NRflex: Enforcing Network Slicing in 5G New Radio

Karim Boutiba[a], Adlen Ksentini[a], Bouziane Brik[b], Yacine Challal[c], Amar Balla[c]

[a]*Communication Systems Department, EURECOM, Sophia Antipolis, France*
{name.surname}@eurecom.fr
[b]*DRIVE EA1859, University of Bourgogne Franche-Comté, France*
bouziane.brik@u-bourgogne.fr
[c]*Laboratoire des Méthodes de Conception de Systèmes, Ecole nationale Supérieure d'Informatique, Algiers, Algeria*
{y_ challal, a_ balla}@esi.dz

Abstract

The emerging 5G networks promise to support novel network services with different requirements in terms of Quality of Service (QoS), such as low-latency and high bandwidth. Thanks to the network slicing concept, 5G is able to fulfill these different requirements while sharing the same physical infrastructure. Although network slicing is gaining maturity, slicing the Radio Access Network (RAN) is still challenging, particularly with the emergence of new physical features added by 5G New Radio (NR), such as Bandwidth part (BWP) and physical numerology. In this paper, we introduce a new framework, namely New Radio flexibility (NRflex), which addresses the challenge of slicing the RAN in 5G. NRflex provides a solution that dynamically assigns BWP to the running slices and their associated User Equipment (UE), aiming to fulfill the slices' required QoS. Simulation results showed the superiority of NRflex to meet network slice requirements while optimizing the 5G RAN resources, compared to other existing solutions.

Keywords: 5G NR, Network Slicing, Numerology, Bandwidth Parts, Resource Allocation

1. INTRODUCTION

Network slicing is considered as one of the critical enablers of 5G to support a wide range of applications and use cases with different requirements [2]. 5G is assumed to support variant network services that are organized

into three main categories: ultra-reliable and low-latency communication (uRLLC), enhanced mobile broadband (eMBB), and massive machine-type communication (mMTC) [14]. Although network slicing has been widely studied and solutions start to appear, particularly at the transport and core network parts, slicing (sharing) the RAN resources is still challenging [22]. Indeed, with the new features introduced by the 5G NR, further investigation is needed to enforce network slicing at the RAN. 5G NR introduces several new features that can be beneficial for slicing the RAN [20]. Among these new features, we may mention the concept of BWPs. The latter aims to enable flexible assignment and configuration of Physical Resource Blocks (PRBs). BWPs are subsets of contiguous PRBs allocated per UE, i.e., the UE expects to use resources only in a specific part of the bandwidth. Besides, 5G NR introduces the concept of numerology, which uses specific physical layer configuration, mainly Sub-Carrier Spacing (SCS). Unlike 4G, where all the slot-time has the same duration (1 ms), 5G NR by adapting SCS allows reducing the slot-time duration down to 125 microseconds, which can considerably reduce the RAN latency. Consequently, each BWP has its own numerology, enabling more efficient sharing of the spectrum among the heterogeneous services in 5G RAN and hence among slices.

In this paper, we introduce NRflex, a novel framework that dynamically enforces RAN slices in 5G, relying on the concept of BWPs. NRflex addresses the joint PRBs scheduling and allocation problem with mixed numerology in 5G NR, in order to meet the latency requirement of uRLLC services while considering the other type of services (mainly eMBB). NRflex redefines the life-cycle management (LCM) of the RAN slices by leveraging on the Open RAN (O-RAN) architecture [19]. Besides keeping the well-known creation and deletion steps, NRflex introduces five new steps that allow the RAN resources' dynamic sharing for running slices. A preadaptation step runs at the RAN Intelligent Controller (RIC), which dynamically defines the size of a BWP dedicated to a slice according to gNBs' feedbacks. The four other steps run at the gNBs where they periodically allow to: (1) decide which BWP (i.e., slice) a UE has to connect to according to the UE's buffer status and the previous active BWP as UEs cannot use two different BWP in parallel; (2) share the BWP of a slice among its UEs according to the UE channel quality and service requirements. The contributions of this work are manifolds:

- We introduce a novel framework mapped to the O-RAN architecture.
- We introduce a novel definition of RAN slice LCM.

- We propose an algorithm to run at RIC near-time to compute dynamically the size of BWP dedicated to a RAN slice.
- We propose a multiplexing and scheduling algorithm to run at the gNB to periodically decide for each UE the active BWP to use and the amount of PRB assigned to it.

The article is organized as follows: Section 2 provides needed background to understand our approach and state-of-the art solutions to slice the RAN in 5G NR. Our approach is presented in Section 3 and evaluated in Section 4. We conclude the article in Section 5.

2. Background

2.1. Network Slicing

Network slicing has been considered as one of the most important features of 5G and beyond, aiming at meeting a wide range of vertical industry use-cases. In this context, many authors have addressed different aspects of Network slicing, where significant works have focused on slicing the RAN, and specifically on radio resource allocation. In [4], authors propose a network slicing framework combining (i) admission control, (ii) resource allocation, and (iii) user dropping; to satisfy minimum throughput requirements for UEs when many Mobile Network Operators (MNOs) share the network. Work in [16] proposes a two-level MAC (Media Access Control) scheduling framework that can effectively handle uplink and downlink transmissions of network slices of different characteristics over a shared RAN, applying different per slice scheduling policies and focusing on reducing latency for uRLLC services. This work offers the necessary flexibility to dynamically manage radio resources to meet the stringent latency and reliability requirements of uRLLC. In [3], the authors propose a simpler algorithm based only on the estimation of the channel's quality. It allows estimating the number of resources to allocate to each slice, which adds more precision to the system.

2.2. 5G Numerology

In 4G, radio resources are assigned to UEs every 1 ms intervals, namely TTI (Time Transmission Interval). A low-latency-demanding service has to spend at least one millisecond in the queue to get the required radio resources, which may not be tolerable by uRLLC services that require a

RAN latency less than 1ms. In this context, 5G NR numerologies come to make radio resource allocation more flexible. Indeed, 5G NR numerologies reshape radio units in time and frequency. It reduces the TTI to 2, 4, 8, 16 times smaller than the 4G's 1 ms. In 5G NR, each numerology μ is defined by a SCS, and a Cyclic Prefix (CP) [12]. 5G NR Release-15 [8] specifies five main numerologies (μ) and defines an SCS of $15 * 2^{\mu}$ kHz and a slot duration of $1/2^{\mu}$ ms, allowing to reduce the access latency considerably at the RAN. Figure 1 shows 4 BWP, with different numerologies, defined in the time domain (x-axis) and frequency domain (y-axis). The BWPs concept with mixed numerology enables a dynamic allocation of numerology and PRBs. Hence, a UE can benefit from more than one service with different numerology values. Besides, It will reduce the UE power consumption since the UE will only operate on a part of the bandwidth instead of processing all the bandwidth, and the power saving schemes with UE adaptation to BWP bandwidth introduced in 5G NR standards [11] show 16% - 45% power saving gain.

