# SDR and NFV extensions in the ns-3 LTE module for 5G rapid prototyping

Marco Miozzo, Nikolaos Bartzoudis, Manuel Requena, Oriol Font-Bach, Pavel Harbanau, David López-Bueno,
Miquel Payaró, Josep Mangues
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Spain
{mmiozzo, nbartzoudis, mrequena, ofont, pharbanau, dlopez, mpayaro, jmangues}@cttc.es

*Abstract*— **The virtualization of mobile network functions constitutes one of the main blocks for addressing the high flexibility requirements of fifth generation (5G) communication systems. Reconfigurable hotspots are expected to be massively deployed to enable on-demand services and dynamically adapt the network capacity according to traffic requirements. In this paper, we present the extensions and modifications of the long term evolution (LTE) module of the ns-3 simulator (LENA) to include a software defined radio (SDR) physical layer implementation. These extensions combine the native flexibility of the simulator with the SDR features of a real-time prototype. Moreover, the framework was designed to distribute the communication functions across different elements of the network with the possibility of adjusting several transmission parameters as in a network function virtualization (NFV) paradigm. Thanks to an emulated full network protocol stack, the prototype allows the experimentation of novel 5G solutions and the evaluation of relevant key performance indicators (KPIs) from the lower layer protocols up to application level. To this aim, we present the experimental evaluation of the KPIs of energy, latency, throughput and reconfiguration time in relevant scenarios.**

***Keywords—5G prototyping, SDR, SDN, NFV, testbed***

## I. INTRODUCTION

The fifth generation (5G) of mobile communications will need to support a wide range of services, including enhanced mobile broadband, ultra-reliable low latency, and massive machine-type communications. According to this, flexibility will play a crucial role in fulfilling the relevant 5G key performance indicators (KPIs). Examples of such KPIs are: 1000x increase in area capacity, reduction of service creation time from hours to minutes, zero perceived downtime, or 90% energy consumption reduction. Due to the flexibility requirements and the increasing capacity of general-purpose processing units, software is acquiring a more prominent role in network architecture design and deployment. To this respect, Software Defined Radio (SDR), Software Defined Networking (SDN) and Network Functions Virtualization (NFV) have been recognized as main building blocks for 5G not just for flexibility, but also for energy efficiency and network programmability.

As for the mobile network architecture, a variety of functional splits have been proposed and deployed in recent years. For instance, Cloud Radio Access Network (CRAN) developments [1][2] have been quite common; in CRAN only some physical layer processing is left next to the antenna and signal samples are carried to data centers for processing. Other options with less stringent data rate and timing requirements have also been proposed (e.g., [3]), providing a new type of virtualization by splitting MAC and PHY layers. More recently, various splits at PHY, MAC, RLC and PDCP layers have been considered for relaxing the stringent requirements of CRAN while maintaining the benefits of centralized processing [4]. Such MAC-PHY splits imply structural modifications and extensions to existing research tools in order to enable or accommodate SDN/NFV features. For instance, prototypes for 5G should enable the reconfiguration of the network according to the actual system conditions.

