

An Improved MAC Layer for the 5G NR ns-3 module

Natale Patriciello, Sandra Lagen, Lorenza Giupponi, Biljana Bojovic
{npatriciello, slagen, lgiupponi, bbojovic}@cttc.cat
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA)
Castelldefels, Barcelona, Spain

ABSTRACT

In this paper, we present a novel 5G NR simulator aligned with Release 15 TS 38.300. The work relies on previous implementations of LTE and mmWave modules. The focus of the paper is on the MAC layer, where we present the refactoring and the improvements to support OFDMA as per standard. A novel, user-friendly and modular interface is also proposed for the scheduler part, that allows a symbol-level distribution of resources. We go through the details of the implementation, and then we present scheduler results for a subset of schedulers that we propose. The code is available for interested users.

CCS CONCEPTS

• **Networks** → **Network simulations**; • **Computing methodologies** → **Discrete-event simulation**.

KEYWORDS

Network Simulator 3 (ns-3), New Radio (NR), MAC, Schedulers

ACM Reference Format:

Natale Patriciello, Sandra Lagen, Lorenza Giupponi, Biljana Bojovic. 2019. An Improved MAC Layer for the 5G NR ns-3 module. In *Proceedings of the 2019 Workshop on ns-3 (WNS3 2019)*, June 2019, Firenze, Italy. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

The 3rd Generation Partnership Project (3GPP) is devoting significant efforts to define the fifth Generation (5G) New Radio (NR) access technology [2], which has flexible, scalable, and forward-compatible Physical (PHY) layer to support a wide range of center carrier frequencies, deployment options, and a variety of use cases. Research institutions or Small and Medium Enterprise (SME)s that cannot develop sophisticated simulation models, due to the cost, time effort and human resources required, are at risk of being cut out from the early stages of the development process. Some of them rely on analytic methods, but the assumptions and the simplifications in all the segments of the network, limit the generality of the results. Moreover, it is tough to represent external network dynamics (such as the burstiness of data traffic) or the interaction

with the core network and the mobility of the users, without the support of a solid full-stack end-to-end simulation model.

While common low-level simulators focus on link level simulations, we are interested in having an overall view of the system, which starts from the application level to the PHY layer and includes an End-To-End (E2E) performance evaluation, from the User Equipment (UE) to the remote host. Our objective is to properly evaluate the performance of a sophisticated and flexible technology, like NR, and to be able to conduct interoperability experiments with other technologies. Hence, we transformed and adapted the mmWave module, developed by New York University and University of Padova [7], to be the first NR non-standalone module of ns-3. At CTTC, we also developed the LTE module [4], from which mmWave was derived.

In this paper, we present the Medium Access Control (MAC) layer of our NR module. We completely redesigned it from the original mmWave version, because of the lack of support of Orthogonal Frequency-Division Multiple Access (OFDMA) (the mmWave module at the time of writing only supported Time-Division Multiple Access (TDMA)-based access). In addition, we present a selection of scheduler strategies, to support resource allocation in NR OFDMA access, NR operational timing delays, and standard compliant message exchange procedures (scheduling requests and uplink grants) for uplink traffic. The scheduler code has been rewritten from scratch to eliminate the problems and the duplication of the code that are present in the mmWave and LTE modules. The details of this new implementation refactoring are not given in this paper due to space constraints. However, it is worth mentioning it, since the result is a much more modular code, which considerably facilitates the work of the user. Since the scheduler was one of the main challenges to tackle for an Long Term Evolution (LTE) user, we believe that this contribution is also an important improvement for the usability of the code. Also, the management of the LTE scheduler code is very time consuming, since every change introduced in the scheduler structure needs to be replicated for all the schedulers. The new implementation overcomes this limitations and also solves many maintenance problems. It has not been possible to compare our MAC layer with the original mmWave one, either from a functional or a performance point of view, giving the big differences in the feature list of the two modules. Instead, we provide a simulation set in an asymmetric scenario that proves the reliability of our changes, demonstrating that the simulated results are close to the theoretical ones. We also present some guidelines for authors that want to start their own module inside the ns-3 simulator, highlight our lessons learned, and discuss a potential future merging process with ns-3-dev.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WNS3 2019, June 2019, Firenze, Italy

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-9999-9/18/06...\$15.00

<https://doi.org/10.1145/1122445.1122456>

1.1 Related work

In the recent years a lot of effort has been done by New York University and the University of Padova to develop a simulator that allows simulations of communications in millimeter-wave (mmWave) bands, which represents a central technology of future 5G wireless cellular systems. Hence, they developed a mmWave module for the widely used ns-3 network simulator. Their module is still not part of the standard ns-3 distribution, but it implements a complete protocol stack, where the PHY layer supports a new mmWave-based channel, propagation, beamforming, and antenna models; and the MAC layer supports Time Division Duplex (TDD), TDMA MAC scheduling, and enhanced Hybrid Automatic Repeat reQuest (HARQ) for low latency applications. The higher layers are mostly based on ns-3 LTE module functionalities, thus still following 3GPP LTE specifications, but the authors extended them to involve some of the advanced features that are expected to emerge in 5G networks, such as dual connectivity [9] and low latency Radio Link Control (RLC) layer.

