# Extending QoS-aware Scheduling in ns-3 5G-LENA: A Lyapunov based Solution

Neco Villegas, Ana Larrañaga, Luis Diez, Katerina Koutlia, Sandra Lagén, Ramón Agüero
[1]Universidad de Cantabria
Santander, Spain
[2]Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA)
Barcelona, Spain
villegasn@unican.es,ana.larranaga@cttc.cat,ldiez@tlmat.unican.es
kkoutlia@cttc.cat,sandra.lagen@cttc.es,ramon@tlmat.unican.es

## ABSTRACT

In this paper, we integrate a scheduling solution based on the Lyapunov framework within the ns-3 5G-LENA extension. This approach allows implementing policies that consider the fulfillment of Quality of Service (QoS) requirements, as well as an efficient use of allocated resources. The operation of the proposed solution can be configured from the user file, enabling the use of different strategies. We compare the performance of the proposed scheduler with that exhibited by alternative solutions. The results show that it can guarantee the required bit rate, while yielding a more efficient use of the available resources.

## CCS CONCEPTS

• **Networks** → *Network simulations*; Data path algorithms.

## KEYWORDS

ns-3, 5G-LENA, MAC scheduler, QoS, Lyapunov, max-weight, XR

**ACM Reference Format:**
Neco Villegas, Ana Larrañaga, Luis Diez, Katerina Koutlia, Sandra Lagén, Ramón Agüero. 2024. Extending QoS-aware Scheduling in ns-3 5G-LENA: A Lyapunov based Solution. In *2024 Workshop on ns-3 (WNS3 2024), June 05–06, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3659111.3659118

## 1 INTRODUCTION

The development of novel services and their support by mobile networks pose stringent requirements in terms of Quality of Service (QoS). In order to tackle this, 5th Generation (5G) networks are designed to bear a large variety of services, having different characteristics. Along with the increasing capacity demanded by enhanced Mobile Broadband (eMBB) based applications, massive Machine Type Communications (mMTC) and Ultra Reliable Low Latency Communications (URLLC) require a very large number of connected devices and extremely good communication performance. Among new services, it is worth mentioning those that fall under the umbrella of eXtended Reality (XR) [1, 4], which require both high data-rates and low latencies. XR refers to all services featuring the combination of real and virtual environments, and human-machine interaction, such as: Virtual Reality (VR), which provides users the perception of being physically at a given place; Augmented Reality (AR), which adds overlaid content to the physical environment; or Mixed Reality, where users can interact with virtual elements.

In all cases, 5G networks and beyond need to dynamically and efficiently adapt the underlying communication resources to leverage the aforementioned services. Compared to previous cellular technologies, 5G New Radio (NR) introduces an extended QoS provisioning framework, based on QoS flows, that imposes requirements at flow level [2], and which are then applied to services, for instance XR [3].

One of the key aspects to enforce appropriate QoS is the MAC scheduling, which manages how the radio-resources are shared among traffic flows, and whose behavior is not standardized. Therefore, the design of appropriate schedulers tailored to the new QoS management framework introduced by 5G NR is of utter importance. There exist many approaches that have been proposed to design such MAC schedulers. It is worth highlighting those rooted in Lyapunov control theory, and max-weight scheduling in particular, which have been extensively used, due to their stabilization properties. Lyapunov drift-plus-penalty was first used in [9] for optimal routing and scheduling in multi-hop networks and then in [10] for single hop communications. In both cases, the application of these techniques assumes random connectivity, as well as changing stability properties. Since then, these techniques have been applied to a multitude of areas and techniques to tackle time-varying stochastic problems.

In his seminal work [6] Neely proposed a general framework using Lyapunov techniques for queue stabilization. It includes max-weight (or "min-drift") approaches, as well as the general min-drift-plus-penalty control problem that does not only address queue stabilization, but it also allows including time-average objectives to the decision process.

In this paper, we describe the initial design and implementation of a generic scheduler for 5G supported QoS based on a drift-plus-penalty control problem. The scheduler design pursues multiple objectives, whose importance can be configured: traffic flow stability, guaranteed throughput and minimization of communication resources. The designed scheduler has been implemented and validated in the 5G-LENA network simulator [8] over scenarios using XR traffic.