In this paper, we present a novel work to combine the flexibility of the well-known and widespread open source ns-3 LENA long term evolution (LTE) simulator/emulator [5] with a real-time field programmable gate array (FPGA) implementation of the PHY-layer [6]. Towards this end, we implemented an interface that enables the real-time communication between the FPGA and LENA modules, which in turn allows the FPGA to implement the channel resource allocation defined by LENA. Moreover, we enabled the emulation of different flexible functional splits for moving the communication functions across different network nodes with processing capabilities for mimicking the virtual small cells (vSC) paradigm [4]. This approach allows maintaining the typical advantages of a simulator (e.g., scalability, replicability, flexibility and low computational complexity), and simultaneously move toward a rapid prototyping approach for 5G networks. In fact, the SDR PHY layer extension for LENA enables the real-time over-the-air transmission of actual RF signals, contemplating likewise real-world wireless channels. Thus, real-time hardware-based experiments can be conducted to more closely represent the behavior of real-world wireless networks. This will allow to realistically evaluate the performance of next generation 5G wireless networks, through the analysis of KPIs not only in an end-to-end basis, but also when focusing on the individual modules. Moreover, the integration of the underlying software (SW) and hardware accelerated (HWA) modules was designed to allow validating scenarios with different system bandwidths (BWs), modulation and coding schemes (MCSs), resource block (RB) loads, transmitter output power levels, waveforms (e.g., LTE vs. 5G candidates), antenna schemes and transmitted output power levels. Thanks to this design, the framework maintains the typical scalability of simulators and can be used to emulate new solutions in wide scenarios with deterministic environmental conditions, while allowing to evaluate realistic over-the-air transmissions. This twofold nature represents an important and unique feature since it enables moving rapidly from the preliminary simulation-based analysis to the proof of concept with prototypes.

## II. SYSTEM ARCHITECTURE

### A. *Overall System Architecture*

The main blocks comprising the architecture of the system are based on the SW and HWA parts shown in Fig. 1, which according to the specific setup can target different 5G virtualization configurations. Considering that the 5G standardization is under definition, we utilized the 4G LTE technology, since the basic architectural concepts still apply. Thanks to the flexibility of the implementation, any extensions required to include new 5G techniques can be contemplated in the future. The stack is composed by a SW part based on the LENA open source network simulator and emulator. LENA has been mainly developed at CTTC, which nowadays maintains the module in the official ns-3 distribution. LENA has been designed to simulate a full LTE protocol stack, including the evolved nodeBs (eNBs), the user equipment (UE) and the evolved packet core (EPC). The HWA includes the LTE-based downlink (DL) physical-layer (L1) developed at CTTC [7] that targets FPGA-based system-on-chip (SoC) devices. As it will be detailed in this section, the previous two building blocks were significantly modified and extended in order to be integrated in a single platform. The SDR-based LENA testbed guarantees that computationally-intensive real-time functions can run without performance degradation, allowing to validate 5G KPIs at run-time.

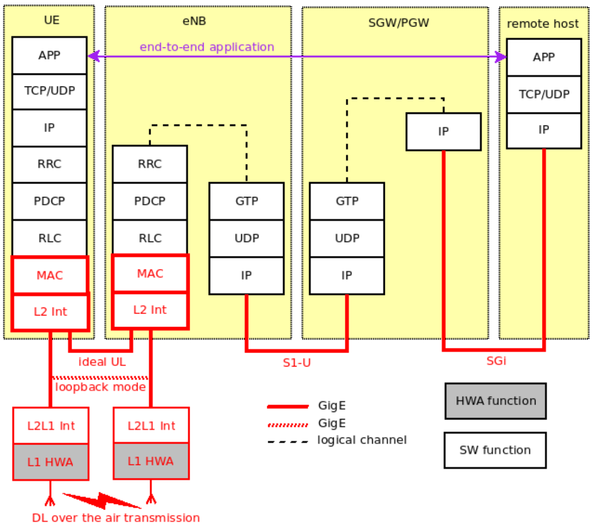


Fig. 1. Overview of the system.

The design of the system took particular attention in providing a high level of flexibility with respect to function partitioning. HWA and SW functions can be placed and executed in different processing elements of the network, which may be seen as an enabler for NFV, and each of the software building blocks, as a virtual network function (VNF). Currently, LENA when combined with the SDR and NFV extensions allows the emulation of three function splits (FS). The first one is the classical CRAN, where the entire eNB protocol stack is placed in the Cloud together with the EPC; the eNB site hosts only the remote radio head (RRH). In the second one, the L1 is placed locally at the eNB site, whereas the higher layers starting from upper MAC are in the Cloud, as in the split MAC architecture [4]. Finally, the third FS is similar to the previous one, except for the higher layers functions of the eNB are placed in a processing node close to the L1, which is inspired by the multi-access edge computing (MEC) approach.