We shared our initial development process of the NR module in [5], where we presented the support for the novel 3GPP NR frame structure, with inclusion of the novel numerology concept introduced by NR, defined by different subcarrier spacings and cyclic prefix overheads. Then, we presented the Frequency Division Multiplexing (FDM) of numerologies feature, based on the LTE Carrier Aggregation functionality, which allows dividing the entire bandwidth in two or more bandwidth parts, in which the traffic can be routed without interacting with the other parts. The development was really useful to understand the inner parts of the mmWave module, and we used the lessons that we learned as a base to expand even more what is now our NR module.

1.2 Organization

The paper is organized as follows: in Section 2, we give an overview of the NR simulator components, before entering into the details of the MAC improvements in Section 3. We present the implemented scheduling policies in Section 4, and in Section 5 we present some results gathered in an asymmetric scenario. Then, in Section 7 we discuss some interesting points for a future merge with ns-3-dev, before concluding the work in Section 8.

2 NR MODULE OVERVIEW

We have designed the NR module to be able to perform E2E simulations of 3GPP-oriented cellular networks. As a starting point of our work, we used the mmWave module [7]. The implementation of the mmWave module started in a moment in time when NR 3GPP specifications were not available, and the general vision of the technology was not as solid as it is today. As a result, many implemented aspects from the development version where we started to work, were not standard compliant and need a revision. Many other things, such as the modeling of the channel and the beam management, are, on the contrary, entirely in line with the 3GPP standard, and therefore we have not modified them. We hope to merge the efforts and to be able in the future to share the code, following a model in which the user can take the part he or she needs.

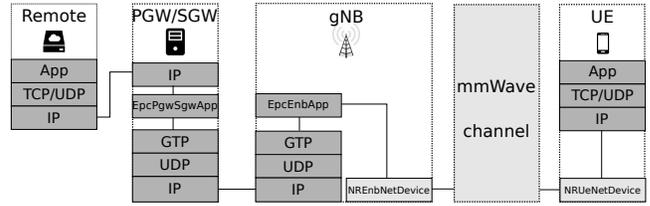


Figure 1: End-to-end class overview. In dark gray, we represent the existing, and unmodified, ns-3 and LENA components. In light gray, we represent the mmWave/NR components.

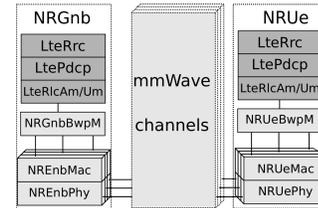


Figure 2: RAN class overview. In dark gray, we represent the existing, and unmodified, LENA components. In light gray, we represent the mmWave/NR components.

The mmWave module is defined starting from the ns-3 LTE module (LENA) [4], which has been entirely designed and developed at CTTC. As such, the NR module is also highly influenced by the previous design of the LTE module. In particular, both mmWave and NR modules reuse from LTE all the higher protocol (Radio Resource Control (RRC), RLC, Packet Data Convergence Protocol (PDCP), Non-Access Stratum (NAS)), as well as the Evolved Packet Core (EPC). As all the code is inside the ns-3 framework, the NR module will further benefit from additions that future contributors will make to the ns-3 simulator, and it can reuse exciting features such as the Direct Code Execution [8]: users can perform simulations with realistic TCP/IP implementations and existing applications.

We draw in Figure 1, the E2E overview of a typical simulation. On one side, we have a remote host (depicted as a single node for simplicity, but there can be many) that connects to an Service GateWay (SGW)/Packet data network GateWay (PGW), through a link. Such a connection can be of any technology available in ns-3. Currently, it is implemented through a single link, but it can be substituted by an entire subnetwork with many nodes and routing rules. Inside the SGW/PGW, the EpcSgwPgwApp encapsulates the packet using the GPRS Tunneling Protocol (GTP) protocol. Through an IP connection, which represents the backhaul of the NR network (again, represented with a single link, but the topology can vary), the GTP packet is received by the next-Generation Node B (gNB). Here, after decapsulating the payload, the packet is transmitted over the Radio Access Network (RAN), through the entry point represented by the class NRGNBNETDEVICE. The packet, if received correctly, is passed to higher layers by the class NRUENETDEVICE. The path crossed by packets in the UpLink (UL) case is the same as described above, but in the backward direction.

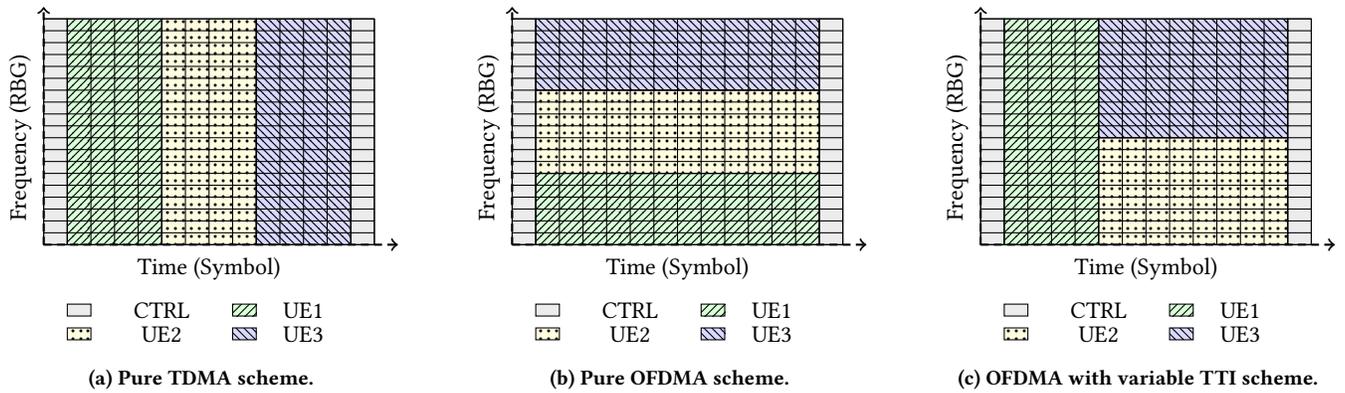


Figure 3: Possible allocation schemes for a slot.