*WNS3 2024, June 05–06, 2024, Barcelona, Spain*

ACM ISBN 979-8-4007-1763-5/24/06
https://doi.org/10.1145/3659111.3659118

The contributions of this paper are:

- We extend the ns-3 5G-LENA scheduling approach, by integrating a generic solution that can be used to implement Lyapunov-based policies.
- We show that the proposed scheme is able to guarantee a minimum data rate, while ensuring an efficient use of allocated resources.
- We show that the operation of the scheduling policy can be tuned based on a configuration parameter.
- We compare the performance of the proposed solution with alternative schedulers already available in 5G-LENA.
- The implementation has been made available in a public repository.

The rest of the paper is structured as follows. Section 2 discusses the main concepts related to 5G QoS management, and the solutions currently available in 5G-LENA, positioning the contribution discussed herewith. The scheduler design and its implementation and integration within 5G-LENA are described in Section 3. Section 4 validates the correct operation of the proposed scheduler, and compares its performance with alternative solutions. Finally, Section 5 concludes the paper and provides an outlook of our future work.

## 2 RELATED WORK AND BACKGROUND

End-to-end connectivity in 5G networks is managed by Packet Data Unit (PDU) sessions, which represent logical connections between a User Equipment (UE) and a data network. In turn, a PDU session supports one or more traffic flows (QoS flows), which represent the finest granularity of QoS. In the radio access network, QoS flows are mapped onto Data Radio Bearer (DRB), which are then assigned to Radio Link Control (RLC) entities. Each flow is given a QoS flow profile that defines how it should be treated, and it includes various parameters. A 5G QoS profile specifies, among others:

- **5G QoS Identifier (5QI).** It is a scalar that identifies a set of QoS parameters.
- **Allocation and Retention Priority (ARP).** It contains information used to handle the flow in cases of limited resources.
- **Flow bit rates.** The 5G QoS considers the definition of Maximum Flow Bit rate (MFBR) and Guaranteed Flow Bit Rate (GFBR) for each traffic flow, establishing the upper and lower bit rate thresholds.

In more detail, 5QI defines the following parameters:

- **QoS flow Type.** The 5G architecture categorizes the traffic flows into 3 general types:
  - **Non-GBR QoS flow.** It does not require a specific bit rate, and is typically used for non-time-sensitive applications.
  - **GBR QoS flow.** It specifies a bit rate threshold to the end user, so it is suitable for time-sensitive applications, such as voice and video calls, real-time gaming or V2X.
  - **Delay Critical (DC)-GBR QoS flow.** It does not only require a certain bit rate, but it also ensures lower latencies (i.e., $< 10\,ms$ delay budget) and error rates for mission-critical applications.
- **Priority Level.** In the case of congestion, it should be used to select which flow is prioritized.
- **Packet Delay Budget (PDB).** It establishes the maximum delay that a packet may suffer between the UE and the User Plane Function (UPF) which is connected to the data network [2].
- **Packet Error Rate (PER).** It defines the limit for the ratio of IP packet that have been processed by the transmitter, for instance at the RLC layer, but have not been successfully received by the corresponding upper layer at the receiver [2].
- **Maximum Data Burst Volume (MDBV).** It corresponds to the largest amount of data that the access network might serve within a PDB period.
- **Averaging window.** Time window for the computation of the bit rate thresholds defined below. It is set to 2 seconds according to the standardized 5QI values [2].

At the time being, the 5G-LENA network simulator implements a set of schedulers, some of them being generic, while others are tailored to the 5G QoS framework. The simulator can be configured to use the traditional Round Robin (RR) and Proportional Fairness (PF) schedulers, which aim to fairly distribute the resources between active flows. In addition, it provides a Maximum Rate (MR) scheduler, which seeks to maximize the accumulated throughput by leveraging information of the radio channel state, exploiting the Modulation and Coding Scheme (MCS) indicator. Furthermore, a novel QoS scheduler has been recently added [5], which explicitly uses the 5QI parameters to prioritize resource assignment.

