Development of New Radio Schedulers in OpenAirInterface

Pablo Andrés Vázquez
Universidad de la República
Montevideo, Uruguay
pablo.vazquez@fing.edu.uy

Walter Piastri
Universidad de la República
Montevideo, Uruguay
walter.piastri@fing.edu.uy

Wilder Pena
Universidad de la República
Montevideo, Uruguay
wilder.pena@fing.edu.uy

Lucas Ingles
Universidad de la República
Montevideo, Uruguay
lucasi@fing.edu.uy

Claudina Rattaro
Universidad de la República
Montevideo, Uruguay
crattaro@fing.edu.uy

ABSTRACT

This paper presents an analysis, implementation, and validation of new radio schedulers within the OpenAirInterface (OAI) framework. Leveraging the flexibility of OAI, we scrutinize existing MAC layer functionalities and propose enhancements through the integration of new scheduling algorithms. Our study involves a meticulous evaluation of the new scheduling techniques, considering factors such as throughput and fairness. The implementation process is thoroughly detailed, highlighting the integration of these schedulers into the OAI ecosystem. The validation of these new scheduling techniques is conducted using a 5G Standalone network testbed, incorporating real radio interfaces facilitated by software-defined radio technology. Results demonstrate significant improvements in network performance, particularly in terms of throughput and fairness. Our implementation also supports dynamic control of scheduler behavior during runtime, offering a robust platform for future innovation. This work lays the groundwork for future research in advanced scheduling algorithms for 5G and beyond, providing a robust platform for further innovation.

CCS CONCEPTS

• **Networks → Network experimentation**; **Network performance analysis**; **Network algorithms**; *Wireless access points, base stations and infrastructure*; Network protocols.

KEYWORDS

Radio Scheduler, OpenAirInterface, 5G Mobile Networks

1 INTRODUCTION

The advent of the fifth generation (5G) of mobile networks has underscored the critical importance of advanced scheduling techniques in efficiently managing the limited frequency spectrum while satisfying the transmission demands of 5G technology. There is extensive research proposing new scheduling techniques for 5G networks and future 6G networks, see for example the survey [2].

Innovative wireless communication systems can be crafted leveraging the open-source mobile network technology called OpenAirInterface (OAI) [4]. From handling the intricacies of the physical layer to orchestrating the network layer, it provides a dynamic foundation for constructing comprehensive wireless setups. It implements the full 5G protocol stack for user equipment (UE), gNodeB (gNB), and the core network (CN), and is capable of operating on general-purpose x86 processors.

By harnessing the Linux kernel and Linux IP protocol stack in conjunction with off-the-shelf software-defined radios (SDRs), they furnish a comprehensive mobile network that adheres to 3GPP standards. Moreover, its modular nature and well-defined interfaces between components streamline the integration of new functionalities and features. OAI extends its support to simulation, emulation, as well as real-time experiments.

This work utilizes the OAI emulation platform, a C developed system that runs on Ubuntu Linux, over Commercial Off-The-Shelf (COTS) hardware to construct a wireless network scenario and develop new resource allocation algorithms, laying the groundwork for future research in this field. In particular, our work scenario is within the OAI5GRANProject [6], which has only one resource allocation algorithm available in its current version. As a final product of our work, we provide the community with a repository containing the new development of schedulers, which also includes the possibility of external configurations (for selecting the algorithm) without the need to modify the OAI base code [9].

It is essential to highlight several closely related antecedents pertinent to our work. In [1], the performance of 5G mobile networks is analyzed by comparing the results with existing 5G scheduler algorithms available in OAI, specifically in terms of throughput. However, this study does not implement any new algorithms or variants of existing ones. While the authors of [7] develop new schedulers within the MOSAIC5G project [3], our work differs as we focus on the base OAI RAN project, OAI5GRANProject. Another significant difference from previous works is that we will verify the performance of the new algorithms using SDR equipment with an RF interface.

This paper is structured as follows. In Section 2, we provide a brief introduction to the 5G scheduling system. Section 3 presents our 5G Standalone network testbed, which is based on the OAI development framework integrated with real radio interfaces facilitated by SDR technology. Additionally, we introduce the OAI implementation of their default radio scheduler. In Section 4, we describe our design and implementation of new schedulers. The validation of these schedulers is detailed in Section 5. Finally, in Section 6, we conclude the work and discuss potential future research directions.