Concerning the RAN, we detail what is happening between `NRGNBNETDEVICE` and `NRUENETDEVICE` in Figure 2. The `NRGNBMAC` and `NRUEMAC` MAC classes implement the LTE module Service Access Point (SAP) provider and user interfaces, enabling the communication with the LTE RLC layer. The module supports RLC Transparent Mode (TM), Saturation Mode (SM), Unacknowledged Mode (UM), and Acknowledged Mode (AM) modes. The MAC layer contains the scheduler (`NRMACSCHEDULER` and derived classes). Every scheduler also implements a SAP for LTE RRC layer configuration (`LTEENBRRC`). The `NRPHY` classes are used to perform the directional communication for both DownLink (DL) and UL, to transmit data and control channels. Each `NRPHY` class writes onto an instance of `MMWAVESPECTRUMPHY` class, shared between the UL and DL part. We did not modify the internal of `MMWAVESPECTRUMPHY`, and as by the original design, it contains many PHY-layer models: interference calculation, Signal-to-Interference-plus-Noise Ratio (SINR) calculation, the Mutual Information (MI)-based error model for LTE Turbo Codes (to compute the block error probability), as well as the Hybrid ARQ PHY-layer entity to perform soft combining.

Interesting blocks in Figure 2 are the `NRGNBBWPM` and `NRUEBWPM` layers. 3GPP does not explicitly defines them, and as such they are virtual layers, but they help to construct a fundamental feature of our simulator, the multiplexing of multiple Bandwidth Part (BWP)s. We have explained the design and the implementation of the bandwidth part manager in a paper presented at the 2018 edition of the workshop on ns-3 [5], and optimized the BWP configuration based on such development in [6].

3 MAC LAYER IMPROVEMENTS

We have implemented the MAC layer in the classes `NRENBMAC` and `NRUEMAC`. They interact directly with the physical layer, through a set of SAP APIs, and indirectly with the RLC layer. The messages exchanged through the API between RLC and MAC are captured and adequately routed by the bandwidth part manager. As an example, the RLC sends to the MAC many Buffer Status Report (BSR) messages (one per bearer), to inform the scheduler of the quantity of data currently stored in the RLC buffers. The scheduler, based on such information, will then take scheduling decisions. We have completely transformed the multiple access schemes, the UL scheduling

schemes, the scheduler timings, and the scheduler implementation part inside the MAC layer, as we detail in next subsections.

3.1 Multiple Access Schemes

We support OFDMA, as the first schedulers in LTE ns-3 did, but we adapted the code to be able to assign a variable number of Orthogonal Frequency Division Multiplexing (OFDM) symbols and Resource Block Group (RBG)s inside a slot. Visually, a TDMA-based scheme looks as depicted in Figure 3a. Three UEs are scheduled, each one during a period of time that spans some OFDM symbols and with data in all the RBGs. A pure-OFDMA scheme allocates data of different UEs on different RBGs, but using all the available OFDM symbols, such as in Figure 3b, as in LTE. The OFDMA-based scheme with variable Transmission Time Interval (TTI), instead, is the most flexible way to assign resources. It can allocate different RBGs with a limit on the total number of OFDM symbols. Moreover, we added another degree of flexibility, that allows scheduling UEs also following a TDMA-based scheme. An example is reported in Figure 3c: UE1 is allocated in a TDMA fashion in the first part of the slot, while UE2 and UE3 are scheduled in the rest of the OFDM symbols, each one with a different set of RBGs. This leads to an OFDMA with variable TTI scheme.

In the NR simulator, these multiple access schemes, as well as the scheduler policies for them, can be freely chosen (as we will explain later). However, it is worth noting that there are physical limitations when applying them in different spectrum regions. For instance, in the higher spectrum region (e.g., mmWave part) it would be more difficult to use the pure-OFDMA scheme due to incompatibility with the radio-frequency architectures that are based on single-beam capability [3]. As such, the current implementation supports OFDMA with variable TTI under single-beam capability only, i.e., only UEs associated to the same beam can be allocated to the same OFDM symbols in different RBGs.

3.2 Scheduling Schemes

NR, like LTE, uses dynamic scheduled-based access for DL, based on which the gNB makes the scheduling decisions. Each UE monitors the Physical Downlink Control Channel (PDCCH), and upon the detection of a valid Downlink Control Information (DCI), follows

the given scheduling decision and receives its DL data. In the case of UL, NR considers UL grant-based and UL grant-free access (also known as autonomous UL) schemes [2]. The former is the conventional dynamic scheduled-based access, as per LTE DL/UL and NR DL, based on which the gNB makes the scheduling decisions in both UL and DL. Each UE monitors the PDCCH and, upon the detection of a valid DCI, follows the given scheduling decision and transmits its UL data. The latter is a contention-based scheme. At the time, we have implemented only the UL grant-based access, as per NR specifications, but the UL grant-free implementation is in our future roadmap. As a result, the NR module supports dynamic scheduled-based accesses both for DL and UL.