As can be seen, Lyapunov based scheduling strategies, based on either max-weight or extended drift-plus-penalty, have not yet been integrated within the simulator. In this sense, this paper aims to provide a first version of a generic Lyapunov based scheduler adapted to 5G QoS. In particular, the proposed scheduler design considers traffic queue stability, GFBR, as well as an efficient use of radio resources.

## 3 SCHEDULER DESIGN AND IMPLEMENTATION

This section presents the scheduler design, identifying its relationship with the 5G specific concepts that were defined in Section 2. We also discuss the assumptions that were taken, as well as the parameters that can be tuned. Afterwards, we describe the implementation and integration of the scheduler within the 5G-LENA framework.

### 3.1 Scheduler Design

The scheduler design considers a number of traffic queues, each having a 5QI and potentially requiring a GFBR. Let $\mathcal{N}$ be the set of queues, A the amount of available resources (i.e., Physical Resource Blocks), and $\Gamma_n$ the GFBR value of each queue. The proposed design aims to keep the traffic queues stable, while ensuring that the GFBR is respected, and at the same time to minimize the amount of used resources. For simplicity, we will assume that each queue corresponds to one user, although a more general approach, mapping queues to logical channels, could have been also assumed. The latter will be discussed below, in Section 3.2.

We consider a slotted time, so that $Q_n(t)$ indicates the backlog of queue $n \in \mathcal{N}$ at time slot $t$, and $\alpha_n(t)$ holds for the scheduling decision for each queue. As a consequence of the scheduling, as well as other random parameters, for instance the Signal to Interference and Noise Ratio (SINR), $\omega(t)$, $b_n(t)$ traffic units leave the $n^{th}$ queue at slot $t$, while $a_n(t)$ enter it. According to the incoming and outgoing traffic, queues in the system are updated as follows:

$$Q_n(t+1) = \max\{Q_n(t) - b_n(t), 0\} + a_n(t) \tag{1}$$

In addition, we define the instantaneous throughput perceived by the $n^{th}$ user at time slot $t$ as $\rho_n(t)$. Altogether, we can express the incoming and outgoing traffic per queue, as well as the throughput perceived by the corresponding user, as generic functions of the scheduling decisions and the random events, as follows:

$$a_n(t) = \hat{a}_n(\alpha_n(t), \omega(t)) = \hat{a}_n(\omega(t)) \tag{2}$$

$$b_n(t) = \hat{b}_n(\alpha_n(t), \omega(t)) \tag{3}$$

$$\rho_n(t) = \hat{\rho_n}(\alpha_n(t), \omega_n(t)) \tag{4}$$

As mentioned before, the GFBR is calculated using an averaging window of 2 seconds, while the scheduling decisions are taken at a much shorter time units, in the range of milliseconds. Thus, we aim to ensure that the throughput requirement is fulfilled, in average, over time, and we therefore consider its time-average expectation $\overline{\rho_n} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{\rho(t)\}$. As mentioned before, we also aim to minimize the amount of allocated resources, being $f(t) = \sum_{n\in\mathcal{N}} \alpha_n(t)$ the resources that were allocated at slot $t$. Since the greedy minimization of resources may hinder the throughput, we define as an objective the average minimization of allocated resources, $\overline{f} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{f(t)\}$. All in all, the scheduler solves Problem 1, where Eqs. (6) and (7) ensure that the throughput requirement is fulfilled, and that the amount of assigned resources do not exceed the available ones. On the other hand, as was said earlier, the scheduler also aims to keep traffic queues $Q_n(t)\ \forall n \in \mathcal{N}$ are mean rate stable, $\lim_{t\to\infty} \frac{\mathbb{E}\{|Q_n(t)|\}}{t} = 0$.

