# The OpenAirInterface 5G New Radio Implementation: Current Status and Roadmap

Florian Kaltenberger, Guy de Souza, Raymond Knopp
EURECOM
Campus SophiaTech, 06410 Biot, France
Email: florian.kaltenberger@eurecom.fr

Hongzhi Wang
TCL 5G Labs
Campus SophiaTech, 06410 Biot, France

***Abstract*—OpenAirInterface (OAI) is an open-source project that implements 3GPP technology on general purpose x86 computing hardware and off-the-shelf software defined radio cards like the USRP. At its base OAI implements 4G LTE, but recently we have also started implementing 5G New Radio (NR). In this paper we describe the OAI 5G NR project, the current state of its development, and the roadmap for the future. 5G NR is much more demanding on both processing power, latency, and radio capabilities compared to 4G LTE. At the time of writing, we implemented all the necessary functions to support basic downlink functionality at both gNB and UE showing the feasibility to run 5G NR in real-time on a software defined radio platform. The roadmap is to have a fully standard compliant implementation of 5G NR that is inter-operable with commercial equipment by the end of 2019.**

## I. Introduction

The 5th generation (5G) mobile broadband standard is finally here. 3GPP Release 15 has been frozen in summer 2018 and this release includes a brand new core network and radio interface, called 5G new radio (5G-NR). The network has been designed from ground up to support enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (URLLC), as well as massive machine type communications (mMTC) enabling new use cases for a large variety of industries. This has been achieved by a large number of new features compared to 4G LTE, such as flexible subcarrier spacing and slot lengths (also called numerology), increased bandwidth (up to 400MHz), flexible slot structure (including mini-slots and slot aggregation). 5G-NR also includes new channel codes: polar codes for control and LDPC for data. A good overview of all new features is given in [1], [2].

Trials are currently underway in many parts of the world and the first commercial products (both infrastructure and mobile terminals) are expected to hit the market in the first half of 2019. End users will probably have to wait until the end of 2019 to be able to enjoy 5G coverage from their mobile operator.

OpenAirInterface is an open source initiative that today provides a 3GPP compliant reference implementation of eNodeB (eNB), User Equipment (UE), and evolved packet core (EPC) that runs on general purpose computing platforms (x86) together with off-the-shelf software defined radio (SDR) cards like the ETTUS USRP, Lime SDR, and ExpressMIMO2. It allows users to set up a compliant 4G LTE network and inter-operate with commercial equipment.

The OpenAirInterface team has been working on an implementation of 5G-NR since 2017 with a focus on the eMBB use case. A first pre-5G demonstration was given at Mobile World Congress 2018 where we had shown the feasibility of maintaining a throughput of 300Mbps over a 80MHz channel using the new 5G-NR LDPC channel coding. This was achieved by offloading the most computationally expensive task, the LDPC decoder, to an FPGA. Using a software-only implementation of the LDPC decoder, a throughput of 150Mbps can be supported.

In this paper we describe the current status of the implementation of 5G-NR in OpenAirInterface and the upcoming roadmap. We start by describing the main features of 5G-NR in Section II. In Section III we describe the software architecture and interfaces of our implementation. This also includes an evolution of the FAPI interface between PHY and MAC, which was already used in our eNB implementation. In Section IV we describe the current implementation status and in Section VI we describe the upcoming roadmap.

It should be noted that the while other software defined radio implementations of 5G new radio exist [3], OpenAirInterface is the only project that is 100% open source and empoys a unique open source software license, called the OAI public license [4].

## II. 5G New Radio

Initial deployments of 5G-NR will most likely use the architecture option 3 of 3GPP, also called EUTRA-NR dual connectivity (EN-DC). In this option the 5G cell is operating under the control of a 4G cell, which serves as an anchor to the system and carries all control plane traffic. UEs first need to connect to the 4G network and will receive all the necessary configuration to connect to a 5G cell through RRC signaling on the 4G link. This setup will allow a smooth migration from 4G to 5G.

3GPP has also defined a new 5G Core architecture that supports service delivery over wireless, fixed or converged networks. This new 5G Core (5GC) uses a cloud-aligned Service-Based Architecture (SBA) that supports control plane function interaction, re-usability, flexible connections and service discovery that spans all functions [5].