The design that we followed aims to adopt different scheduling policies (round-robin, proportional fair, etc.) to a TDMA with variable TTI, or an OFDMA with variable TTI access scheme. Also, we aim to reduce to the minimum the amount of duplicated code, while respecting the FemtoForum specification for LTE MAC Scheduler Interface. To do so, we considered that the primary output of a scheduler is a list of DCIs for a specific slot, each of which specifies (among other values) three crucial parameters. The first is the transmission-starting symbol, the second is the duration (number of symbols) and the last one is the RBG bitmap, in which a value of 1 in the position m represents a transmission in the RBG number m . This is compliant with DL and UL resource allocation Type 0 in NR [1, Sect. 5.1.2.2 and Sect. 6.1.2.2], as far as frequency-domain is concerned, and follows the standard time-domain resource allocation that includes Start and Length Indicator Value (SLIV) both for DL and UL [1, Sect. 5.1.2.1 and Sect. 6.1.2.1].

Scheduler Timings: We consider that the scheduler works "ahead" of time: at time t , when the PHY is transmitting slot x over the air, the MAC is working to allocate slot $x + d$, where d is a configurable delay, defined as a function of the number of slots. It represents the operational latency (in the simulator, we use the attribute `L1L2CTRLLATENCY` and `L1L2DATA_LATENCY` of the class `NRPHYMACCOMMON`). For the DL DCIs, this is the only delay to consider: when the slot $x + d$ is over the air, the DL DCI is transmitted in the first symbols and will apply for the same slot. However, for the UL case, we must consider an additional delay which represents the time needed by the UE to decode the DCI and to prepare the UL data to transmit. The standard refers to this further delay as K_2 [1, Sect. 6.1.2.1], which is measured in number of slots and can take any integer value from 0 to 7. We model it through the attribute `ULSCHEDDELAY` of the class `NRPHYMACCOMMON`. To keep it in consideration, if the PHY is transmitting over the air the slot x , the MAC will work on the UL part of the slot $x + d + K_2$. These UL DCIs are transmitted over the air in slot $x + d$, and the UE has K_2 slots of time for preparing its data. Figure 4 illustrates the scheduler operation and the DL/UL transmissions by taking into account these timings, for $K_2=2$ slots and `L1L2DATA_LATENCY=L1L2CTRLLATENCY=2` slots.

3.3 UL handshake

We have improved the dynamic scheduled-based access for UL (i.e., the UL grant-based scheme), as follows: upon data arrival at the UE RLC queues, the UE sends a Scheduling Request (SR) to the gNB through the Physical Uplink Control Channel (PUCCH) to

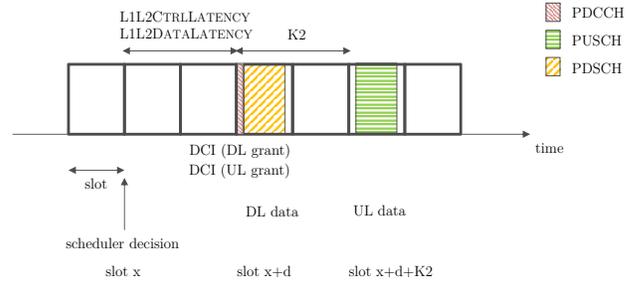


Figure 4: Scheduler timings in the ns-3 NR simulator, for $K_2=2$ slots and `L1L2DATA_LATENCY=L1L2CTRLATENCY=2` slots.

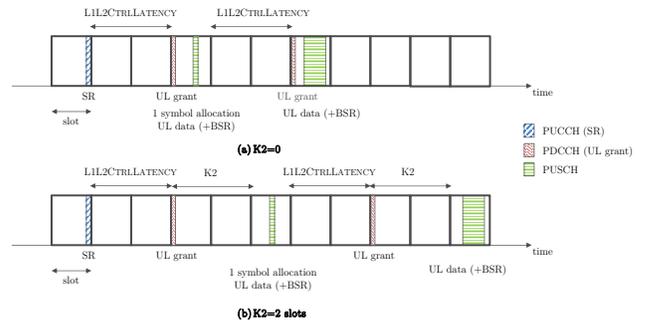


Figure 5: UL handshake procedure for NR UL grant-based access, including NR timings and processing delays, as implemented in the ns-3 NR simulator. (a) $K_2=0$, (b) $K_2=2$ slots.

request an UL grant from its gNB. Then, the gNB sends the UL grant (DCI in PDCCH) to indicate the scheduling opportunity for the UE to transmit. Note that the first scheduling assignment is blind since the gNB does not know the buffer size at the UE yet. In this regard, since this is implementation-specific, we assume that the first scheduling opportunity consists of the minimum amount of symbols that permit at least a 4 bytes transmission. In the majority of cases, this value equals to 1 OFDM symbol. Next, the UE, after receiving the UL grant, performs the data transmission in the Physical Uplink Shared Channel (PUSCH), which may contain UL data and/or BSR in the PUSCH. After that, if a BSR is received, the gNB knows the UE RLC buffer status and can proceed with another UL grant to account for the remaining data. Note that the main difference in the NR module with respect to mmWave and LTE ns-3 modules is that we have introduced the SR in the PUCCH and the BSR can only be sent in conjunction with the MAC Packet Data Unit (PDU) (according to NR specifications, the BSR is part of the MAC header), while in previous ns-3 modules the BSR was sent periodically and ideally.