Problem 1.

$$\min_{\alpha(t)} \quad \overline{f} \tag{5}$$

$$\textbf{\textit{s.t.}} \quad \overline{\rho_n} \geq \Gamma_n \quad \forall n \in \mathcal{N} \tag{6}$$

$$\sum_{n\in\mathcal{N}} \alpha_n(t) \leq \Lambda \quad \forall t \tag{7}$$

$$\textit{Queues } Q_n(t) \textit{ are mean rate stable } \forall n \in \mathcal{N} \tag{8}$$

In order to tackle the above stochastic optimization problem, we adopt the framework defined in [6], which generalizes the max-weight algorithm. First, we transform the constraint (6) into a virtual queue by defining $D_n(t) = \Gamma_n - \rho_n(t)$, so that $\overline{D_n} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{\Gamma_n - \rho_n(t)\} = \Gamma_n - \overline{\rho_n}$. Then, we define a virtual queue with the following update:

$$G_n(t+1) = \max\{G_n(t) + D_n(t), 0\} \tag{9}$$

Now the problem can be solved by applying the *min drift-plus-penalty* algorithm [6, Chapter 4], which boils down to solving, in every slot, the following problem:

Problem 2.

$$\min_{\alpha(t)} \quad V \cdot \sum_{n\in\mathcal{N}} \alpha_n(t) + \sum_{n\in\mathcal{N}} Q_n(t)[a_n(t) - b_n(t)] + \sum_{n\in\mathcal{N}} G_n(t) D_n(t) \tag{10}$$

$$\textbf{\textit{s.t.}} \quad \sum_{n\in\mathcal{N}} \alpha_n(t) \leq \Lambda \quad \forall t \tag{11}$$

$$b_n(t) \leq Q_n(t) \quad \forall n \in \{1, \dots, N\}, \forall t \tag{12}$$

where the first term of the objective function corresponds to the minimization of allocated resources. Then, the second and third term are related to the stabilization of the original traffic queues and the virtual queues that address the throughput requirements. As can be observed, the first term is multiplied by a design parameter $V$ which permits giving more relevance to the objective function compared to the queues' backlog. In each slot the scheduler follows the following steps:

(1) Observe the states of $Q_n(t), G_n(t)\ \forall n \in \mathcal{N}$
(2) Take the decision that minimizes Problem 2
(3) Update queues as defined in Eq. (1) and (9)

### 3.2 Implementation within 5G-LENA

The proposed scheduler has been implemented following the design approach already used in the NR Module [7]. The core class of the MAC scheduler is `NrMacSchedulerNs3`. Both Time Division Multiple Access (TDMA) and Orthogonal Frequency Division Multiple Access (OFDMA) schemes are the subclasses responsible for allocating radio resources as well as generating the corresponding Downlink Control Information (DCI). A radio resource is composed by an Orthogonal Frequency-Division Multiplexing (OFDM) symbol in time domain and a Resource Block Group (RBG) in frequency domain. The implemented TDMA in 5G-LENA assigns all RBGs of an OFDM symbol to the same packet, while the implemented OFDMA in 5G-LENA assigns an RBG in every OFDM symbol within a slot to the same packet. Our implementation will consider only the OFDMA scheme, but the implementation discussed hereinafter would straightforwardly apply to TDMA scheme as well. References to TDMA appear through inheritance relationships with OFDMA classes. The schedulers that use OFDMA multiple access scheme are subclasses of the `NrMacSchedulerOfdma` class (e.g., `NrMacSchedulerOfdmaRR`, `NrMacSchedulerOfdmaQos`, etc.). The type of scheduling and the multiple access scheme can be specified from the user example. These classes can implement the virtual functions that update the metrics for each UE, the same way the QoS scheduler does [5]. These metrics are then available if needed for the class representing the UE (e.g., `NrMacSchedulerUeInfoRR`, etc.), which sorts active UEs (i.e., those with data to transmit) according to the scheduling policy, to allocate resources based on the implemented strategy.

Figure 1 shows the Unified Modeling Language (UML) diagram of the classes of interest. For clarity, only the classes corresponding to RR are shown, as the layout is the same for the rest of the supported schedulers. The functions `AssignDLRBG()` and `CreateDlDci()` defined in `NrMacSchedulerOfdma` class are responsible for assigning the requested radio resources and creating the DCI messages, respectively. The proposed scheduler has been implemented as a new specialization in the aforementioned framework, for which two new classes have been created: `NrMacSchedulerOfdmaDPP` and `NrMacSchedulerUeInfoDPP`. These are highlighted in green color in the figure.