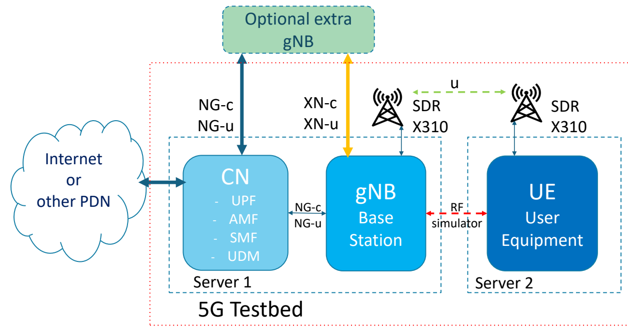

Figure 1: Schematic Diagram of the Testbed: Illustrating Component Blocks and Interfaces

2 INTRODUCTION TO 5G SCHEDULING

Resource allocation involves distributing network resources effectively to enhance system performance, ensure fairness, and meet Quality of Service (QoS) standards for various applications and users. The scheduler is a high-level process responsible for distributing available bandwidth among users. It allocates frequency ranges and time periods to each user, managing this process at the MAC layer while interacting with the physical layer (PHY). This interaction determines the priority order of bandwidth allocation. In the RAN (Radio Access Network), the physical resource block (PRB) serves as the fundamental unit for this allocation. In the realm of 5G, which employs the OFDM (Orthogonal Frequency Division Multiplexing) modulation system, a PRB comprises 12 consecutive OFDM subcarriers in the frequency domain and a transmission time interval (TTI) in the time domain.

For downlink (DL) at the MAC layer, the transport block (TB) is assembled and delivered to the PHY layer, multiplexing all user and control data streams directed to all UEs in the cell. Resources are allocated for all control processes, both mandatory and optional, and then the remaining resources are allocated, multiplexing data from different data streams according to the established priority. The PHY layer is instructed on how to configure itself to transmit this TB, and control messages (Control Elements, CE) must be sent to the UEs with information on where, when, and how the corresponding information is transmitted to them. This information includes modulation, coding, allocated PRBs, and slots. A similar process is performed in the uplink (UL) direction. The UE is configured, and the PHY layer is informed about where and how it receives the information.

In addition to assigning frequency and time, and accounting for these resources based on modulation for each UE and the number of layers it is receiving and transmitting, it is also necessary to coordinate resource allocation across all Component Carriers (CC) being used by each UE, and furthermore, allocate simultaneous resources when massive MIMO is involved. When different services are differentiated into different slices, the problem of prioritizing resources between slices and scheduling them among the UEs of each slice overlaps.

3 OAI 5G NETWORK TESTBED FOR TESTING AND DEVELOPMENT

Our testbed consists of three software applications: a Core Network (CN), a base station (gNB, in 5G terminology), and a User Equipment (UE). The CN handles tasks such as user registration and authentication, IP assignment, tunneling for internet access, managing mobility, enforcing security measures, implementing policies, handling accounting, and overseeing network management. The gNB defines the radio access interface, establishes user connections, dynamically allocates physical resources, and maintains control over the radio link to ensure reliable navigation for all connected users. Given its pivotal role, the gNB configurations are the most critical decisions in each test scenario. Meanwhile, the UE serves as the user-side counterpart, operating under the control of the gNB.

In our deployment, both the CN and the gNB are installed on the same physical machine, while the UE operates on a separate computer. Both computers are connected to an SDR USRP X310 via a 10 Gb Ethernet interface (we have also tested the solution with USRP B210, a lower end SDR device). This setup allows the second computer to establish connections to external networks through CN gateways, utilizing its services and control, and using the radio interface between the gNB and the UE. The UE provides a virtual network interface to the host machine. This architectural configuration is depicted in Figure 1. All three systems are implemented using the OAI 5G Core Network project and the 5G RAN project's software. All the details of the testbed and its configuration can be found in our previous article [8]. Figure 2 shows the testbed in operation. On the left side is the server and SDR implementing the CN and gNB, and on the right is the server and SDR implementing the UE. It is possible to connect additional UEs, both commercial ones and those composed by server and SDR.