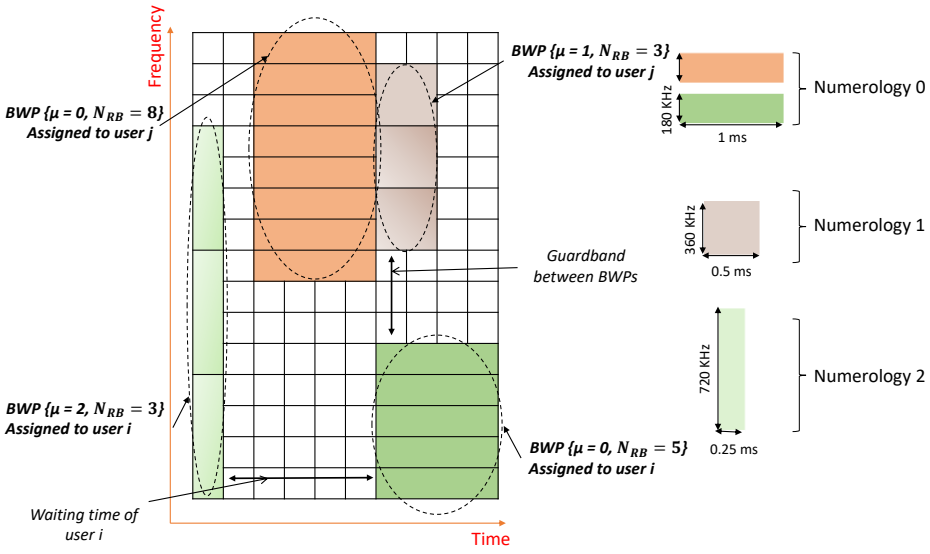


Figure 1: BWPs definition in time and frequency domain with mixed numerologies

Several works in the literature have investigated the usage of flexible numerology and frame structure to optimize service RAN performance, in particular, to reduce latency for uRLLC services. In [21], the authors review

the impact of changing numerology on the throughput and end-to-end latency while taking into account the traffic patterns and the processing delays. Work in [24] explores the potential of optimizing resource allocation with flexible numerology in the frequency domain and variable frame structures in the time domain while considering the presence of services with different types of requirements. The authors used linear programming and Lagrangian duality to design an optimization algorithm to optimize resource allocation in that case.

2.3. 5G Numerology and Network Slicing

Given that flexible numerology plays an important role in network service optimization, especially to reduce latency, several works have considered combining flexible numerology with network slicing. The aim is to satisfy uRLLC slices' low-latency requirements while ensuring eMBB slices throughput requirements.

In [13], the authors study the admission control and network slicing design for 5G-NR systems in which the total bandwidth is sliced (i.e., shared) to support eMBB and uRLLC services. They propose allowing traffic from the eMBB BWP to be overflowed to the uRLLC BWP in a controlled manner by using an efficient iterative algorithm. In [5], the authors introduce a novel 5G slice resource allocation approach that combines the utilization of both complete slots (or PRB) and mini-slots with the adequate physical layer design and service requirement constraints. They advocate for a probabilistic characterization that allows estimating feasibility and characterizes the behavior of the constraints. In [25], the authors propose a self-adaptive flexible TTI scheduling strategy in the eMBB and uRLLC coexistence scenario using Machine Learning. They reduce the delay and packet loss rate of the uRLLC services while guaranteeing the eMBB requirements by dynamically selecting TTI according to traffic load and services requirements.
However, most of the mentioned solutions did not consider a dynamic assignment of BWP as in NRflex; they use the same numerology throughout the network slice life-time. Besides, they consider that a UE is attached only to one slice, which is not realistic as 3GPP allows UEs to be attached to up to 8 network Slices. In contrast, NRflex uses a multiplexing mechanism to allow UEs to be attached to more than one network slice parallelly.

2.4. O-RAN architecture

In parallel with 3GPP groups, a group of network operators as well as big telco names has launched a new initiative known as O-RAN alliance [19], which specifies the new architecture of RAN in 5G and beyond. The alliance objective is to redesign the RAN architecture to unlock it from proprietary-locked solutions to an open system allowing the deployment of novel services and applications on top of the RAN. The O-RAN Alliance is committed to evolving radio access networks by offering open and standardized interfaces for every network component. This transformation will reduce network cost, improve network efficiency and give the agility to import new network capability [18].

O-RAN vision relies on the programmable RAN concept, a new trend that enforces the Software Defined Networking (SDN) concept at the RAN. Programmable RAN introduces the RAN controller's notion that runs different RAN applications, such as mobility management, user scheduling, etc. It enforces the policies issued by these applications on the eNB/gNB under its control via a southbound protocol. An example of a programmable framework is FlexRAN [7] [17], which is composed of a RAN controller and agents deployed on top of OpenAirInterface (OAI) eNB/gNB [15]. Figure 2 illus-