Our scheduler approach to resource allocation differs from the sorting mechanism defined in `AssignDLRBG()` which is followed by the schedulers currently available in 5G-LENA. While the sorting function is called iteratively during a slot until all demanded or available radio resources are allocated, our scheduler implementation makes the decision in one step. This is done by solving the underlying Integer Linear Programming (ILP) problem described in Section 3.1. For this purpose, the corresponding problem instances are solved using the optimization GNU Linear Programming Kit (GLPK)[1]. After allocating the demanded radio resources, OFDMA scheme calls `AssignedDlResources()` in order to update DL information with the allocated radio resources. Then, `NrMacSchedulerOfdma` class will create the DCI messages considering this information.

In the new implementation, it is necessary to override the function `AssignDLRBG()` that informs about the allocated resources. This virtual function is implemented by all scheduler classes, and it establishes a loop that calls the sorting function along with a comparison function defined in the active scheduler. Therefore, in the new `NrMacSchedulerOfdmaDPP()` class, the `AssignDLRBG()` function calls `LyapunovDPP()`, where the GLPK is used in order to allocate radio resources for every packet. As a result, `LyapunovDPP()` returns the resource allocation solution. Then, `AssignedDlResources()` is called to update DL information accordingly with the new allocated radio resources information.

The inputs required by the problem are:

- The occupancy of the RLC queues.
- The state of the virtual throughput queue, defined as a member (`m_g`) of the new `NrMacSchedulerUeInfoDPP` class, and an update every slot through `UpdateDlTputVirtualQueue()`.
- The state of the channel, which we achieve with an access to the TBS calculation.
- The total number of available resources that can be scheduled.
- The $V$ configuration parameter, which is defined as a member (`m_v_lyapunov`) of the new `NrMacSchedulerOfdmaDPP` class, allowing to be configured from the user example.

As previously mentioned, the new scheduler allows some configuration, such as the parameter $V$, to be made from the user example. Additionally, the virtual queue can be enabled or disabled, using the boolean `m_enable_virtual_queue` member, included in `NrMacSchedulerOfdmaDPP`. As a result, the scheduler acquires a great deal of flexibility, allowing it to reflect several policies.

Finally, although the scheduler has been designed to comply with the GFBR that is defined by flows, the scheduling is carried out by UEs. Therefore, it performs optimally when assuming a single flow per UE, but this does not always need to be the case. For these scenarios in which UEs generate multiple flows with different GFBRs, it would be sensible to calculate the average GFBR to create the virtual queue. This will not be needed when scheduling radio resources at the Logical Channel (LC) level becomes possible, in future releases of 5G-LENA.