Thanks to the HWA and SW blocks, the system supports most of the LTE functionalities and is therefore able to emulate the whole end-to-end LTE network. Table I provides the list of the main features. At the time of writing, the system only supports over-the-air transmissions in the DL. The UL is under development and therefore the L1 UL is bypassed, which translates in having an ideal error-free UL channel.

### B. *L2 and above*

L2 and above protocols rely on the SW implementation based on the LENA module. LENA has been originally designed as a simulator. Therefore, it does not include a complete L1 implementation, but relies on link-to-system techniques to build an abstract model to estimate the channel impact on the higher-layer protocol data units. In order to be integrated with the HWA system, new features have been introduced. The main one is implementing the functionalities of the L2-L1 interface that enable the interconnection between LENA and the HWA L1, which in turn interfaces with an SDR-based sub-6 GHz RF front-end. With this setup, a fully real-time, end-to-end, over-the-air DL communication link can be utilized instead of relying on an emulated wireless channel. In detail, two ns-3 functionalities have been integrated. Since LENA is natively working in simulation mode that uses an internal simulated clock, the *RealTime* ns-3 event scheduler is used to synchronize the protocols with the real local clock of the hardware in which the process is running. This allows generating packets in real-time. Additionally, modifications were made to allow the interaction with external SW-HWA modules. The simulator maintains all the data within its simulated scenario, i.e., inside the simulation process. By using the *FileDescriptor NetDevice* functionality of ns-3, LENA can exchange real IP packets with external HWA and SW components. Thanks to these features, LENA is able to split different network functionalities and make them interact. In detail, the function comprising the EPC, the UE and the eNB can be placed in different processes running in different machines. Moreover, LENA can also interact with real applications (e.g., voice and video-streaming clients) and with real hardware, as done with the L1 HWA.

TABLE I. SYSTEM FEATURES

| LTE Protocol | Properties |
|---|---|
| eNB PHY | • DL transmission with 20 MHz BW in FDD mode<br>• LTE-compliant channel coding<br>• Data channels (PDSCH, PUSCH) and a simplified control channel (PDCCH)<br>• Achievable data rates up to 75 Mbps<br>• Reference symbols: CRS, UERS, PSS, SRS<br>• Reusable FPGA implementation |
| UE PHY | • IQ and DC offset correction<br>• Automatic gain control (AGC)<br>• Time and frequency synchronization<br>• Cell-specific and UE-specific channel estimation<br>• Channel equalization of data symbols<br>• Turbo decoding (data channels) and Viterbi decoding (control channel)<br>• Reusable FPGA implementation |
| MAC (Medium Access Control) | • Error correction through CRC<br>• Logical Channel multiplexing<br>• Dynamic resource scheduling with Round Robin<br>• Random Access Procedure |
| RLC (Radio Link Control) | • Fragmentation, Concatenation and packet reordering<br>• Transmission modes: Transparent mode (TM), Unacknowledged mode (UM), Acknowledge mode (AM) |
| PDCP (Packet Data Convergence Protocol) | • Real headers following 3GPP specs<br>• Maintenance of PDCP sequence numbers (SN)<br>• Transfer of SN status (for handover) |
| RRC (Radio Resource Control) | • Connection management (establish and reconfiguration)<br>• System information (MIB, SIB, etc.) |
| NAS (Non-access Stratum) | • Focus on NAS Active state<br>• UE Attachment<br>• Evolved Packet System (EPS) Bearer activation<br>• Multiplexing of data onto active EPS Bearers (based on Traffic Flow Templates) |
| EPC (Evolved Packet Core) | • S1-U and S1-C (user data and control plane) realistic model including GTP-U<br>• X2-U over GTP/UDP/IP packets<br>• X2-C over UDP packets (no standard encoding)<br>• S11 interface abstract model (no GTP-C PDUs exchanged) |