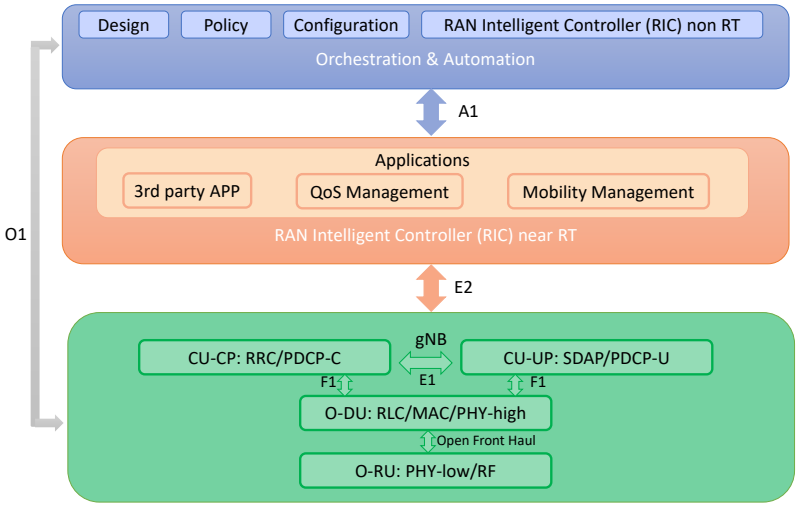


Figure 2: The O-RAN network management architecture and open interfaces

trates the O-RAN architecture that considers a fully functional split of RAN

functions. Indeed, the new trend in terms of 5G RAN architecture is to split the RAN functions, which were constituting monolithic gNB entities, among the Central Unit (CU) Control Plane (CP) and User Plane (UP), the Distributed Unit (DU), and the Radio Unit (RU). CU contains the functions related to higher layers of the RAN, i.e., RRC and PDCP Control Plane, and SDAP and PDCP-U, respectively, for CU-CP and CU-UP. In comparison, O-DU (O-RAN DU) hosts the low-layer functions, i.e., RLC, MAC, and Physical high. It should be noted that CU and DU can run in a virtualized environment, using VM or container technology. Finally, the O-RU (O-RAN RU) hosts all the functions that cannot be virtualized, such as Physical low and RF. The O-RU will be kept in the field and deployed as hardware. Besides, O-RAN adds two layers to the RAN, namely Orchestration and Automation and RIC Near-RT (Near-Real-Time).

Orchestration and Automation: contains the Non-RT (Non-Real-Time) RIC functions, which support intelligent RAN optimization in non-real-time (i.e., greater than one second) relying on data analytics and Machine Learning training/inference solutions. It also manages the network functions such as network design, control, policies, inventory, and configuration.

RIC Near-RT: enables near real-time control (i.e., 10ms to 1s) and optimization of O-CU (CU-CP and CU-UP), O-DU, and O-RU nodes. The RIC Near-RT hosts Applications that use the E2 interface to collect near real-time RAN information (as Key Performances Indicator - KPI) to ease services' deployment using these primitives such as QoS and mobility management services, which can also be guided by the policies provided through the A1 interface by the non-RT RIC.

In summary, O-RAN represents the future of the RAN architecture, therefore NRflex is relying on it to build the components needed to enforce 5G RAN slicing.

3. NRflex framework

NRflex's main idea is to jointly allocate numerology and radio resources for each slice, aiming that UEs can use multiple slices with different requirements. To achieve this objective, NRflex introduces several components that interact together, namely, Bandwidth manager, Pre-processor, and BWP multiplexer, distributed into two different entities, i.e., RIC and gNB. In this section, we detail the NRflex framework's components and algorithms. We

start by defining a RAN slice in NRflex and introducing the revised life-cycle process of RAN slices. Then, we present the components of NRflex, and for each one, we give its role in the O-RAN architecture as well as its related algorithms.

3.1. NRflex slice definition

A network slice is an isolated logical sub-network tailored to fulfill diverse requirements requested by an application. A network slice can be isolated at the RAN level in terms of radio resources (for each slice, an amount of PRBs is allocated) and network functions (each slice has personalized network functions). We consider that a slice is defined at three levels: application, MAC, and Physical level. At the Application level, a slice is an object that contains: the type of the slice (uRLLC or eMBB or mMTC), the requirements of the slice (maximum latency for uRLLC slices, and the desired throughput for eMBB slices), the duration in which the slice is active, the UEs associated with that slice, and the region where the slice is active. These objects are managed by a high-level entity, the RAN Controller, which divides the gNB's bandwidth among slices. The RAN Controller increases/decreases the number of PRBs for each slice according to specific KPIs sent by the gNBs. At the MAC level, a slice is a set of Logical Channels (LC) belonging to different UEs. As the gNB can have more than one data LC for a UE, the latter can be associated with many slices, suitable for multi-service applications. The LCs are scheduled in a way to fulfill services' requirements with a minimal number of PRBs. At the Physical level, a slice is considered as a BWP associated with numerology according to the slice type. For instance [25], numerology 0 for eMBB slices, numerology 1 for uRLLC slices with Max Latency > 5 ms, and numerology 2 for uRLLC slices with Max Latency < 5 ms. The amount of PRBs in a BWP is computed to fulfill the slice requirements (see Section 3.2). For each UE, the BWPs with the same numerology are aggregated to reduce the complexity of the Multiplexing step (see Section 3.3).

3.2. Network Slice life-cycle in NRflex

According to 3GPP [9] an end-to-end network slice LCM is composed by four phases (Figure 3): Preparation phase, Activation phase, Run-time phase and Decommissioning phase. The preparation phase is dedicated to the description of the network slice components and attributes using a blueprint. The activation phase consists of configuring and instantiating the network

slice resources, for example, instantiate the virtual resources (Virtual Machines or containers) and reserve physical resources such as radio, network, and compute resources. The run-time phase covers the supervision and the modification of the resources dedicated to the slice, such as increasing the radio resources or computing resources. Finally, the decommissioning phase consists of releasing the resources which have been reserved to the network slice. In NRflex, we revisit the run-time phase, focusing on the RAN part of the end-to-end network slice, or the RAN slice.

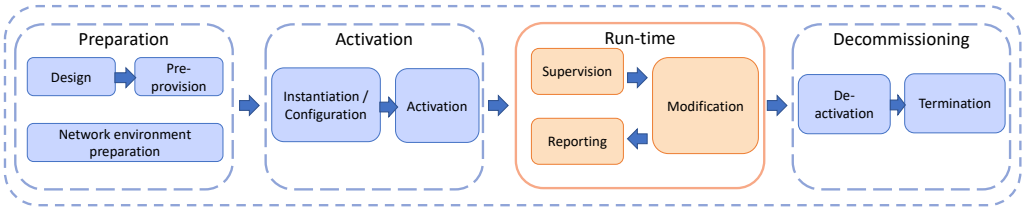


Figure 3: Lifecycle phases of a Network Slice Instance [9]

Since gNB traffic load and UEs channel quality can change over time, NRflex adjusts the slice performance by adding or removing PRBs to running slices. Besides, a slice can not be active during all UE's slots. Indeed, a UE can use only one numerology at a given time (for example, a UE belonging to both an uRLLC slice and an eMBB slice can not use both of them at the same slot). Therefore, the network slice lifecycle at run time is detailed in NRflex as follows (Figure 4):

Pre-processing: in this step, the amount of PRBs needed to ensure the slice requirement is calculated for each slice. At the end of this step, a sorted list of LCs according to deadline criterion is provided to the Multiplexer (at the MAC layer level); each LC has a pre-allocated amount of PRBs.