3.1 OAI's Default Scheduler

In the implementation of OAI, the scheduler's role is split into two main phases: preprocessing and post-processing. During preprocessing, the priority order is established, while post-processing involves assembling the transmission slot with the assigned UE's data and relevant control messaging. This iterative process entails

Figure 2: OpenAirInterface 5G Network Testbed for Testing and Development

reviewing all UEs connected to the gNB to determine which ones will be scheduled and with what priority.

UEs with failure indications, such as exceeding the maximum number of retransmissions without acknowledgment, or those without queued data, are excluded from the scheduling. Additionally, UEs with pending retransmissions are not allocated new resources; instead, they are assigned the same resources as before for the pending retransmission before any leftover resources are allocated.

In the original OAI scheduler UEs are ordered following a proportional fair algorithm. This is done by assigning priority directly proportional to the bitrate that each UE can reach in that subframe, and inversely proportional to the data transferred previously. In this way, the optimal use of the channel is prioritized, but at the same time, the priority is lowered to those who have used it most recently.

In practice, the throughput for each connected UE over the last 1000 ms window is calculated as follows:

$$\text{UE} \rightarrow \text{dl\_thr\_ue} = (1 - a) \times \text{UE} \rightarrow \text{dl\_thr\_ue} + a \times b$$

where a is a fraction of time (specifically, $a = 0.0005$), and b represents the number of bytes sent to that UE in the last slot.

Next, the size in available bytes of a Transport Block (TB) type for that UE is determined. A fixed base of one PRB is utilized for 20 slots to provide a comparison independent of future assignments. The number of carriers, layers, modulation index Q, and coding rate R used by this UE at that time govern the size of the base TB for that UE in that slot. Essentially, this represents the maximum number of bytes that can be transmitted at this time within a PRB, averaged over the next 20 slots under current conditions. This averaging is done to mitigate interference from control symbols, DMRS, UL, and hybrid slots in the decision-making process. Subsequently, the priority coefficient coef_ue is computed as:

$$\text{coef\_ue} = \frac{\text{TBS}}{\text{UE} \rightarrow \text{dl\_thr\_ue}}$$

where TBS denotes the Transport Block Size. This coefficient is stored in a data array, wherein each element contains a pointer to the data structure of each UE eligible for scheduling, along with the calculated coefficient. The list is sorted in descending order of coefficient magnitude using a `qsort` function with a comparison function that compares the coefficients.

In the post-processing stage, the transport channels are populated by adding the control data and transferring the user data to the Layer 1 interface. Initially, relays are placed, followed by populating user data, transferring all possible data from the Radio Link Control (RLC) layer buffer corresponding to the first UE in the list. If its data transmission is completed, the process continues with the next UE. This process is repeated in subsequent slots, recalculating the list with the new priority order.

4 DEVELOPMENT AND INTEGRATION OF NEW SCHEDULERS

In this study, we developed variants of the original proportional fair scheduler to implement other classic algorithms, such as max CQI (or max TBS). The developments are carried out within the pf_dl function. Specifically, where the coefficient is calculated, it is replaced by a new coefficient that allows for greater flexibility in the schedulers.

With the modification depicted in Figure 3, setting the exponent of the numerator to 1 and the denominator to 0 results in max TBS (analogous to a MAX CQI scheduler). This simple modification works well when two UEs have different MCS. However, if both have the same MCS, then the allocation decision is based on the order in the UE list under equal conditions in the comparison. The UE at the top of the list receives the majority of resources. In this case, it would be expected that both UEs receive equivalent allocations. To achieve this, a randomly generated integer was added to the struct containing the UE's data and the calculated coefficient, and the comparison function given to the qsort algorithm was modified. With the new comparison shown in Figure 4, the call to the qsort function sorts the list by coefficient first and in case of a tie it is sorted by the random value. This random value is recalculated once per iteration for each UE that is entered in the list to be scheduled. With this, an equivalent distribution of resources is obtained, when there are equal conditions.