[1] https://www.gnu.org/software/glpk/

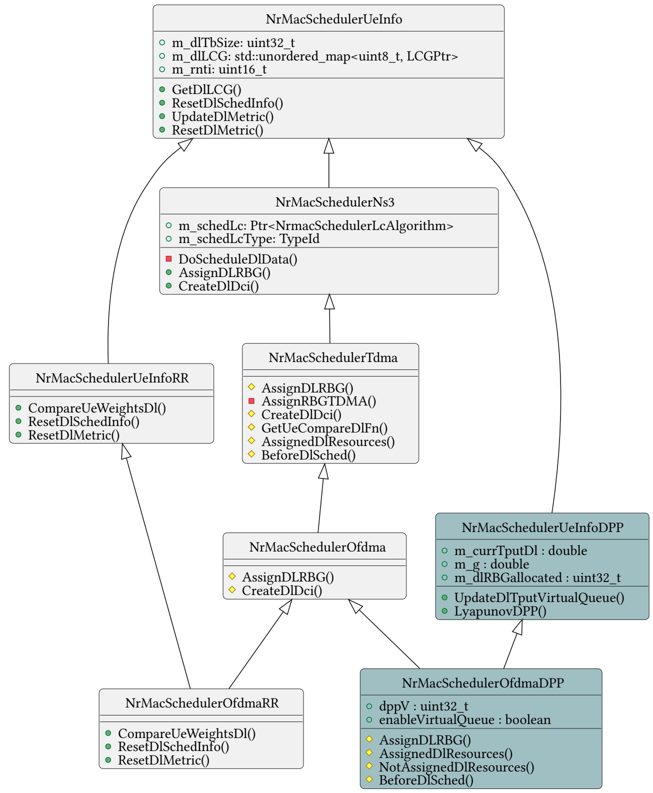


**Figure 1: UML Diagram of the Current Classes Implemented in 5G-LENA (Gray) and the New Proposed Classes (Green)**

## 4 EVALUATION

This section discusses the performance analysis of our proposal. The primary objective of the evaluation is to assess the feasibility of the GFBR aware scheduler, and to compare its performance with that yielded by existing schedulers already implemented in 5G-LENA, such as RR, MR, PF, and QoS. Let us notice that throughout this section, we will refer to the proposed scheduler as Drift Plus Penalty (DPP).

The proposed scheduler is evaluated using a simple topology that includes a gNB and a number of UEs. The scenario configuration is detailed in Table 1. The traffic is generated using the `XrTrafficMixerHelper`, which provides various 3GPP XR traffic flows, and the particular configuration used in the evaluation is shown in Table 2. As can be seen, we do not impose any bit rate guarantee for VR traffic, while the Cloud Gaming (CG) service requires 4 $Mbps$. We set this configuration to better compare the effect of the configuration parameters of the scheduler.

Figure 2 compares the performance of the GFBR aware scheduler (configured with $V = 0$) with that obtained with various existing MAC schedulers. It shows the end-to-end average throughput for the setup shown in Table 1. The first 4 groups of bars show the result for 4 UEs, each of them having one VR traffic flow, while the rightwards bar groups correspond to other 4 UEs, who are using CG traffic (one flow each). Then, we use different colors to represent

**Table 1: Configuration of the Evaluation Setup**

| Parameter | Value |
|---|---|
| Application duration | 10 *sec.* |
| Number of UEs per gNB | 8 |
| Numerology | 0 |
| gNB Tx power | 41 *dBm* |
| Carrier frequency | 4 *GHz* |
| Bandwidth | 10 *MHz* |
| Channel condition | UMa with all the nodes in LoS. |

**Table 2: Configuration of the XR Traffic**

| Virtual Reality traffic | | Cloud Gaming traffic | |
|---|---|---|---|
| Number of flows | 1 | Number of flows | 1 |
| 5QI value | 80 | 5QI value | 3 |
| Tx offered | 5 *Mbps* | Tx offered | 5 *Mbps* |
| Guaranteed Bit Rate | – | Guaranteed Bit Rate | 4 *Mbps* |

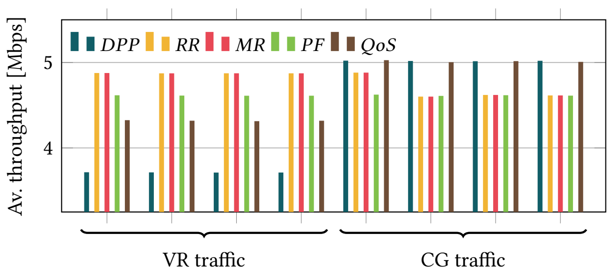


**Figure 2: End-to-end Average Throughput Comparison Between Schedulers**

the results obtained with each of the different schedulers: DPP, RR, MR, PF and QoS.

As can be seen, there is not any difference on the throughput observed for most of the existing schedulers (RR, MR and PF) for the two traffic types. On the other hand, both the QoS scheduler and DPP yield a different throughput for each traffic type, but for distinct reasons. The first one takes scheduling decisions according to the value of 5QI assigned to each user: QoS requirements for *Real Time Gaming* (5QI=3) are more stringent than *Low Latency eMBB applications Augmented Reality* (5QI=80), resulting in higher resource allocation and increased throughput for the CG traffic. On the other hand, the proposed scheduler achieves similar results, but in this case following its main purpose, which is to maintain a throughput that satisfies the corresponding GFBR for each flow, being this requirement higher for the CG users (4 *Mbps*).