Multiplexing: knowing that only one BWP is active for a UE at a

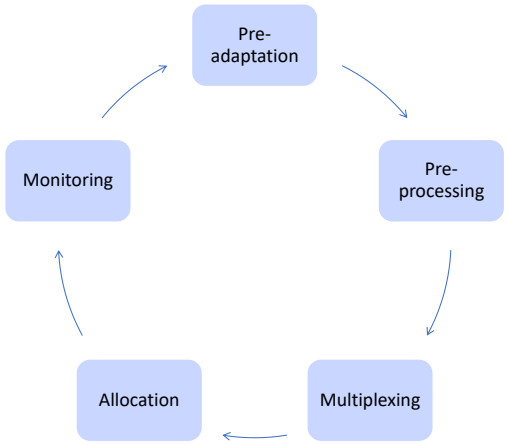


Figure 4: Runtime slice lifecycle in NRflex

given time, and a UE can be associated with more than one slice type (more than one configured BWPs), a multiplexing step is mandatory. This step is needed to select which BWP will be active in the next slot by considering the different slices' requirements to which the UE belongs. At the end of this step, LCs not supposed to transmit in the next slot will be removed from the sorted list, which is transferred to the MAC Scheduler.

Allocation: in this step, the MAC scheduler will allocate PRBs to different LCs by considering the result of the preprocessing and multiplexing steps.

Monitoring: in this step, two KPIs are computed. They are used in the life cycle management of a RAN slice, $throughput\_success\_rate$ measures the eMBB slice performance (best value 1, worst value 0), and $deadline\_failure\_rate$ measures the uRLLC slice performance (best value 0, worst value 1). They are computed at the gNB level and used by RIC to make decisions to add more resources to a slice. If its value has not reached targeted value then more resources need to be assigned to the slice. The $throughput\_success\_rate$ takes into account the actual throughput, the desired throughput, and the queue size. Its value reaches 1, if the actual throughput reaches the desired throughput or the data queue is empty. $deadline\_failure\_rate$ monitors the PDUs that exceed their deadline. A value equals 0 means that all the slice's PDUs respect their max latency.

Pre-adaptation: in this step, the amount of PRBs dedicated to each slice is computed by relying on the previous pre-allocation results and the KPIs, aiming at adjusting the slice performance.

This process is executed in an infinite loop until the slice deletion.

3.3. NRflex and the O-RAN architecture

As stated earlier, to support the deployment of NRflex components to ensure 5G NR network slicing, we consider the O-RAN architecture. In figure 5, we illustrate the different components' interaction in the O-RAN model. Those added by NRflex are highlighted in orange. This figure is slightly different than Figure 2 as we group O-CU, O-DU, and O-RU functions under the same components (i.e., gNB) to ease the figure readability. Also, there is a new entity, Slice Orchestrator (SO), which is not under the scope of O-RAN, but it is an essential element as it manages the LCM of the end-to-end network slices, including the RAN (or RAN slice). In the following, we describe the role of each components highlighted in the figure.

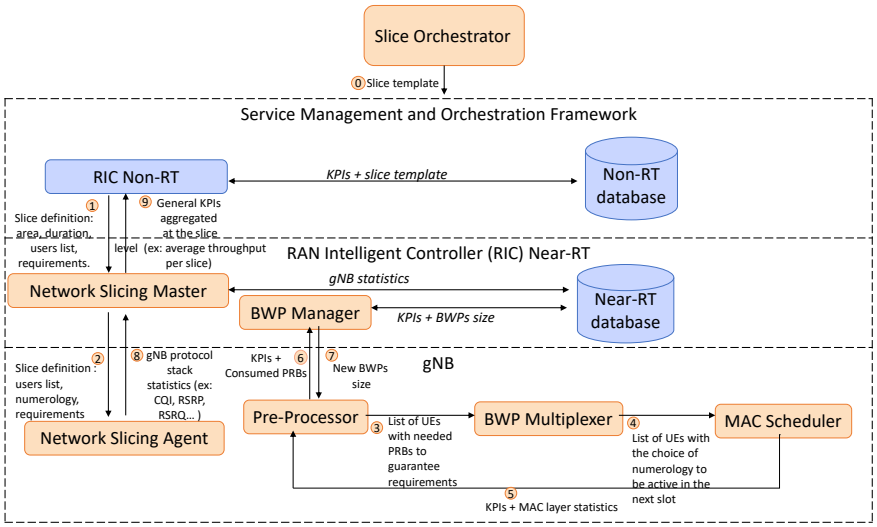


Figure 5: NRflex components mapped to the O-RAN architecture

3.3.1. The Slice Orchestrator (SO)

This component is the entry of the system. It is not defined by the O-RAN architecture. Its role is to provide the end-to-end slice template to be used by the tenant to define the network slice characteristics and its components,

and manages the end-to-end slice LCM (Figure 3). Although the end-to-end slice includes other components to deploy, such as the Core Network, the applications, and the transport network, we focus only on the RAN part in this work. Therefore, the SO enforces the RAN part of the network slice by communicating with the RIC Non-RT module.

3.3.2. The RIC Non-RT

In our system, the RIC Non-RT launches the slice creation and deletion procedures and gathers general performance KPIs from the RAN as throughput, latency, bandwidths, etc., via the O-RAN O1 interface.