In conclusion, this scheme yields three distinct scenarios. With both coefficients set to one, the original Proportional Fair behavior is maintained. In cases where the coefficients are equal, a randomization factor is introduced, although it has negligible impact since the equality of coefficients at the top of the list is not repeated in this algorithm. When the coefficient $\alpha = 1$ and $\beta = 0$, the functioning of MaxTBS is attained. If both coefficients are zero, the distribution is random, exhibiting an average behavior similar to Round Robin.

4.1 OAI Configuration

The gNB is configured using a text file formatted in ASN.1 provided by the libconfig library, which is used for parsing. It consists of a list of parameters with their values. Some parameters are themselves lists of parameters, and thus the file takes on a hierarchical structure with values.

Configuration parameters are stored in structs associated with each group in the form of 'paramdef_t', which is defined in files such as 'MACRLC_nr_paramdef.h' that we have used here. In this struct, variables are defined, associated with the texts naming the parameters in the files, the type of the expected value is defined, along with its range, default value, and some possible flags, such as whether it is a mandatory field, etc. By defining a struct of this type, the 'get_config' and 'ge_configlist' methods can be used to read the values from the configuration file and store them in the created struct. Then, these data must be processed, taking the actions corresponding to each configuration.

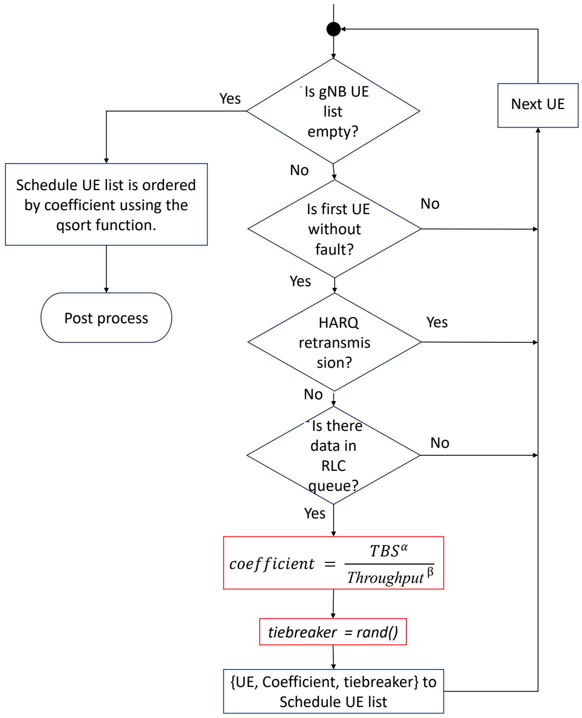

Figure 3: Flowchart of Prioritization of UEs Connected to a gNB in the Modified Scheduler Preprocess. The Red Boxes Are the Ones Modified by Us

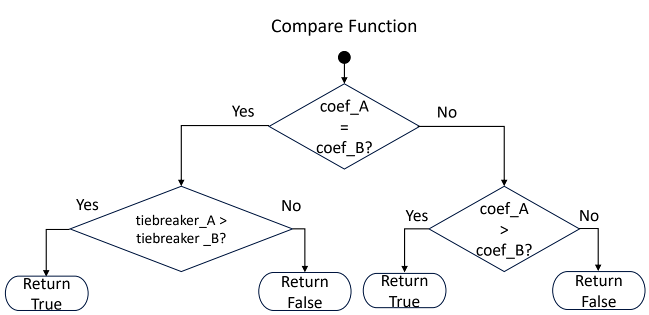

Figure 4: Comparison Function for the Ordering of the List of UEs to be Scheduled

To add the exponents that will manipulate the scheduler, four global variables were created and added to the format of the 'RC.nrmac' struct, which manages the variables of the MAC and RLC context. These variables were added to the 'MACRLC_nr_paramdef_t' struct type used to read the configuration at the start of the program, and copied into the created global variables. In each scheduler iteration, the exponents to be used are read from the global variable. This allows future work to be modified at runtime.

In the version created, the parameters 'ul_alpha', 'ul_beta', 'dl_alpha', and 'dl_beta' can be defined with integer values between 0 and 4 in the 'MACRLCs' section of the configuration file. The limit of 4 was chosen to continue operating with integers. If not included, the default value is 1 and the system will operate with the original Proportional Fair algorithm.