Before sending the UL grants, `L1L2CTRLATENCY` delay has to be considered at the gNB side. Also, upon reception of an UL grant, the UE should send UL data and/or BSR after K_2 slots, being K_2 indicated in the UL grant. So, these two parameters influence the UL handshake. In Figure 5, we show the UL handshake, including also the timings and processing delays that influence it (i.e.,

L1L2CTRLLATENCY and K2) for K2=0 (top) and K2=2 slots (bottom). In this example, we assume a TDD slot structure, with 14 symbols per slot. PDCCH is sent in the 1st symbol, PUCCH in the 14th symbol, while the symbols in between are devoted to shared channels that may contain data (Physical Downlink Shared Channel (PDSCH) and/or PUSCH). Our implementation in the ns-3 NR simulator follows exactly the handshake and timings that are illustrated in Figure 5. The BSR is prepared shortly before the PHY transmission in the UL, reflecting the status of the RLC queue without including the current transmission.

4 IMPLEMENTATION OF SCHEDULER POLICIES AT THE GNB

The core class of the NR schedulers design is `NRMacSchedulerNs3`. This class defines the core scheduling process and splits the scheduling logic into logical blocks. The FemtoForum API splits the UL and the DL scheduling. In the following, we will consider only the DL, but the description also applies to the UL case. The differences lie in the variable and function naming, as well as the delays involved, as explained before. As a starting point, we prepare a list of active UE and their requirements, divided by the specific beams they belong to.

We start with the scheduler implementation details for OFDMA-based schemes. The first step of the procedure consists of distributing OFDM symbols among multiple beams. We need this block for the OFDMA-based schemes because we chose to support single-beam capability only. At high frequencies, the beam is shaped after digital-to-analog conversion due to limitations in the implementation phase. Therefore, with analog beamforming, there is the constraint that a receive or transmit beam can only be formed in a single direction at any given time instant, meaning that if we want to transmit towards two UEs with different beams, we must do so in different time instants. We provide two different ways to assign symbols to the beams: in a load-based or round-robin fashion. We consider as the beam load the sum of the bytes queued in the RLC layer of the UEs that belong to that beam. We consider that the round-robin assignment merely assigns the same number of OFDM symbols to all beams.

After the symbols/beam selection in OFDMA schedulers, it is necessary to distribute the available RBGs in the time/frequency domain among active UEs in each beam. This step depends on the specific scheduling algorithm that the user has chosen. The RBGs can be distributed following a round-robin, proportional fair, or maximum rate algorithm. The resources to be allocated are groups of RBGs spanned over one, or more, symbols.

Finally, the last step consists in the creation of the corresponding DCI, based on the number of assigned resources made in the previous block. The assigned RBGs should be grouped to create a single block for each UE. Then, the RBG bitmap is created¹, so that DCIs for different UEs do not overlap. The bitmap will be an input, later on, for the PHY layer. At the transmission or reception time, the PHY translates the bitmap into a vector of enabled Physical Resource Block (PRB). As the standard indicates in [1, Sect. 5.1.2.2

¹The ns-3 NR module follows NR resource allocation Type 0, as per [1], in which the resource allocation is specified through a bitmap. It provides more flexibility to the scheduler operation, as compared to NR resource allocation Type 1 that specifies the PRB start and number of PRB allocated.

and 6.1.2.2], each RBG is grouping 2, 4, 8, or 16 PRB depending on the BWP size. Then, the transmitter distributes the power, and the receiver decodes, only among these active PRB.

The design also takes into consideration HARQ retransmissions. They have a higher priority in the scheduling policies. When a NACK is received, the scheduler takes the old DCI and tries to put it in the current slot for retransmission. If that is not possible, then it will be queued for the next slot. It is important to remark that the simulator only supports a round-robin policy to select the HARQ process to retransmit.

The user can select different schedulers and different assignment modes by swapping class name during the configuration phase. The available OFDMA schedulers are `NRMacSchedulerOFDMARR` (round-robin), `NRMacSchedulerOFDMAPF` (proportional fair), and `NRMacSchedulerOFDMAMR` (maximum rate). Our OFDMA schedulers are all using the variable TTI strategy, so they are allowed to create TTIs of different length. The extension to pure OFDMA schedulers is straightforward.

For TDMA-based schedulers, the first step (symbols/beam selection) is not performed, as entire symbols are assigned to the UEs and the PHY layer is perfectly capable of switching the beam in time (under the single-beam capability assumption explained before). Therefore, the assignment phase is directly executed, in which a particular scheduler can decide how many OFDM symbols are assigned to a certain UE. We support round-robin (`MmWaveMacSchedulerTDMARR`), proportional fair (`MmWaveMacSchedulerTDMAPF`), and maximum rate (`MmWaveMacSchedulerTDMAMR`) schedulers.

These classes, no matter the access mode, follow the same principles:

- *Round-robin*: The scheduler evenly distributes the available RBGs among UEs associated with that beam (OFDMA), while for TDMA evenly distributes the available symbols.
- *Proportional fair*: In the OFDMA mode, the scheduler evenly distributes the available RBGs among UEs according to a metric that considers the actual rate, based on the Channel Quality Indicator (CQI) elevated to α and the average rate that has been provided in the previous slots to the different UEs. By changing the α parameter the metric also changes. For $\alpha = 0$, the scheduler selects the UE with the lowest average rate. For $\alpha = 1$, the scheduler selects the UE with the largest ratio between actual rate and average rate. For TDMA, the resources to distribute are entire symbols.
- *Maximum rate*: The scheduler evenly distributes the available RBGs (or the available symbols in case of TDMA) among UEs according to a maximum rate metric that considers the actual rate (based on the CQI) of the different UEs.