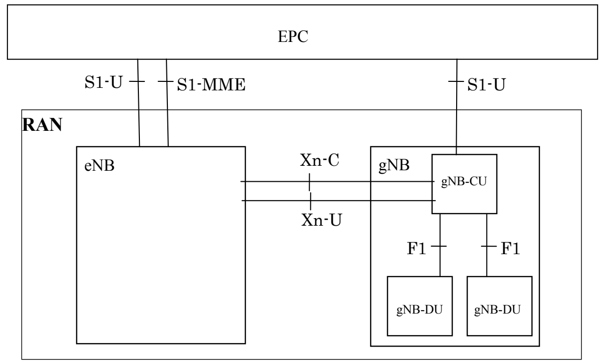


Fig. 1. Architecture of the EUTRA-NR dual connectivity (EN-DC) network

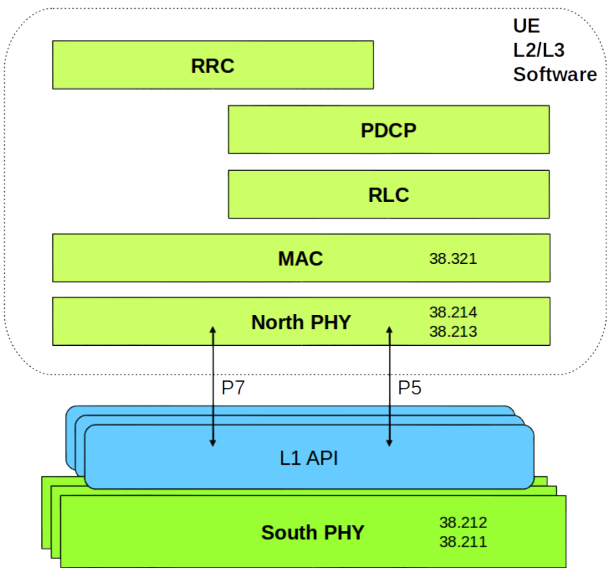


Fig. 2. Architecture of the gNB with FAPI interfaces [6]

Since the first 5G-NR deployments in Europe are going to use the EN-DC architecture we will use it as an example to describe our implementation next.

## III. gNB Architecture and Interfaces

The architecture of the EN-DC network is depicted in Figure 1. Here the LTE eNB takes the role of the master and the 5G gNB the role of the slave, and they are connected over the Xn interface. In this architecture, the anchors of the control plane are always located in the LTE eNB, that is, the S1-MME interface is terminated by the eNB.

Compared to a 4G eNB, a 5G gNB can be separated into a centralized unit (CU) and one or more distributed units (DUs), which are connected over the F1 interface. The CU contains the functionality of the PDCP as well as RRC and has interfaces to the LTE eNB (Xn) and optionally also to the core network (S1-U). The DU contains the functionality of RLC, MAC, and PHY.

Further splits can be employed within the DU, but this is out of the scope of 3GPP. In OpenAirInterface we have decided

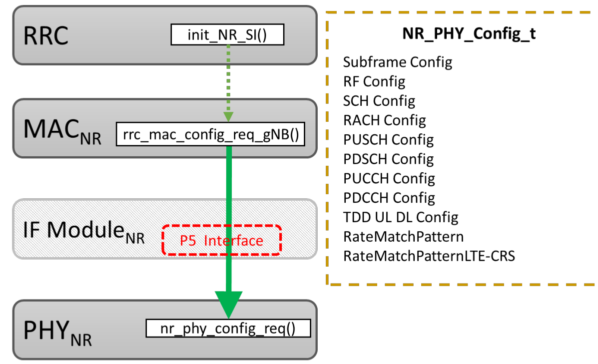


Fig. 3. FAPI P5 procedure [8]

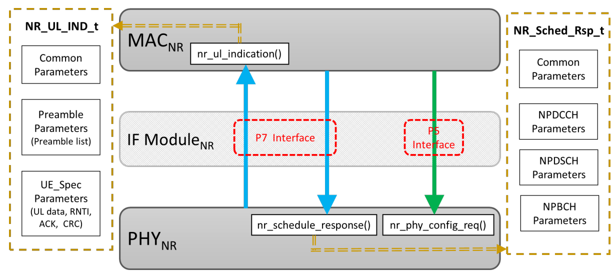


Fig. 4. FAPI P7 procedure [8]

to use the FAPI interface between PHY and MAC (see Figure 2). This interface has originally been specified by the Small Cell Forum [7] and has been extended to 5G-NR by us [8]. A similar interface has also been specified for the UE [6]

The FAPI P5 interface is used for transmitting static configuration from the MAC to the PHY and the P7 interface is used for transmitting data or semi-dynamic configuration from the MAC to the PHY. The basic P5 procedure is depicted in Figure 3. First, the gNB reads the configuration parameters from a configuration file and passes them to the RRC. The RRC then passes the MAC and the PHY configuration parameters further down to the MAC using an internal function call (in the future this will be transported over the F1 interface). The MAC finally passes on the necessary PHY configuration parameters over the FAPI P5 interface.