3.3.3. The RIC Near-RT:

O-RAN RIC Near-RT allows flexible onboarding of third-party control-applications as QoS enforcement, connectivity management, and handover control. NRflex adds two control-applications:

- **The Network slicing master:** it manages the slices among many gNBs. It can be instantiated for a region and checks for UEs association with their slices. It collects gNB statistics used by O-RAN analytic applications.
- **The BWP manager (algorithm 1):** it uses O-RAN E2 interface to re-adapt the BWP size for each slice according to the schedulers' feedbacks (i.e., the KPIs of the Monitoring step) and the consumed PRBs of each slice. We consider two KPIs: $throughput\_success\_rate = \frac{\sum_{i=1}^{slots} Ti \mid Qi}{number\ of\ slots}$ where $T_i = 1$ if the scheduled bytes at slot i exceed the throughput per slot, and $Q_i = 1$ if the traffic queue is empty (all traffic scheduled); $deadline\_failure\_rate = \frac{D}{number\ of\ slots}$ where D is number of times a PDU (Protocol Data Unit) had exceed its deadline. $N_{embbslices}$ ($N_{urllcslices}$) is the number of eMBB (uRLLC) slices, respectively. $NewPRB_j$ represents the number of PRBs to be allocated for slice j (BWP size of slice j) in the next time interval, $consumedPRB_j$ represents the average number of used PRBs by slice j during the past time interval.

3.3.4. The gNB

NRflex uses the following components at the gNB:

Algorithm 1 BWP manager

for $j = 1, 2, \ldots, N_{embbslices}$ **do**
  if $throughput\_success\_rate_j < 1$ **then**
    $NewPRB_j = consumedPRB_j + 2$
  end if
end for
for $i = 1, 2, \ldots, N_{urllcslices}$ **do**
  if $deadline\_failure\_rate_i > 0$ **then**
    $NewPRB_i = consumedPRB_i + 2$
  end if
end for
Send the new BWP sizes to the gNB Pre-processor

- **The Network slicing agent:** it manages slices at the MAC level, receives slices configuration from the Network slicing master, and stores the information to be used by the pre-processors and the MAC scheduler.
- **Pre-processor:** it executes for each slice an instance of the pre-processor. We considered two types of pre-processors:
  - **The eMBB pre-processor (algorithm 2):** it takes data from the LCs associated with the slice instance and calculates the effective number of PRBs needed to send the buffered data. It considers the buffer size, the CQI (Channel Quality Indicator) of each UE, and the maximum number of PRBs allocated by the BWP manager that achieves the desired throughput. The α_{UE}

Algorithm 2 eMBB pre-processor for slice i

Sort LCs according to the weight α_{UE}
for $LC = 1, 2, \ldots, N_{LCs}$ **do**
  $R \leftarrow bytes\_to\_RBs(CQI_{UE}, \alpha_{UE} * req_{LC})$
  $N \leftarrow bytes\_to\_RBs(CQI_{UE}, queue\_size_{LC})$
  $prealloc\_res_{LC} \leftarrow \min(R, N)$
  Update the available PRBs for slice i
end for

is a weight associated with a UE indicating how many eMBB

slices of this UE were discriminated in the Multiplexing stage; it is updated in the Multiplexing algorithm. Based on this variable, NRflex controls the required amount of throughput to achieve in the current slot to guarantee the desired throughput in a 1s time interval. $bytes\_to\_RBs$ is a helper function that returns how many PRBs are required to carry the amount of data with the CQI passed in parameters.

- **The uRLLC pre-processor (algorithm 3):** it sorts LCs according to their head PDU deadline, then computes PRBs amount needed to only transmit PDUs that meet their deadline in the near future.

Algorithm 3 uRLLC pre-processor for slice i

Sort LCs according to the remaining time of the first pdu
for $LC = 1, 2, \ldots, N_{LCs}$ **do**
    $size \leftarrow 0$
    while $remaining\_time(pdu_j) <= 2TTI_\mu$ **do**
        $size \leftarrow size + pdu\_size_j$
    end while
    $prealloc\_res_{LC} \leftarrow bytes\_to\_RBs(CQI_{UE}, size)$
    Update the available PRBs for slice i
end for

- **The BWP multiplexer (algorithm 4):** it selects which BWP to activate for each UE and generates DCI (Downlink Control Indicator) [23] to be sent to the concerned UEs indicating the decision.

Algorithm 4 BWP multiplexer

```
1: for UE = 1, 2, ..., N_UEs do
2:     for LC = 1, 2, ..., N_LCs do
3:         if μ_LC = 2 and sched_LC > 0 then
4:             if α_UE < α_max1 then
5:                 activate numerology 2 for UE
6:                 α_UE ← α_UE + 1
7:                 break
8:             end if
9:         else
10:            if μ_LC = 1 and sched_LC > 0 then
11:                if α_UE < α_max2 then
12:                    activate numerology 1 for UE
13:                    α_UE ← α_UE + 1
14:                    break
15:                end if
16:            else
17:                if sched_LC > 0 then
18:                    activate numerology 0 for UE
19:                end if
20:                α_UE ← 1
21:                break
22:            end if
23:        end if
24:    end for
25: end for
```

This algorithm prioritizes uRLLC traffic as it has to respect a deadline. α_{max_1} and α_{max_2} are thresholds to avoid starvation of eMBB traffic. $sched_{LC}$ is the amount of data to schedule in the next slot (the result of the pre-processing step).

- **The MAC Scheduler:** it allocates PRBs for each UE based on how many PRBs it needs and how many are available for each slice.

4. Performance Evaluation

In order to evaluate NRflex framework algorithms, we used a reliable 5G simulator based on Matlab that supports different numerology (table 1) and BWPs with dynamic scheduling. Note that this simulator is an improved version of the one used in [3]; it includes 5G NR features. We validated the simulator using the 3GPP 5G NR simulator [1]. Both of them gave the same throughput with different configurations (Bandwidth, numerology, etc.). Table 1 describes the available system bandwidths in the simulation environment. These bandwidths will be divided among different BWPs.

