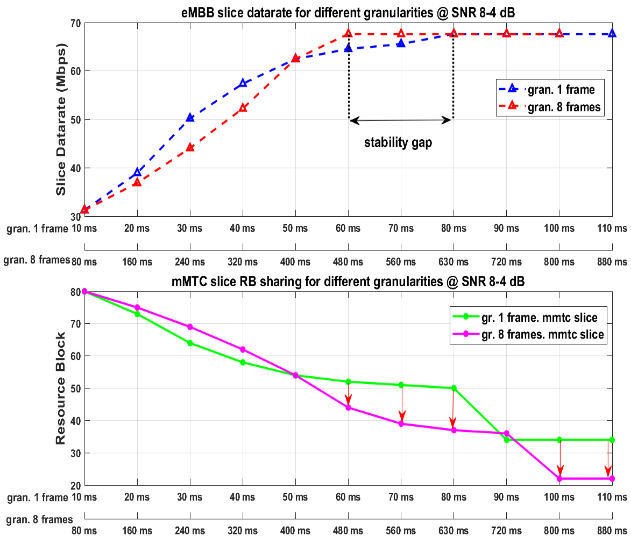


Fig. 8. Performance under different frame granularity

the SM to collect the input information (CQI, SINR values, and services statistics) necessary for the estimation of the appropriate amount of RBs to be shared among the slices. Even though employing a conspicuous number of RBs can result problematic for time sensitive applications, this operation increases the sharing accuracy capabilities of the algorithm.

Fig. 8. top part compares the data rate of slice eMBB for a granularity of 1 frame (blue line) and 8 frames (red line), while bottom part illustrates the amount of RBs released by slice MMTC under the same granularities. For 1 frame granularity, the ”rough” resource sharing estimation brings the slice mMTC to over-release RBs during the first 50 ms, for then converging to the system stability around 60 ms. This behaviour, even though presents higher data rate during the initial phase, can compromise the SLA of the *Slave* slice if the granularity time is too small for correctly evaluates the system resources, channel quality, etc. On the other side, a granularity of 8 frames represents the optimal approach in case there are few available RBs, and the SM must be meticulous during the selection process.

As final observation to conclude the results section, Fig. 8. bottom part, for the first 50 ms of the 8 frames granularity case, the SM releases slowly the RBs for the slice mMTC, at the price of a lower data rate for the slice eMBB (Fig. 8, top part). After this initialization phase, the system acquires sufficient understanding of the surrounding environment, and applies a more relaxed sharing policy, as shown by the red arrows in Fig. 8, increasing the system accuracy and rapid alignment with the required slices SLAs.

V. CONCLUSION

In this work, we propose a novel framework for real-time RAN network slicing resources sharing, where the radio access resources are driven considering the dynamic behavior of the user traffic and SP slice specific SLAs.

The core innovative idea behind our solution is the partitioning of each slice in different modes, according to the traffic load and custom SP specifications. This allows the SM to separate the slices close to resources saturation, from the slices more predispose to release resources without impacting the SLA.

The SM always processes the optimal amount of resources for each slice in order to keep the system balanced and avoiding resources overprovisioning. Moreover, from our analysis, with a variation in terms of the types of resources utilized, it is possible to extend the aforementioned solution even for NS in the Core Network.

The proposed solution is tested for the novel 5G-NR standard. However, its completely technology independent implementation makes it suitable for 5G using Option 3, reflecting the initial launch strategy being adopted by multiple operators [15].

REFERENCES

[1] the 2nd 5G Verticals Workshop, ”5G: Serving Vertical Industries”, Brussels, 9 November 2015
[2] BEREC, ”Report on infrastructure sharing”, 14 July 2018
[3] 5GPPP Architecture Working Group, ”View on 5G Architecture”, Version 2.0, December 2017
[4] Ferrús, Ramon, et al. ”On the automation of RAN slicing provisioning and cell planning in NG-RAN.” 2018 European Conference on Networks and Communications (EuCNC). IEEE, 2018
[5] Chang, Chia-Yu, and Navid Nikaein. ”RAN runtime slicing system for flexible and dynamic service execution environment.” IEEE Access 6 (2018): 34018-34042
[6] Raza, Muhammad Rehan, et al. ”Reinforcement learning for slicing in a 5G flexible RAN.” Journal of Lightwave Technology 37.20 (2019): 5161-5169
[7] Sciancalepore, Vincenzo, et al. ”Mobile traffic forecasting for maximizing 5G network slicing resource utilization.” IEEE INFOCOM 2017-IEEE Conference on Computer Communications. IEEE, 2017
[8] GSM Association, ”An Introduction to Network Slicing.”, London, UK: GSMA (2017)
[9] 3GPP TS 23.501 version 15.2.0 Release 15, ”System Architecture for the 5G System”, 06-2018
[10] M. R2019a, ”https://it.mathworks.com/products/5g.html,” 5G toolbox
[11] 3GPP TS 36.101 version 14.3.0 Release 14, ”LTE, Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception”
[12] 3GPP TS 38.212. ”NR; Multiplexing and channel coding (Release 15).” 3rd Generation Partnership Project; Technical Specification Group Radio Access Network
[13] 3GPP TS 38.213. ”NR; Physical layer procedures for control (Release 15).” 3rd Generation Partnership Project; Technical Specification Group Radio Access Network
[14] 3GPP TS 38.214. ”NR; Physical layer procedures for data (Release 15).” 3rd Generation Partnership Project; Technical Specification Group Radio Access Network.
[15] GSM Association, ”5G Implementation Guidelines,”, London, UK: July 2019