In the UL, we currently support only TDMA. This means that, even for OFDMA schedulers, such a phase is treated as it was in the TDMA schedulers.

5 SIMULATION CAMPAIGN

5.1 Scenario

Due to space constraints, we will evaluate only the DL access, considering a simple scenario composed of 1 gNB and 4 UEs. In our scenario, four UEs are allocated in two different beams (two UEs per beam), at a fixed distance from the gNB, in such a way that UE1,

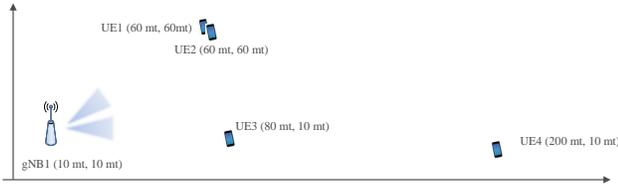


Figure 6: Scenario topology.

Parameter	Value
Channel model	3GPP, LoS
Channel bandwidth	100 MHz
Central frequency	28 GHz
Scenario	Urban Macro
Shadowing	disabled
Beamforming method	Beam search
MCS	Adaptive
Encode latency	2 slots
Numerology	3

Table 1: Scenario simulation parameters.

UE2, and UE3 have MCS 26, while UE4 has MCS 20. Thus, in one beam both UEs have symmetric channel conditions (UE1 and UE2), while in the other beam, UE3 and UE4 have asymmetric channel conditions. All UDP constant bit rate applications start at the same time.

The specific 2D location of UEs is depicted in Figure 6. As simulation parameters, we use a carrier frequency of 28 GHz, an NR numerology $\mu=3$, a channel bandwidth of 100 MHz, packet size=1370 B, and a varying load per application. We calculated the theoretical saturation and derived the rates used in the simulation. The first rate of 40 Mbps allows testing the schedulers in a close-to-saturation mode, but still with some available space, while the second rate of 80 Mbps makes the system entirely saturated, to see how the schedulers are responding in this situation. In the next Table 1, we summarize other relevant simulation parameters. It is important to note that simulation results have been averaged over multiple runs to get statistical significance.

5.2 Theoretical calculations

Before proceeding, we present some simple computations that allow us to define the theoretical expectations.

Theoretical Throughput for one UE at maximum MCS. The configuration of $\mu=3$ gives a slot length of 125 μs and a resource block width of 1.44 MHz. Thus, in our scenario, there are available 69 Resource Block (RB)s (with 12 subcarriers each). As a configuration parameter, we assume that $\frac{1}{4}$ of the subcarriers are used for reference signals so that there are 9 subcarriers useful for data transmission, in each RB. In the time domain, there are 14 symbols. The first one is dedicated to DL control, and the fourteenth to UL control. Hence, only 12 symbols on each slot can be filled with data. To compute the maximum potential throughput, we consider the maximum MCS (28) that corresponds to modulation 64-QAM with

code rate=0.92. Finally, the Code Block (CB) size is 6144 with a CRC length of 24 bits. The result for the maximum throughput is, therefore, 327.7 Mbps. In our scenario, the highest MCS is lower than 28, due to the users' location, so that the expected UE throughput will be lower than the maximum one. The expected result computed here is however useful to understand the simulation results.

Theoretical Latency. The different factors that affect the delay are the queuing time at all the levels, the waiting time until the slot boundary, two unused slots due to the encode processing latency, the air transmission time, and then finally 100 μs of decoding latency. In case of pure OFDMA, with $\mu=3$, the minimum delay (without queuing time and assuming that the packet is arriving to lower layers exactly at the beginning of the slot) is $2 * slot + slot + 100 \mu s = 475 \mu s$. On average, we can say that any packet will wait half of a slot before reaching the slot boundary. Therefore, using $0.5 * slot$ as waiting time, under a no-saturation regime (so, without queuing time) we expect an average minimum delay of $0.5 * slot + 2 * slot + slot + 100 \mu s = 537.5 \mu s$.

5.3 Simulation results

In this section, we compare the different scheduler policies (round-robin, proportional fair, and maximum rate) for TDMA and OFDMA with variable TTI schemes, as detailed in Section 4. In the OFDMA case, we apply a load-based strategy for distributing symbols to the different beams, and only after that phase we distribute resource blocks for the UEs that belong to a beam. Such a strategy adds fairness to any scheduler type: if a UE has a bad channel condition, it will have more bytes in the queue with respect to another UE, with the same traffic pattern, but with better conditions. Hence, the load-based strategy (that is based on queue occupancy) will assign more symbols to the UE with the worst channel conditions.

In the first beam, the UEs have the same channel conditions, and therefore the same potential achievable rate. In the second beam, one UE is more distant than the other from the gNB. Hence, we expect a lower MCS and lower performance at UE4 than at UE3.

We present in Table 2 the result for the round-robin scheduler. For the non-saturated case (the leftmost columns) we can see that TDMA and OFDMA presents delay differences: in OFDMA, the data is spanned over many symbols, that adds time on the sum, while in most cases TDMA transmissions last less than two symbols. The difference in throughput is due to the load-based strategy that is done at the beginning of each assignment, to distribute the symbols to all the beams that will be active in the slot. As explained before, this introduces differences in the throughput for the different UEs, while in the TDMA case is evident how the UEs with the same channel conditions achieve the same performance.