We have simulated 3 scenarios: (1) the evolution of the number of users over time and its impact on the KPIs as well as the PRBs allocated for each slice; (2) the variation of the users' number as well as the traffic load and their impact on the KPIs and the PRBs allocated for each slice; (3) the variation of the traffic load and its impact on the numerology selection. We have run the simulation for 100 iterations. Each value presented in the figures represents the average. We did not include the minimum and maximum values as they are very close to the average, and adding them will reduce the figures' readability. For eMBB slices, we compared our algorithm with

Table 1: 5G NR parameters [10]

Numerology	System available BandWidth (MHZ)	Number of PRBs
0	20	106
1	40	106
2	80	107

the one introduced in [3] (called standard solution for eMBB slices in the rest of the paper), which shares the same idea with other eMBB resource allocation algorithms (as [7; 6]); i.e., they use the slice throughput constraint to compute the required amount of PRBs. To recall, NRflex, in addition to throughput, considers the queue size (see algorithm 2) to allocate only the needed PRBs for the slice. Thus, we minimize the number of PRBs allocated for eMBB slices while respecting their throughput requirements. For uRLLC slices, we compared our algorithm with the Fair Proportional Scheduling algorithm combined with a resource allocation strategy that allocates the amount of PRBs needed to schedule all uRLLC traffic first (called standard solution for uRLLC slices in the rest of the paper). NRflex (algorithm 3) considers PDUs' deadline to allocate only the needed PRBs to schedule PDUs

that will exceed their deadline shortly. We have simulated four slices with different requirements (table 2) and a random CQI which takes values in [12-15] interval for each UE, indicating a medium to a good channel condition.

We specified the arrival rate and the packet size of the traffic associated with each user's slice type in table 3. uRLLC slice traffic is characterized by small data chunks with high frequency, while data chunks' sizes are big with low frequency sending for eMBB slice traffic. Moreover, the UEs can join more than one slice. In all scenarios, we are associating each UE to two slices (one embb and another one uRLLC). However, UEs cannot serve two slices at the same time (i.e. same slot in ms granularity) when the slices use different numerology (5G NR physical layer constraint). We are using the term slot relative to the slice numerology, if numerology n is selected then slot duration is 2^{-n} ms. At t=0s, the system includes 10 UEs among them 5 UEs are

Table 2: slices requirements

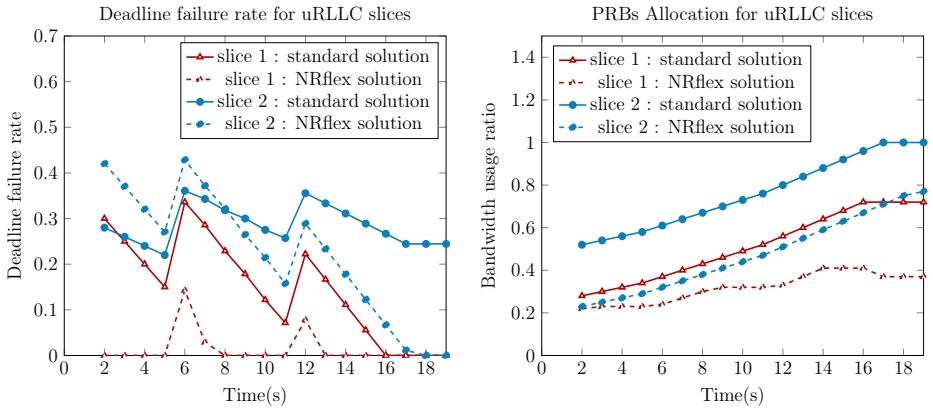
slice	Max Latency	desired throughput per UE
embb1	-	0.9 mbps
embb2	-	1.5 mbps
urllc1	5 ms	-
urllc2	1 ms	-

Table 3: traffic simulation parameters

Slice	Inter-arrival time	Packet size
urllc1	4 ms	800 Bytes
urllc2	2 ms	400 Bytes
embb1	70 ms	4596 Bytes
embb2	70 ms	6516 Bytes

connected to each slice tuple ((urllc1,urllc2,embb1), (urllc1,urllc2,embb2)). At t=5s and t=11s, 2 more UEs connect to each slice tuple; i.e., at t=6s, the system includes 14 UEs, and at t=12s, 18 UEs.

Figure 6a shows the evolution of the $deadline\_failure\_rate$ over time as the number of UE increases. At t=6s and t=12s, the $deadline\_failure\_rate$ increases since 4 more UEs have joined the system. However, the preadaptation phase in NRflex decreases the $deadline\_failure\_rate$ in an incremental way. Hence, it is important to recall that the preadaptation phase is realized



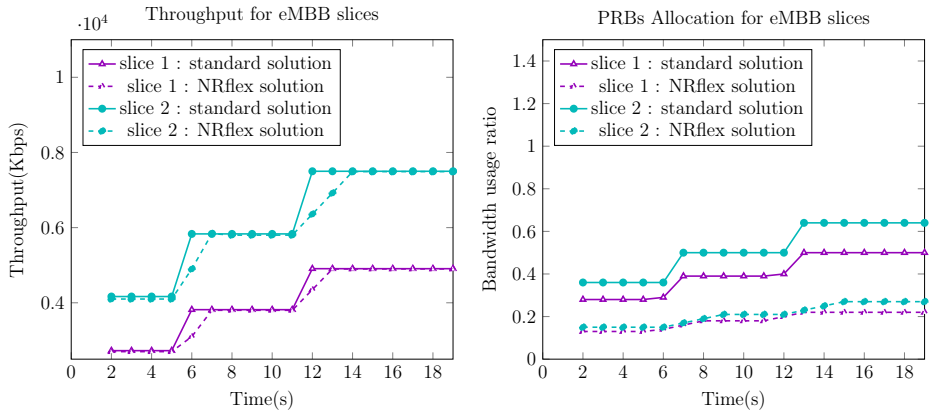
(a) (b)

Figure 6: uRLLC performance and resource allocation over time

by RIC that computes dynamically the size of a BWP dedicated to a slice according to gNB's feedbacks (algorithm 1).

Figure 6b shows the evolution of the number of allocated PRBs for each slice. We note that for the first slice, which requires a latency of 5 ms, NRflex ensures a lower $deadline\_failure\_rate$ (Figure 6a) than the standard solution. Moreover, NRflex adapts itself quickly to the new cell load by adding more PRBs to the slice. We also observe that NRflex meets the slices' required latency deadline with a smaller number of PRBs (Figure 6b). For the second slice, which requires a latency of 1 ms, we remark that, at t=17s (Figure 6b), the standard solution can not reduce the $deadline\_failure\_rate$ (Figure 6a) as it consumed all the bandwidth dedicated for numerology 2 (table 1). In comparison, NRflex is able to reduce the $deadline\_failure\_rate$ to zero after 5s of adaptation with the same bandwidth size. We can also see that our NRflex allocates lesser PRBs (Figure 6b) to satisfy the slice latency requirement. Thus, we can argue that combining deadline aware scheduling with our resource allocation strategy can achieve very low latency while using lesser radio resources.