Figures 3 and 4 illustrate the impact of the $V$ configuration parameter of the proposed scheduler, which trades off between the throughput requirement (stabilization of the corresponding virtual queue) and the usage of resources (objective function). We carried out a number of experiments, increasing $V$ from 0 (where the scheduling policies would not consider the amount of allocated resources at all) to $1e10$ (prioritizing the minimization of allocated resources over queue stabilization). For each value of the parameter we obtain the decisions from 10 independent experiments of 10 seconds, which conveys around $1e5$ allocation decisions for each user.

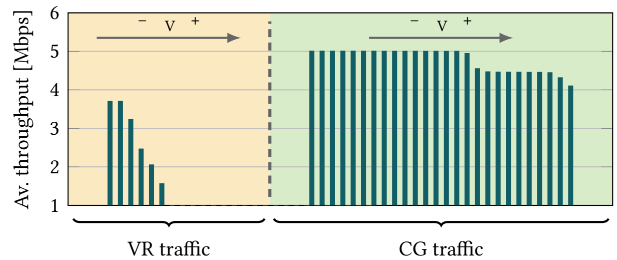


**Figure 3: End-to-end Average Throughput Achieved by the GFBR Aware MAC Scheduler as a Function of $V$**

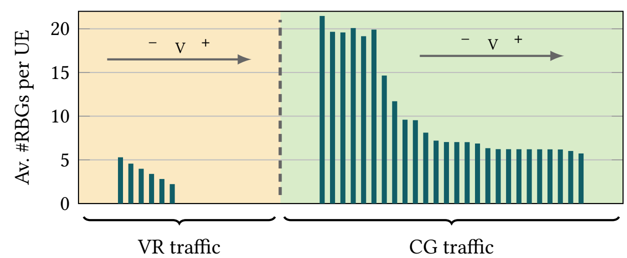


**Figure 4: Variation of the Average Resources Allocated per UE, for the GFBR Aware MAC Scheduler, when Sweeping $V$**

Figure 3 shows the average end-to-end throughput, while Figure 4 represents the average amount of resources used by each UE. In both figures, we group the results according to the type of UEs (and so their XR traffic type). There are still 4 UEs of each type, but in this case, we represent the average value of the performance indicator to yield a more clear representation of the results. In addition, the value of $V$ is increased for each bar from left to right, starting from the leftwards bar, which corresponds to $V = 0$.

As can be seen in Figure 3, as we increase $V$ the throughput of the VR traffic decreases until it becomes zero (the penalty is more relevant than the traffic queue). On the other hand, the throughput of the CG traffic always remains above the threshold, even with high values of $V$. Accordingly, Figure 4 shows that the resources granted stands above a minimum value (around 5 resources), which guarantees the GFBR configured, no matter the value of $V$ is. From a more general perspective, for the analyzed scenario and configuration the scheduler tends to prioritize the CG traffic even if its throughput is well above the threshold. Nevertheless, this configuration could be modified by using weights for the different queues, both traffic and virtual queues.

Altogether, these results validate the behavior of the proposed MAC scheduler and its implementation in 5G-LENA. They show that the scheduler is aware of the GFBR requirements of the QoS flows and, at the same time, is able to balance the amount of allocated resources and the stabilization of the RLC queues (and so performance). Additionally, it can be configured to tune the scheduling policies by using the $V$ configuration parameter.

## 5 CONCLUSION

In this work, we discuss the design and implementation of a novel multi-objective MAC scheduler, tailored for 5G QoS-aware networks in the ns-3 simulator. We first describe the design of the solution, which is based on a drift-plus-penalty control approach. In a nutshell, the scheduler aims to stabilize traffic flows, ensuring the required throughput for each flow based on their QoS requirements, while minimizing the radio resources allocated by base stations.