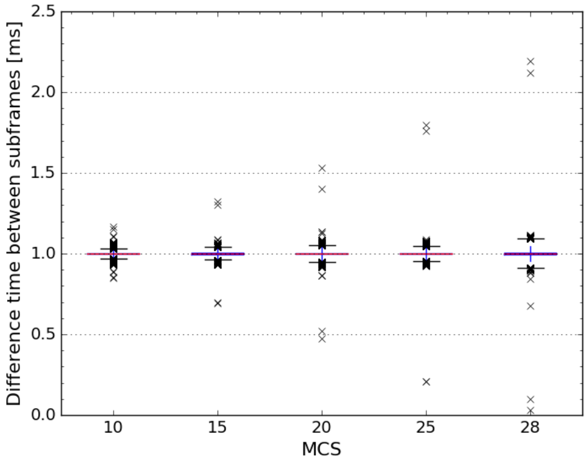


Fig. 2. Difference time between consecutive L2-L1 frames generated by the SW.

The L2-L1 interface has been implemented starting from the scheduler application programming interface (API) of LENA, which is based on the standard defined by the Small Cell Forum, to mimic a MAC split virtualization [4]. Thanks to this API, the common control messages have been serialized in time-stamped frames to be exchanged with the L1. For instance, the DL control information (DCI) allows managing the MCS and RB allocation profile of the UEs on a millisecond basis, or the transmission time interval (TTI) in LTE nomenclature. The original LTE L1 modules of LENA for both the eNB and UE have been replaced by new ones in charge of encapsulating the control and data plane in each TTI to be transmitted through the L2-L1 interface ("L2 Int" in Fig. 1). The control plane includes the main API primitives for allowing the DL communications. The data plane serializes the transport blocks (TBs) per logical channel (LC) basis enabling their multiplexing and includes the error detection through cyclic redundancy error check (CRC). The CRC has been adopted since HARQ is not implemented at this stage. In LTE, one subframe is generated every millisecond and the MAC and PHY operations are conditioned by this stringent requirement. Fig. 2 shows the jitter between subframes generated at the MAC layer for a 20 MHz BW configuration. As it can be seen in the figure, this difference is smaller than 100 microseconds for most of the subframes (i.e. lower and upper quartiles). In the boxplots of the figure, lower and upper quartiles are at 1 % and 99 %, respectively. Only a few subframes have a bigger difference when the MCS is increased. This dispersion is compensated in the PHY layer by buffering a few subframes. For smaller BWs, there is no such outlier.

At the MAC layer, the Round Robin (RR) scheduler has also been updated to include the constraints of the HWA L1. In fact, natively, LENA does not consider any limitation in terms of the number of UEs that can be allocated simultaneously due to physical DL control channel (PDCCH) resource limitations. This requires extending the RR scheduler to allow the simultaneous transmission of a limited number of UEs as a function of the specific BW configuration. Similarly, the amount of data that can be transmitted in subframe 0 has been reduced in order to enable the L1 to fit the physical broadcast channel (PBCH) within the PDSCH.

A sketch of the main elements modified in the LENA simulator is depicted in **Error! Reference source not found.** highlighting the modules and the logical connections that have been extended in bold red lines. For debugging purposes, the loopback mode (highlighted with a red dashed line) allows emulating the network without the HWA L1 (i.e., the L1 will be a cable connection, which translates to an ideal error-free channel). It has to be underlined that, all these extensions have been implemented in a transparent mode with respect to LENA, which enables working in two different operative modes: the standard simulation and the SDR emulation.