The basic FAPI P7 procedure is depicted in Figure 4. It consists of four basic messages: the uplink indication message (NR_UL_INDICATION), which is sent every slot to the MAC to provide synchronization, regardless if anything was received on the uplink. The MAC replies with a schedule response message (NR_SCHEDULE_RESPONSE), which carries information abourt the BCH and DCI PDUs as well as the DCI parameters. The messages TX_REQ and RX_IND carry the MAC PDU from the MAC layer to the PHY layer and vice versa.

## IV. Current Implementation Status

This section describes the implementation status as of January 2019. Since this project is under active development, please check the project webpage[1] for updated information.

As already noted, we are implementing the EN-DC architecture, which also requires an LTE cell. However, in our initial implementation, since we have full control of the gNB and UE, we will emulate the LTE link. In fact all the configuration parameters that would usually be signaled by the LTE RRC (most notably the ServingCellConfigCommon and the ServingCellConfig configuration messages) are made available through configuration files to both the gNB and the UE.

Unless otherwise noted all the described features apply to gNB and UE.

### A. Supported Hardware Targets

The most popular hardware target used with OpenAirInterface are the USRP devices from ETTUS research. The two devices that are compatible with the larger bandwidth of NR are the X3x0 and the N3x0. The X3x0 provides bandwidths of up to 160MHz, but the maximum usable bandwidth for NR is only 80MHz (217 PRB; which is due to the master clock rate of 184.32Msps and requires to use 3/4 sampling). The N3x0 supports a master sampling rate of 122.88 Msps and thus supports bandwidths of up to 100MHz (273 PRB).

As an alternative to the USRP, there is also the SDR platform developed by SYRTEM[2], which is based on the Xilinx ZYNQ-7000 ZC706 evaluation kit and an RF daughterboard based on the Analog Devices ADRV9371 chip (the same chipset as the USRP N3x0). It therefore has the same capabilities as the N300, but the communication with the host is over PCI express rather than 10Gbit Ethernet. Also not all parts of the driver are open source and need to be purchased from SYRTEM.

### B. PHY

For the moment, we have focused mainly on the development of the downlink and for frequency range 1 (below 6GHz). The supported subcarrier spacing is 30kHz and the supported bandwidths are 40, 80, and 100MHz (106, 217, and 273 PRB). The following physical channels are supported

- **NR-PSS and NR-SSS.** Initial synchronization procedure at UE.
- **NR-PBCH.** with highly efficient polar encoder and decoder. Receiver validated for single beam and known SSB offset.
- **NR-PDCCH.** Type0-PDCCH common search space set configured by pdcch-ConfigSIB1 in MasterInformationBlock. Highly efficient 3GPP compliant polar encoder and decoder. Supported DCI formats: 00 (UL), 10 (DL)
- **NR-PDSCH.** Highly efficient 3GPP compliant LDPC encoder and decoder (BG1 and BG2 supported). Single symbol DMRS, dmrs-TypeA-Position Pos2, DMRS configuration type 1, PDSCH mapping type A, single layer.

The LDPC decoder has been implemented in software by TCL and takes advantage of the advanced vector extension 2 (AVX2) SIMD instruction set of the Intel x86 architecture [9]. It has an average latency of 700$\mu$s for 8 segments of 8448 bits each (largest segment size in NR) on a single core[3] providing an average throughput of 96.5Mbps. More than 8 segments can be parralelized over multiple cores. Taking into account all the overheads from PHY and MAC and we managed to achieve a throughput of 150Mbps end-to-end on the real-time testbench. To achieve higher throughputs, the LDPC decoder can also be offloaded to an FPGA board. Such a board has been developed by SYRTEM and with it we could achieve a throughput of 300Mbps.