Still in Table 2, on the right side, we can see the results for the saturated case. We still have a difference in delay, as well as in throughput. The TDMA assignment is generally working better for the round-robin scheduler, with lower delays and higher throughput.

We represent in Table 3 the result obtained by using the proportional fair scheduler. In TDMA, it works as expected, properly distributing the resources to UEs in such a way that all UEs have the same average throughput (and same average delay) at the simulation end. There is a slight difference in the delay, as the number of

UE	Offered load: 40 Mbit/s				Offered load: 80 Mbit/s			
	TDMA		OFDMA		TDMA		OFDMA	
	delay (ms)	goodput (Mb/s)	delay (ms)	goodput (Mb/s)	delay (ms)	goodput (Mb/s)	delay (ms)	goodput (Mb/s)
1	0.61	40.00	43.04	38.87	183.75	70.14	312.29	63.17
2	0.57	40.00	13.30	39.66	125.57	73.29	287.78	64.51
3	0.60	40.00	0.66	40.00	158.59	71.51	180.65	70.33
4	100.02	37.33	103.58	37.24	634.71	45.66	641.77	45.32

Table 2: Delay and goodput results obtained with the round robin scheduler.

UE	Offered load: 40 Mbit/s				Offered load: 80 Mbit/s			
	TDMA		OFDMA		TDMA		OFDMA	
	delay (ms)	goodput (Mb/s)	delay (ms)	goodput (Mb/s)	delay (ms)	goodput (Mb/s)	delay (ms)	goodput (Mb/s)
1	0.60	40.00	0.58	40.00	339.96	61.67	314.38	63.03
2	0.61	40.00	0.73	40.00	339.94	61.67	314.94	63.01
3	0.65	40.00	0.77	40.00	339.91	61.68	440.16	56.23
4	0.63	40.00	0.73	40.00	341.05	61.62	440.21	56.23

Table 3: Delay and goodput results obtained with the proportional fair scheduler.

UE	Offered load: 40 Mbit/s				Offered load: 80 Mbit/s			
	TDMA		OFDMA		TDMA		OFDMA	
	delay (ms)	goodput (Mb/s)	delay (ms)	goodput (Mb/s)	delay (ms)	goodput (Mb/s)	delay (ms)	goodput (Mb/s)
1	0.48	40.00	1.01	40.00	0.51	79.99	302.88	62.50
2	0.49	40.00	9.66	40.00	0.49	79.99	207.34	67.70
3	0.53	40.00	0.55	40.00	0.54	79.99	0.88	79.99
4	198.40	34.64	28.27	34.64	1385.76	5.56	889.68	39.57

Table 4: Delay and goodput results obtained with the maximum rate scheduler.

symbols assigned is always less in the TDMA case (as, in OFDMA, the data is spanned over multiple symbols). When looking to the saturated case, on the right, we observe that the OFDMA scheduler is achieving the same rate for the UEs within a beam (i.e., UE1 and UE2 get the same goodput, and UE3 and UE4 as well), but users in different beams get a different goodput. The reason lies in the symbol allocation per beam, which follows a load-based metric (as said before, it allocates the number of symbols based on the queue occupancy). Then, within a beam, all the UEs get the same amount of resources.

Finally, we report in Table 4 the results of the maximum rate scheduler. The scheduling rationale is to allocate more resources to the UEs able to achieve higher throughput. The results on the right (40 Mbit/s load) show that the scheduler is assigning the maximum amount of data to the first three UE (the ones with the best channel condition) and what is left, is given to UE4. This is reflected in the delay and throughput performance. For OFDMA, instead, we see that the load-based strategy for symbol assignment is introducing some variability, and is offering more symbol to the beam that contains UE4. This reflects in a performance which is more balanced with respect to the previous case.

If we move to the right part of Table 4, we see that there is almost no space for the poor UE4 in the TDMA case. In the OFDMA case, due to the load-based strategy, it has more area, that is eaten up by its neighbor (UE3) that has better channel conditions. In the end, in absolute value, maximum rate OFDMA is performing with more fairness, but as the name of the scheduler suggests, it is not recommended to use it in a production environment. In general, MR it is not a good strategy to perform fair scheduling, and our results confirm this finding.

6 FUTURE WORKS

The work on the NR simulator is still ongoing. The first release, which can be obtained at this link <https://5g-lena.cttc.es/>, includes the following set of features: NR frame structure, OFDMA, basic schedulers, scheduler timings, UL grant-based scheme, FDM of numerologies and bandwidth part manager. Next releases will include a complete refactoring of the PHY, with a 3GPP NR compliant Link to System Mapping following details of TS 38.212 and TS 38.214. Currently, both the mmWave and NR modules use the PHY proposed in the LTE module, which means for example that Turbo Codes are considered and not Low-Density Parity-Check (LDPC) codes. We are working towards the support of LDPC coding, with

appropriate block segmentation, revised Modulation and Coding Scheme (MCS) tables, and HARQ.

Furthermore, the road-map currently foresees effort in the area of NR use in unlicensed spectrum, both below and above 6 GHz. This is a key feature recently committed as a work item in 3GPP, which will pave the way for novel interesting verticals, like Industry 4.0, among others. Finally, in the long term, we would like to devote more efforts to the redefinition of the software for the upper layers of the protocol stack and the core.