### C. Real-time FPGA-based PHY-layer (DL)

The digital signal processing (DSP) blocks of the L1 were implemented as real-time FPGA-based HWA functions ("L1

HWA" in Fig. 1). The register transfer level (RTL) design made use of Xilinx intellectual property cores (i.e., precompiled synthesizable DSP functions) which along with custom-designed DSP blocks and control units ensure a flexible operation of the logic. A concise representation of the L1 processing blocks is shown in Fig. 4. The HWA L1 can be adapted in a subframe basis to the requirements of the L2, the configured FS and the selected BW. The L2 defines the configuration of the different blocks comprising the PHY-layer according to the instantaneous operative requirements of the eNB (e.g., channel coding parameters). The control information propagates from the L2 via a processing system (PS) embedded in the FPGA device; a series of custom L2-L1 interfacing frames are generated in the PS. The HWA L1 disposes a central state machine that parses this information and places the user data in the DL shared channel (DLSCH), programs the required parameters for the turbo encoding stage, generates the contents of the PDCCH and the PBCH and finally programs the parameters of the convolutional encoding stage. This state machine is also responsible for handling missing or wrongly decoded control information (e.g., errors in the L2-L1 communication). In that case, the eNB discards incoming data until receiving valid one from the L2 (i.e., new frame generation starting from subframe 0). The error occurrences are signaled to an embedded memory buffer, which resides in the HWA part of the L2-L1 interface; the latter issues an interrupt to the PS guaranteeing that exceptions will be appropriately handled from all the involved building blocks. A second state machine allocates the contents to each resource element (RE) in the DL signal, which is then used as an input to the inverse fast Fourier transform (iFFT) and the cyclic prefix (CP) insertion DSP blocks. The result of different L2-L1 communication errors is also handled from this state machine; if for example the required DLSCH control is not available when needed, a request to the central state machine halts the eNB transmission.

## III. L2-L1 INTERFACE

The L2-L1 interface enables the time-constrained interaction between the SW and HWA modules ("L2L1 Int" in Fig. 1). It has been designed to be modular facilitating the interconnectivity and virtualization of the system. The SW and the HWA can be treated as separate entities that communicate in this case via UDP, guaranteeing a real-time and over-the-air communication between the eNB and UE PHY layers. The real-time L2-L1 interface is placed in the ARM-based PS of the Xilinx Zynq FPGA. In order to satisfy the stringent requirements of latency, it is executed on a customized Linux distribution with a fully preemptive kernel, which enables real-time tasks. Apart from this, different queuing and buffering techniques have been applied. This ensures that the strict 1 millisecond timing requirement for the combined low MAC/PHY operations is met, including the exchange of control and user plane data between the two layers. An example of the latency measured over 1 minute in the UDP communication between a laptop hosting the SW part of the system and the board hosting the HWA part (via an Ethernet switch) is reported in Fig. 3. The variance in latency is caused by the network connection and the CPU processing of incoming packets. The L2-L1 interfacing frames are designed with a custom format based on the Small Cell Forum API, which has been adapted to efficiently exchange the parameters of both control and data plane. Table II provides the list of the main parameters exchanged for the DL transmission.