The polar decoder has been implemented by Eurecom using a standard successive-cancellation (SC) list decoder algorithm described in Section II.A of [10] again taking advantage of AVX2 instruction set. This algorithm is not optimal in BLER performance but it is extremely fast. The decoding time for, e.g., the PBCH payload of 32bits is less than 4 $\mu$s. In the future we might improve the algorithm based on [11].

### C. MAC

The MAC receives the configuration from higher layers and further configures the PHY using the FAPI P5 interface. At the moment the gNB still uses a dummy scheduler with a single pre-configured UE and statically allocated resources. Using the FAPI P7 interface it can generate all the parameters necessary for the physical channels mentioned above and can transmit random data on the PDSCH.

At the UE the MAC also stimulates the PHY via the FAPI interface and configures the receivers for all the physical channels mentioned above.

### D. RRC

The RRC receives the configuration from the main gNB application and passes down the configuration for the MAC and the PHY. It also handles the encoding and decoding of all the RRC messages. For the EN-DC architecture that is used at the moment this is limited to only the MIB.

### E. gNB Software Architecture

The software architecture for the gNB has been completely revised to meet the more stringent requirements of 5G-NR. We have introduced a pipeline structure that distributes the processing of different blocks in the processing chain over different threads. Additionally some computational expensive tasks, like the FFTs of the front end processor or the LDPC encoder can be further parallelized using worker threads. The architecture can be adapted to the number of cores available on the system. More details are described in [12].

[1] https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/5g-nr-development-and-releases

[2] http://www.syrtem.com

[3] Tested with "ldpctest -l8448 -s10 -n1000 -S8" on an Intel Xeon Gold 6154 CPU 3.00GHz.

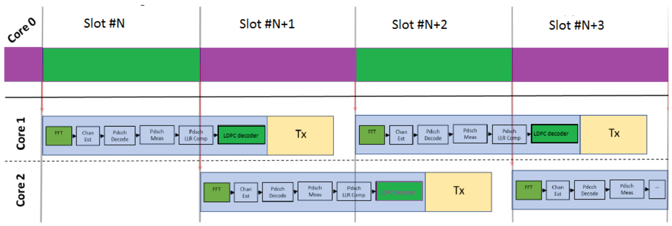


Fig. 5. UE threading structure

.

### F. UE Software Architecture

Even though many computational expensive modules have been already optimized, it is still hard to perform all process within one slot period. Since there is no dependency between slots, it is possible to parallelize them over different parallel threads.

In our current implementation, one thread is dedicated to the reading and writing samples to and from the SDR, and one or more worker threads are responsible of processing the downlink signal of slot $N$ and prepare the uplink signal of slot $N+4$, where $N = t \mod T$, $t$ is the current thread and $T$ is the total number of threads. The number of threads is configurable to adapt the different requirement in 5G standard and different hardware execution time. An example for $T = 2$ threads is shown in Figure 5: the processing time allowed for each slot is doubled from 1 slot to 2 slots.

## V. Demo and interoperability tests

In the demo we show both OAI gNB and OAI UE both running together with an USRP N310 in real-time. The gNB has a static configuration for one UE and will generate all the DL channels. The UE will synchronize to the gNB and will decode all the channels in real-time. The gNB can also be connected to a signal analyzer such as the Rohde & Schwarz FSW, if available. This was done in a lab setup in collaboration with Orange and the standard compliance of all the channels was already checked. A video of the demo can be found at https://www.youtube.com/watch?v=EHp87qkzf5k. At WSA we will show increased downlink throughput and some video transmission.

## VI. Roadmap

The next milestone is planned for mid 2019, where we would like to have full uplink and downlink capabilities of gNB and UE such that we can pass traffic between them. For the end of 2019 we plan to have our gNB integrated with an updated LTE eNB and all the EN-DC procedures should be operational such that we can establish a connection with a commercial UE, which should be availiable by then.