Figure 7a shows the throughput evolution over time. We remark that both approaches achieve the same throughput, which is different from the desired throughput. We argue this because the practical throughput is calculated on the gNB, while the desired throughput is just a theoretical throughput. In figure 7b, we see that NRflex consumes lesser PRBs to meet the same throughput, compared to the standard solution. As the number of UEs



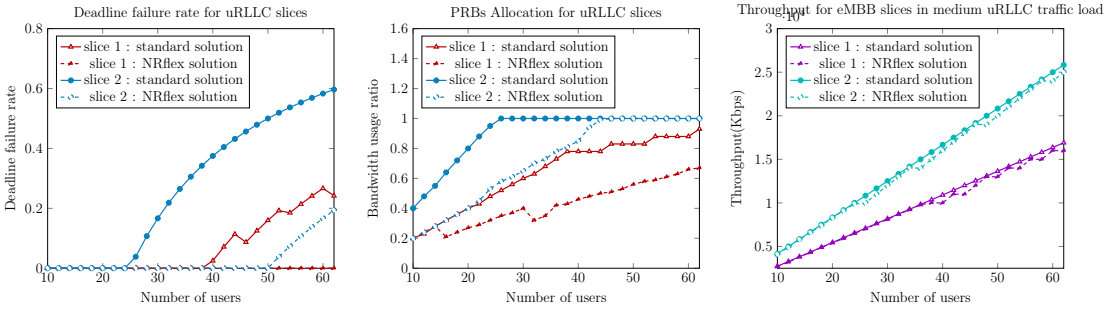
(a) (b)

Figure 7: eMBB performance and resource allocation over time

increases, the difference between the two approaches increases in terms of used PRBs. We explain this by the fact that UEs do not require all the PRBs allocated by the standard solution at each slot; hence the needed PRBs depend on the state of traffic queues (LCs) of the UEs and their CQI. UEs, which have a good CQI and a small amount of data in their LC, do not need many PRBs even if the desired throughput is significant. That proves the pertinence to calculate the number of needed PRBs from an entity close to the gNB, which is aware of the state of the queues (the CQI of the UEs) and has a global vision to manage the radio resources and distribute them among the RAN slices. Therefore more UEs can be connected, and hence more RAN slices can be created.

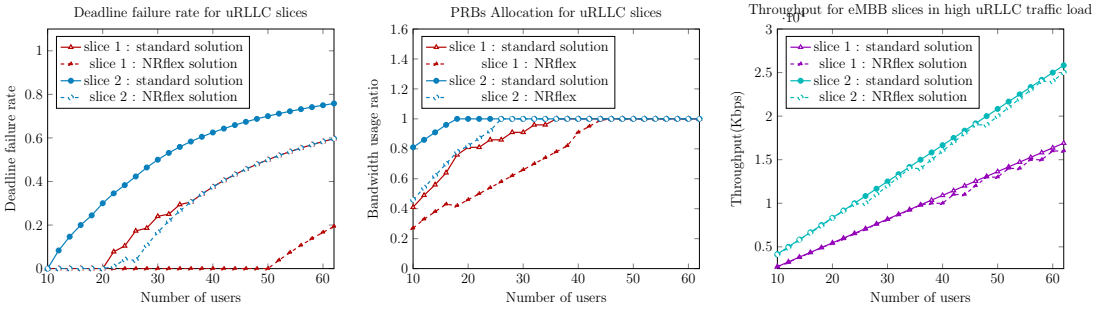
It is worth noting that NRflex may introduce overhead due to the exchanged messages ($number\ of\ messages * message's\ size$) between RIC and gNBs. Each 1s interval, gNB sends a message containing a list of slices with the average of used PRBs and calculated KPIs (i.e., $deadline\_failure\_rate$ and $throughput$), and receives a message containing a list of slices with their new BWP configuration. However, these messages' impact is very low in terms of needed bandwidth, as their size is very small. Hence, only a few bytes are transmitted periodically. In the second scenario, we varied the uRLLC traffic load to see the limits of both approaches. By traffic load, we mean the inter-arrivals time of packets and packet size. Figure 8a shows $deadline\_failure\_rate$ evolution according to the number of UEs (UEs are equally distributed between the two slices) with a medium uRLLC traffic

load for each UE. NRflex can handle up to 50 UEs in numerology 2 and up to 60 UEs in numerology 1 with a $deadline\_failure\_rate$=0 (no packet is lost due to deadline exceeded). In contrast, the standard solution can handle up to 25 UEs in numerology 2 and 40 UEs in numerology 1 before the $deadline\_failure\_rate$ increases. For high uRLLC traffic load (Figure 9a), NRflex can handle up to 50 UEs that require a 5 ms latency and 20 UEs that require a 1ms latency, while the other approach can handle only 20 UEs and 10 UEs, respectively. At the same time, NRflex ensures that the eMBB throughput is respected for both medium and high uRLLC traffic loads (Figures 8c, 9c).



(a) (b) (c)

Figure 8: Medium uRLLC traffic load



(a) (b) (c)

Figure 9: High uRLLC traffic load

Table 4: uRLLC traffic load

Index	average Inter-arrival time (ms)	Average packet size (Bytes)
1	5	200
2	4	280
3	3	360
4	2	440
5	1	520

In the third scenario (Figure 10, we varied the uRLLC traffic load (table 4) to show the impact of uRLLC slices on eMBB slices, when uRLLC traffic becomes dominant. As the traffic becomes more intensive, it requires more active slots in the uRLLC numerology; hence less slots for the eMBB numerology. (only one numerology active at a time slot). We argue this by the fact NRflex prioritizes uRLLC traffic (algorithm 4). We remark that, from index = 3 (see table 4), uRLLC traffic starts impacting eMBB traffic. However, the results show that even with the higher loads, eMBB slices are still getting served (eMBB numerology is selected).