7 TOWARDS NS-3 DEVELOPMENT VERSION: LESSONS LEARNED

Before concluding our paper, we would like to emphasize the work that we are carrying out to be able to merge our finalized module inside the development version of ns-3. We are working to extend the number of provided examples, that span from a typical and simple usage to more sophisticated examples using all the features. We are adding unit and system-level tests to the mix, following a test-first style of development.

In the case of entire new modules there is no clear development pattern. First of all, the new module may require modifications to the existing modules. To give an example, our NR module needs edits to some part of the LTE stack. In general, these changes may impact existing functionalities or worse conflicts with other external modules. Then, the development of ns-3 is not stopping during the module development. Hence, some maintainer can introduce changes in the module dependency that can conflict with the code that has been developed in parallel in the new module.

Given the above explanation, it is not a surprise that many people follow the "hard-fork" policy when developing external modules. The strategy consists on forking ns-3 at a particular version and then decoupling the own work with what is happening in ns-3. In the luckiest case, there is a periodic merge of ns-3 advancements in the private fork, while in the typical case we developers advance their work only until a paper is submitted for publication.

On the contrary, we stick to an "upstream-first" strategy. When we started developing on top of the mmWave module, we planned an ns-3-dev migration as well. In the hope that our experience will help others to do the same, for the benefits of all ns-3 users, we share the most important points that helped us have an entire module that is based on the upstream version of ns-3:

- We did not copy an ns-3 release in another directory and then developed our module inside this newly created directory. Instead, we used a git copy of the ns-3-dev repository, and started preparing the module there;
- We did not add the module files into the git history of the central ns-3 repository. Instead, the history was kept as a separated git repository. In this way, updating the main repository (maintaining its history) was as easy as doing a simple command;
- We put every modification to the upstream ns-3 into a dedicated branch, ready to be submitted for review;
- Our module release consists in a single directory that can be put under the src/ path of any ns-3-dev installation.

In this way, the developed module has an independent run from the ns-3 core, but still the two are strongly connected. If all the

external modules followed this strategy, there would be another advantage: coexistence tests between different modules could be run more easily, and developers and institutions could share more effectively their code.

8 CONCLUSIONS

In this paper, we have presented the MAC layer of a novel NR module. We completely redesigned it because of the lack of different basic standards features like OFDMA-based access, schedulers to allocate time and frequency resources, support for the UL scheduler operational delay, and the 3GPP-compliant message exchange procedure for uplink traffic. We have also provided a substantial reorganization of the critical scheduler functionality, compared to previous LTE and mmWave modules, which facilitates maintenance, usability and modularity. We have provided a simulation set in an asymmetric scenario that proves the reliability of our changes, demonstrating that the simulated results are close to the theoretical ones. Examples and tests are also available, together with the source code, for all the interested users. Finally, we have presented some guidelines for authors that want to start their own module inside the ns-3 simulator, discussing our learned lessons.

ACKNOWLEDGMENTS

This work was partially funded by Spanish MINECO grant TEC2017-88373-R (5G-REFINE) and Generalitat de Catalunya grant 2017 SGR 1195. Also, it was supported by InterDigital Communications, Inc. In particular, authors would like to thank A. Demir, S. Goyal, and M. Beluri, for their fruitful discussions and advice on the development.

REFERENCES

- [1] 3GPP TS 38.214 Release 15, v15.2.0, June 2018. *TSG RAN; NR; Physical layer procedures for data*. 3GPP TS 38.214.
- [2] 3GPP TS 38.300 Sept. 2018. *TSG RAN; NR; Overall description; Stage 2 (Release 15), v15.3.0*. 3GPP TS 38.300.
- [3] J. G. Andrews et al. 2017. Modeling and analyzing millimeter wave cellular systems. *IEEE Trans. Commun.* 65, 1 (Jan. 2017).
- [4] Nicola Baldo, Marco Miozzo, Manuel Requena-Esteso, and Jaume Nin-Guerrero. 2011. An Open Source Product-oriented LTE Network Simulator Based on Ns-3. In *Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '11)*. ACM, New York, NY, USA, 293–298.
- [5] B. Bojovic, S. Lagen, and L. Giupponi. 2018. Implementation and evaluation of frequency division multiplexing of numerologies for 5G new radio in ns-3. *submitted to ns-3 Workshop 2018* (2018).
- [6] S. Lagen, B. Bojovic, S. Goyal, L. Giupponi, and J. Mangues. 2018. Subband configuration optimization for multiplexing of numerologies in 5G TDD New Radio. *IEEE Int. Symp. Personal, Indoor and Mobile Radio Commun.* (Sept. 2018).
- [7] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi. 2018. End-to-End Simulation of 5G mmWave Networks. *IEEE Communications Surveys Tutorials* 20, 3 (thirdquarter 2018), 2237–2263. <https://doi.org/10.1109/COMST.2018.2828880>
- [8] Hajime Tazaki, Frédéric Uarbani, Emilio Mancini, Mathieu Lacage, Daniel Camara, Thierry Turlletti, and Walid Dabbous. 2013. Direct Code Execution: Revisiting Library OS Architecture for Reproducible Network Experiments. In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT '13)*. ACM, New York, NY, USA, 217–228.
- [9] Tommaso Zugno, Michele Polese, and Michele Zorzi. 2018. Integration of Carrier Aggregation and Dual Connectivity for the Ns-3 mmWave Module. In *Proceedings of the 10th Workshop on Ns-3 (WNS3 '18)*. ACM, New York, NY, USA, 45–52. <https://doi.org/10.1145/3199902.3199909>