### A. PHY

For the gNB we plan to support

- Subcarrier spacing 60 and 120kHz
- SSB: Make flexible and configurable, Support multiple beams
- PDCCH: Configure common search space from FAPI (currently using MIB parameters)
- PDSCH: Support for up to 4 layers, Integration of hybrid beamforming algorithm, Control of active antenna array
- PRACH receiver
- PUSCH receiver
- PUCCH receiver
- SRS receiver

For the UE we plan to support

- subcarrier spacing 60 and 120kHz
- Initial sync: Improve performance, Detect multiple SSB, SSB frequency search, Detect and compensate frequency offsets
- Dual layer receiver

### B. Protocol stack

For the gNB we plan to support

- Random access procedures
- HARQ procedures
- Real MAC scheduler (DL and UL) + interface to RLC
- update PDCP/RLC to 5G
- F1 interface for CU/DU split
- Xn interface to master eNB

For the UE we plan to support

- Finish FAPI interface and procedures
- Interface with higher layers (RLC/PDCP)

## VII. Conclusions

This paper provides some background information on the implementation of 5G-NR within the OpenAirInterface project. Of course this is only the first milestone of the project showing a first version of the gNB and the UE running downlink only. However, it shows the feasibility of running 5G-NR on a software radio platform using highly optimized LDPC and Polar decoders. By summer 2019 we are planning to have a full DL/UL running and by the end of 2019 we hope to be able to show the first interoperability with commercial UEs, which should be available on the market by then.

OpenAirInterface is freely available and completely open source and available on our gitlab server at https://gitlab.eurecom.fr/oai/openairinterface5g. It is a collaborative project and external contributions are welcome. Please get in touch with one of the authors, if you would like to do so.

## Acknowledgements

OpenAirInterface is a community developed open source project and the work described in this paper is the result of many contributors, such as Raymond Knopp, Guy de Souza, Elena Lukashova, Tsu-Han Wang, Hongzhi Wang, Sebastian Wagner, Jacques Tricoire, Augustin Mico Pereperez. The work described in this paper has been partially supported by the H2020 project 5GENESIS (5genesis.eu).

# REFERENCES


[1] B. Bertenyi, S. Nagata, H. Kooropaty, X. Zhou, W. Chen, Y. Kim, X. Dai, , and X. Xu, "5G NR radio interface," *Journal of ICT Standardization*, vol. 6, no. 1, pp. 31–58, 2018. [Online]. Available: https://doi.org/10.13052/jicts2245-800X.613

[2] B. Bertenyi, R. Burbidge, G. Masini, S. Sirotkin, and Y. Gao, "NG radio access network (NG-RAN)," *Journal of ICT Standardization*, vol. 6, no. 1, pp. 31–58, 2018. [Online]. Available: https://doi.org/10.13052/jicts2245-800X.614

[3] Amarisoft, "Amarisoft 5g software suite release announcement," Press Release, Feb. 2019. [Online]. Available: https://www.amarisoft.com/app/uploads/2019/02/Press_release_Amarisoft_20190209_5G_MWC19.pdf

[4] OpenAirInterface Software Alliance, "Oai public license v1.1," Aug. 2017. [Online]. Available: https://openairinterface.org/docs/OAI%20Public%20License%20-%20V1.1.pdf

[5] Nokia Networks, "5G core," Jan. 2019. [Online]. Available: https://networks.nokia.com/5g/5g-core

[6] R. Knopp and K.-H. Hsu, "5G new radio user equipment FAPI-like interface," Eurecom, Tech. Rep., 2018. [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/5g-nr-development-and-releases

[7] Small Cell Forum, "FAPI and nFAPI specifications," Small Cell Forum, Tech. Rep. 082.09.05, May 2017.

[8] R. Knopp, G. D. Souza, and W.-T. Chen, "OAI software architecture for dual connectivity in E-UTRA and 5G-NR and nFAPI for MAC/PHY interface," Eurecom, Tech. Rep., 2018. [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/5g-nr-development-and-releases

[9] S. Wagner, "NR LDPC decoder," TCL, Tech. Rep., Mar. 2018. [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/raw/develop-nr/openair1/PHY/CODING/nrLDPC_decoder/doc/nrLDPC/nrLDPC.pdf

[10] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast list decoders for polar codes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 318–328, Feb 2016.

[11] Y. Shen, C. Zhang, J. Yang, S. Zhang, and X. You, "Low-latency software successive cancellation list polar decoder using stage-located copy," in *2016 IEEE International Conference on Digital Signal Processing (DSP)*, Oct 2016, pp. 84–88.

[12] T.-H. Wang and R. Knopp, "OpenAirInterface: A pipeline structure for 5G," in *DSP 2018, IEEE International Conference on Digital Signal Processing, 19-21 November 2018, Shanghai, China*, Shanghai, CHINA, 11 2018. [Online]. Available: http://www.eurecom.fr/publication/5733