We implemented the new scheduler in the 5G-LENA ns-3 module, extending the scheduling solutions that are already available. Then, in order to assess the effectiveness of the proposed scheduler, we performed system-level end-to-end simulations with multiple UEs using QoS demanding flows, corresponding to XR based services. The results show that the proposed scheduler provides the required throughput, according to the QoS characteristics of each flow. In particular, we have shown that the scheduler can indeed prioritize GBR flows, while guaranteeing, at the same time, the stabilization of other flows. On top of that, the scheduler seeks the minimization of allocated radio resources, according to a configuration parameter. We plan to exploit the implementation that has been introduced to facilitate further research. Concretely, in our upcoming work, we will expand the model in various ways. First, we will analyze the performance of the optimization problem when adding more complex functions, such as logarithmic functions, to foster different trade-offs between resource consumption and queues stabilization. Additionally, we pretend to design, implement and analyze in ns-3 delay-based back pressure schedulers, particularly designed to foster DC-GBR, including in this way the QoS requirements relating to the delay to the scheduling framework presented herein.

## ARTIFACTS

The source code of our work is available at https://github.com/tlmat-unican/5glena-lyapunov-mac-scheduler. The repository contains ns-3 version 3.39 and 5G-LENA version 2.5, including the GFBR aware scheduler, as well as scenarios to replicate the results discussed in this paper. A Jupyter notebook is provided to automate experiment execution.

## ACKNOWLEDGMENTS

This work has been supported by the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union – NextGenerationEU with the project SITED: Semantically-enabled Interoperable Trustworthy Enriched Data-spaces (PID2021-125725OB-I00), OPTIMAIX_OaaS (TSI-063000-2021-34) and 6GBLUR/joint (TSI-063000-2021-57). We have also received funds from the regional Cantabria Government, Fondo Europeo de Desarrollo Regional (FEDER), with the TCNIC program (2023/TCN/002).

## REFERENCES

[1] 3rd Generation Partnership Project (3GPP). 2022. Study on XR (Extended Reality) Evaluations for NR. 38.838 (2022).

[2] 3rd Generation Partnership Project (3GPP). 2023. System Architecture for the 5G System (5GS). 23.501 (2023).

[3] Biljana Bojović, Sandra Lagén, Katerina Koutlia, Xiaodi Zhang, Ping Wang, and Liwen Yu. 2023. Enhancing 5G QoS Management for XR Traffic Through XR Loopback Mechanism. *IEEE Journal on Selected Areas in Communications* 41, 6 (2023), 1772–1786. https://doi.org/10.1109/JSAC.2023.3273701

[4] Margarita Gapeyenko, Vitaly Petrov, Stefano Paris, Andrea Marcano, and Klaus I. Pedersen. 2023. Standardization of Extended Reality (XR) over 5G and 5G-Advanced 3GPP New Radio. *IEEE Network* 37, 4 (2023), 22–28. https://doi.org/10.1109/MNET.003.2300062

[5] Katerina Koutlia, Sandra Lagén, and Biljana Bojović. 2023. Enabling QoS Provisioning Support for Delay-Critical Traffic and Multi-Flow Handling in ns-3 5G-LENA. In *Proceedings of the 2023 Workshop on ns-3* (Arlington, VA, USA,) *(WNS3 '23)*. Association for Computing Machinery, New York, NY, USA, 45–51. https://doi.org/10.1145/3592149.3592159

[6] Michael J. Neely. 2010. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. http://dx.doi.org/10.2200/S00271ED1V01Y201006CNT007

[7] OpenSim CTTC/CERCA. 2024. NR Module Manual. https://cttc-lena.gitlab.io/nr/nrmodule.pdf

[8] Natale Patriciello, Sandra Lagén, Biljana Bojović, and Lorenza Giupponi. 2019. An E2E Simulator for 5G NR Networks. *Simulation Modelling Practice and Theory* 96 (2019), 101933. https://doi.org/10.1016/j.simpat.2019.101933

[9] Leandro Tassiulas and Anthony Ephremides. 1992. Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks. *IEEE Trans. Automat. Control* 37, 12 (1992), 1936–1948. https://doi.org/10.1109/9.182479

[10] Leandro Tassiulas and Anthony Ephremides. 1993. Dynamic Server Allocation to Parallel Queues with Randomly Varying Connectivity. *IEEE Transactions on Information Theory* 39, 2 (1993), 466–478. https://doi.org/10.1109/18.212277