5 VALIDATION

The performance of the implemented scheduler variants was validated using the testbed defined in Section 3. Tests were conducted using both the simulated radio interface through the RF simulator[1] and the over-the-air interface. In this article, we chose to present the tests with the real radio interface. Other tests can be found in [5].

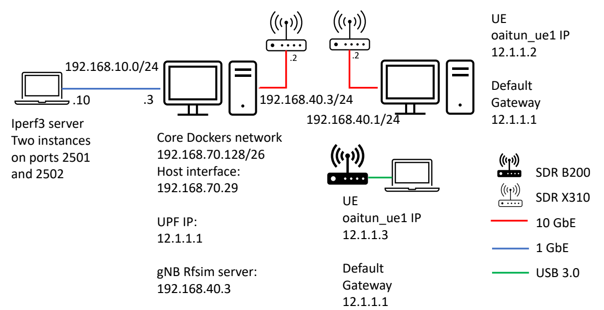

Figure 5: Testbed Setup for Scheduler Tests with Three SDRs

To test with a radio interface, three SDRs or one SDR and two phones are required. Since several Ettus B200 SDRs are available in our lab, which have a lower maximum bandwidth than the Ettus X310s, a TDD scenario is configured with 24 PRBs and a 30 kHz SCS, using the n41 band. This configuration provides a 10 MHz bandwidth for this test, allowing the use of both devices. Under these conditions, an X310 is used as the gNB, another as a UE, and a B200 with a laptop for the connection of the second UE.

The network diagram for testing is shown in Figure 5. An iperf3 server is used located in an external Data Network (DN) to generate downlink (DL) traffic. The iperf clients are located in UE machines, and connect to server through the hole 5G network. A throughput test is done in each UE to adjust the iperf datarate just over the maximal capacity in each radio condition. UDP is used to avoid congestion control by TCP. Then two test run simultaneously but not synchronized. All the assignment of each 0.5 ms slot are logged to a file, saving the first PRB, the amount of PRBs, the number of symbols in time domain, the MCS, and the total bytes assigned to

[1]RF simulator is an OAI feature that helps to test the OAI without any RF board since it imitates one.

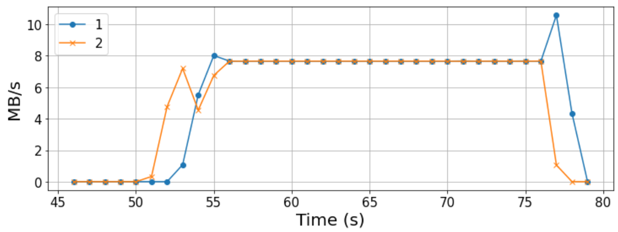

Figure 6: Allocation of Bytes in Two UEs with Proportional Fair

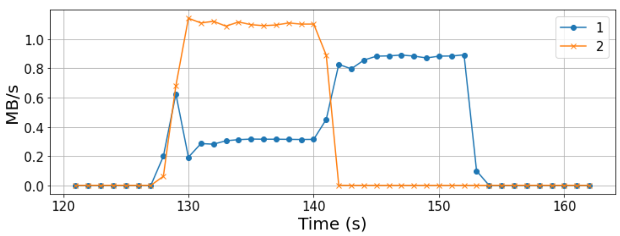

Figure 7: Allocation of Bytes in Two UEs with MaxTBS

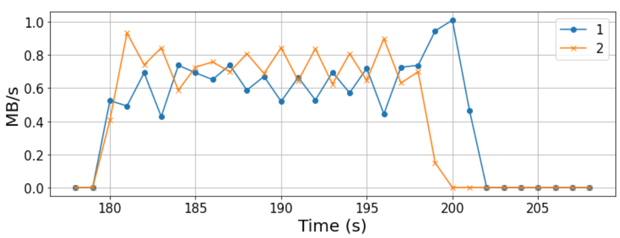

Figure 8: Allocation of Bytes in Two UEs (Equal MCS) with MaxTBS

each UE. In post processing, the accumulated bytes per second are calculated for each UE. The results are shown in Figures 6, 7, and 8.