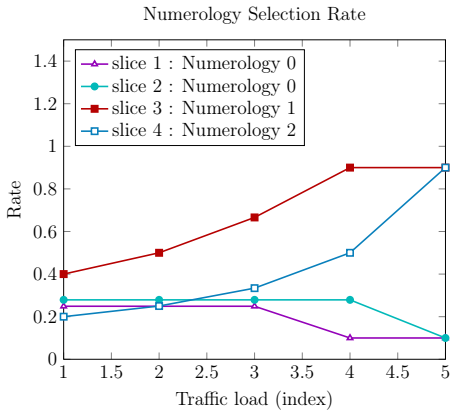


Figure 10: Numerology selection over uRLLC traffic load

5. CONCLUSION

In this paper, we have introduced a 5G NR Network Slicing framework aligned with O-RAN architecture. This framework, namely NRflex, enables UEs to benefit from multi-service applications and leverages 5G NR numerologies to achieve uRLLC services latencies while respecting eMBB

services throughput. NRflex is divided into two parts: (1) one executed by RIC near-RT to dynamically compute the size of BWP to be dedicated to a slice; (2) one executed by gNBs that periodically decides for each UE the active BWP to be used and the amount of PRBs assigned to it. Numerical results show that NRflex succeeds in meeting services requirements, scheduling more UEs, and optimizing PRBs allocation compared to the standard solution. Besides, NRflex architecture offers modularity that allows a flexible modification of the different introduced entities, i.e., schedulers, the BWP manager, preprocessors, etc.
As future work, we intend to implement NRflex in OpenAirInterface (OAI) 5G [15] to test it in the real deployment.

Acknowledgment

This work was partially supported by the European Union's Horizon 2020 Research and Innovation Program under the 5G!Drones project (Grant No. 857031).

References

[1] 3GPP 5G tools, 2021. 5g nr throughput calculator. URL: `https://5g-tools.com/5g-nr-throughput-calculator/`.

[2] 5G Slicing Association, 2020. Categories and service levels of network slicing white paper. White Paper, March 2020 .

[3] Bakri, S., Frangoudis, P., Ksentini, A., 2019. Dynamic slicing of ran resources for heterogeneous coexisting 5g services. GLOBECOM 2019, IEEE Global Communications Conference .

[4] Caballero, P., et al., 2018. Network slicing for guaranteed rate services: Ad-mission control and resource allocation games. IEEE Transactions on Wireless Communications .

[5] Escudero-Garzas, J.J., Bousono-Calzon, C., Garcia, A., 2019. On the feasibility of 5g slice resource allocation with spectral efficiency: A probabilistic characterization. IEEE Access .

[6] Foukas, X., Marina, M., Kontovasilis, K., 2017. Orion: Ran slicing for a flexible and cost-effective multi-service mobile network architecture. The 23rd Annual International Conference on Mobile Computing and Networking (MobiCom '17) .

[7] Foukas, X., Nikaein, N., Kassem, M.M., 2016. Flexran: A flexible and programmable platform for software-defined radio access networks. CONEXT 2016, 12th International on Conference on Emerging Networking Experiments and Technologiess .

[8] 3rd Generation Partnership Project (3GPP), 2018a. Physical channels and modulation. 3GPP TS 38.211 version 15.3.0 Release 15 .

[9] 3rd Generation Partnership Project (3GPP), 2018b. Study on management and orchestration of network slicing for next generation network (release 15). 3GPP TS 28.801 version 15.1.0 Release 15 .

[10] 3rd Generation Partnership Project (3GPP), 2019a. 5g nr user equipment (ue) radio transmission and reception; part 1: Range 1 standalone. 3GPP TS 138 101-1 V15.5.0 Release 15 .

[11] 3rd Generation Partnership Project (3GPP), 2019b. Study on user equipment (ue) power saving in nr. 3GPP TR 38.840 V16.0.0 Release 16 .

[12] 3rd Generation Partnership Project (3GPP) TR 38.804, 2017. 5g; study on new radio access technology; radio interface protocol aspects. 3Gpp Tr 38.804 Release 14.

[13] Ha, V.N., Nguyen, T.T., Le, L.B., Frigon, J.F., 2019. Admission control and network slicing for multi-numerology 5g wireless networks. IEEE Communications Letters .

[14] ITU-R, 2015. "framework and overall objectives of the future development of imt for 2020 and beyond m.2083 .

[15] Kaltenberger, F., De Souza, G., Knopp, R., Wang, H., 2019. The OpenAirInterface 5G new radio implementation: Current status and roadmap, in: WSA 2019, 23rd ITG Workshop on Smart Antennas, Demo Session, 24-26 April 2019, Vienna, Austria, Vienna, AUTRICHE. URL: http://www.eurecom.fr/publication/5822.

[16] Ksentini, A., Frangoudis, P.A., PC, A., Nikaein, N., 2018. Providing low latency guarantees for slicing-ready 5G systems via two-level MAC scheduling. IEEE Network, Vol.32, N.6, November/December 2018 URL: http://www.eurecom.fr/publication/5310, doi:http://dx.doi.org/10.1109/MNET.2018.1800005.

[17] Ksentini, A., Nikaein, N., 2017. Toward enforcing network slicing on RAN: flexibility and resources abstraction. IEEE Commun. Mag. 55, 102–108.

[18] ORAN Alliance, 2020a. O-ran use cases and deployment scenarios: Towards open and smart ran. White Paper, February 2020 .

[19] ORAN Alliance, 2020b. Operator defined next generation ran architecture and interfaces. URL: https://www.o-ran.org/.

[20] Parkvall, S., Dahlman, E., Furuskär, A., Frenne, M., 2017. Nr: The new 5g radio access technology. IEEE Communications Standards Magazine .

[21] Patriciello, N., et al., 2018. 5g new radio numerologies and their impact on the end-to-end latency. IEEE CAMAD .

[22] Schmidt, R., Nikaein, N., 2020. Radio access network slicing system. Chapter book of "Wiley 5G Ref", 2020. URL: http://www.eurecom.fr/publication/6101.

[23] sharetechnote, 2020. Dci. URL: https://www.sharetechnote.com/html/DCI.html.

[24] You, L., Liao, Q., Pappas, N., Yuan, D., 2018. Resource optimization with flexible numerology and frame structure for heterogeneous services. IEEE Communications Letters .

[25] Zhang, J., Xu, X., Zhang, K., Zhang, B., Tao, X., Zhang, P., 2019. Machine learning based flexible transmission time interval scheduling for embb and urllc coexistence scenario. IEEE Access .