In Figure 6, it can be seen that with Proportional Fair, the adjustment under equal conditions is almost perfect, with an initial transient due to the tests not starting at the same instant, and another at the end for the same reason. In Figure 7, with the MaxTBS algorithm, the UE with the best TBS receives the majority assignment, and when it finishes, the second UE can transmit at a higher bit rate. The different maxim throughput show the necessary MCs difference to achive this behavior. Finally, in Figure 8, two UEs with the same MCS receive random assignments with similar average results, using MaxTBS.

All the test at this article are configured with Single Input Single Output (SISO) antenna port configuration, with only one Component Carrier (CC), and without changes in Bandwith Part (BWP). The OAI gNB allows two CC, MIMO 2x2 and BWP management, and the schedulers assign all the resources available per UE. Other configurations, and handovers may be tested, too.

6 CONCLUSIONS

In this study, we thoroughly examined the architecture of the OAI RAN software, focusing on the operation of the existing scheduler and its integration into the protocol stack implementation. We also explored the system's initialization and configuration functions. Our primary contribution is the development and implementation of new scheduling algorithms within the OAI framework, enhancing the flexibility and performance of the scheduler.

We developed and integrated variants of the original proportional fair scheduler, including the max CQI (or max TBS) algorithm. Additionally, we introduced new coefficients that allow for greater flexibility in scheduling decisions, resulting in different scheduling behaviors depending on the configuration. The full development can be seen in our repository [9].

Our study establishes a foundation for future research in advanced scheduling algorithms for 5G networks. The flexibility and performance improvements demonstrated by our new schedulers provide a robust platform for further innovation. Future research could focus on dynamic and adaptive scheduling techniques, leveraging machine learning and AI to optimize resource allocation in real-time.

ACKNOWLEDGMENTS

This work was partially supported by CSIC R&D Project "5/6G Optical Network Convergence: an holistic view".

REFERENCES

[1] Bhavana D. and Shilpa Chaudhari. 2023. Performance Evaluation of OpenAirInterface's Scheduling Algorithms for 5G Networks. In *2023 4th International Conference for Emerging Technology (INCET)*. 1–4. https://doi.org/10.1109/INCET57972.2023.10170129

[2] Md. Emdadul Haque, Faisal Tariq, Muhammad R. A. Khandaker, Kai-Kit Wong, and Yangyang Zhang. 2023. A Survey of Scheduling in 5G URLLC and Outlook for Emerging 6G Systems. *IEEE Access* 11 (2023), 34372–34396. https://doi.org/10.1109/ACCESS.2023.3264592

[3] FlexRAN MOSAIC5G. 2023. OpenLTE: An Open-Source Implementation of the 3GPP LTE Specifications. https://mosaic5g.io/flexran/.

[4] Navid Nikaein, Mahesh K. Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. 2014. OpenAirInterface: A Flexible Platform for 5G Research. *SIGCOMM Comput. Commun. Rev.* 44, 5 (oct 2014), 33–38. https://doi.org/10.1145/2677046.2677053

[5] Wilder Peña, Walter Piastri, and Pablo Vázquez. 2023. Implementación de una Maqueta de Pruebas y Desarrollo de una Red 5G Stand Alone Completa. https://hdl.handle.net/20.500.12008/42953

[6] Mauri Seidel, Andreas Ingo Grohmann, Peter Sossalla, Florian Kaltenberger, and Frank H.P. Fitzek. 2023. How to Get Away with OpenAirInterface: A Practical Guide to 5G RAN Configuration. In *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. 1–6. https://doi.org/10.1109/ICECCME57830.2023.10252787

[7] Răzvan-Mihai Ursu, Arled Papa, and Wolfgang Kellerer. 2022. Experimental Evaluation of Downlink Scheduling Algorithms Using OpenAirInterface. In *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. 84–89. https://doi.org/10.1109/WCNC51071.2022.9771597

[8] Pablo Vazquez, Wilder Peña, Walter Piastri, Lucas Inglés, and Claudina Rattaro. 2024. MAQ5G: Deployment of a Complete 5G Standalone Network Testbed for Testing and Development. In *2024 Congreso de Tecnología, Aprendizaje y Enseñanza de la Electrónica (XVI Technologies Applied to Electronics Teaching Conference)*. IEEE.

[9] Pablo Vázquez, Wilder Peña, and Walter Piastri. 2024. MAQ5G. https://gitlab.fing.edu.uy/maq5g-pfc/maqueta-5g