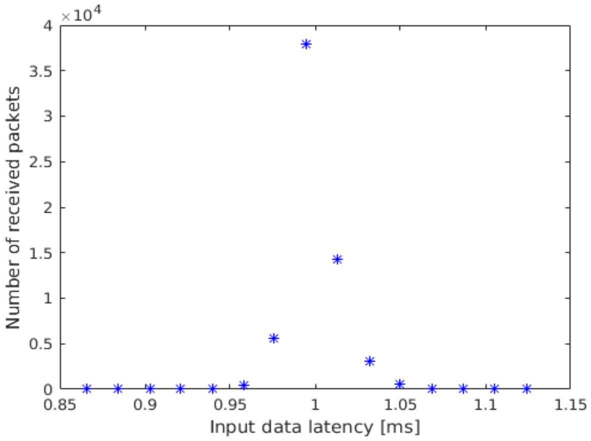


Fig. 3. Input data latency for BW=20MHz, MCS=10 system configuration

TABLE II. L2-L1 INTERFACE DL MAIN PARAMETERS

| Frame | Description |
|---|---|
| DL-DCI | • Radio Network Temporary Identifier (RNTI)<br>• MCS<br>• TB size<br>• RB bitmask |
| UL-DCI | • RNTI<br>• MCS<br>• TB size<br>• RB start<br>• RB length |
| MIB (Master Information Block) | • DL BW<br>• System Frame Number |
| SIB1 (System Information Block) | • Cell identity<br>• Public Land Mobile Network (PLMN) identifier |
| RAR (Random Access Response) | • Random Access RNTI<br>• RAR Num<br>• Random Access Procedure (RAP) id.<br>• RAR UL grant |

## IV. HARDWARE SETUP

In this section, we present the hardware boards and equipment that have been used to validate the SDR extension of LENA. The L1 HWA processing blocks and the L2-L1 SW interface of the eNB prototype are hosted in the Xilinx ZC706 board, which features the Xilinx Zynq XC7Z045 SoC device. The Analog Devices AD-FMCOMMS3 RF transceiver board is plugged to the Xilinx ZC706 board. A suitable power amplifier (PA) unit, RF band-pass filter and antenna were also interfaced with the AD-FMCOMMS3 board. A Linux kernel space application that runs at the PS side of the Zynq XC7Z045 device is used to tune and program the AD9361 RF transceiver IC (RFIC) at the AD-FMCOMMS3 board. The

Xilinx ZC706 board is interfaced with CTTC's EXTREME Testbed [8] using a GigE connection. EXTREME comprises generic purpose servers configured either as a datacenter or distributed throughout the network and hosts LENA's SW-based eNB and UE stack (i.e., L2 and above), as well as the EPC. The setup can also include a real-time multi-channel emulator (Elektrobit Propsim C8) able to realistically reproduce the effect of standard or custom designed mobile channels. A diagram of the overall hardware setup is provided in Fig. 4.

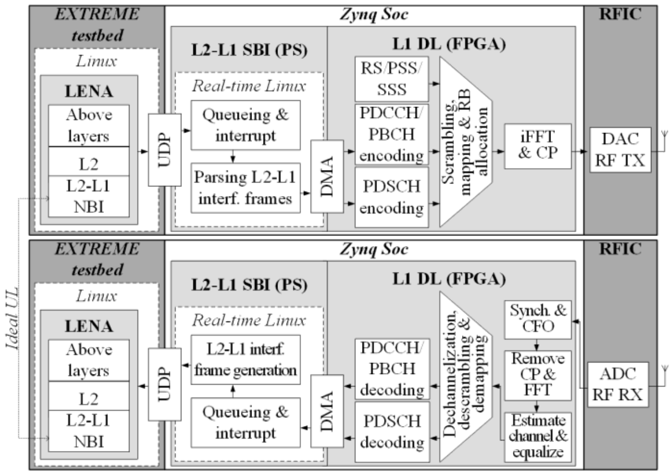


Fig. 4. Overall HW diagram and RTL design of the HWA L1 DL system.

## V. TESTING AND VALIDATION

### A. Energy efficiency KPI

This KPI was assessed by measuring the energy footprint of the following eNB ICs: the Xilinx Zynq SoC baseband processor (XC7Z045 device), the AD9361 RFIC and the Ethernet PHY IC. Fig. 5 a) shows the impact of BW scaling on the power consumption of the Xilinx Zynq SoC device. As it can be seen downscaling the signal BW can provide important power savings, around 40% when changing from 20 Mhz to 1.4 MHz, which are stemming from the reduction of the digital circuit activity (L1). Similarly, Fig. 5 b) shows the consumption when applying a different FS in which the baseband processor at the eNB site does not host the L1 any more (i.e., classic CRAN FS configuration). Hence, by comparing Fig. 5 a) and Fig. 5 b) we can observe the notable power savings that can be attained by adopting a different FS. More relevant results can be found in [6].

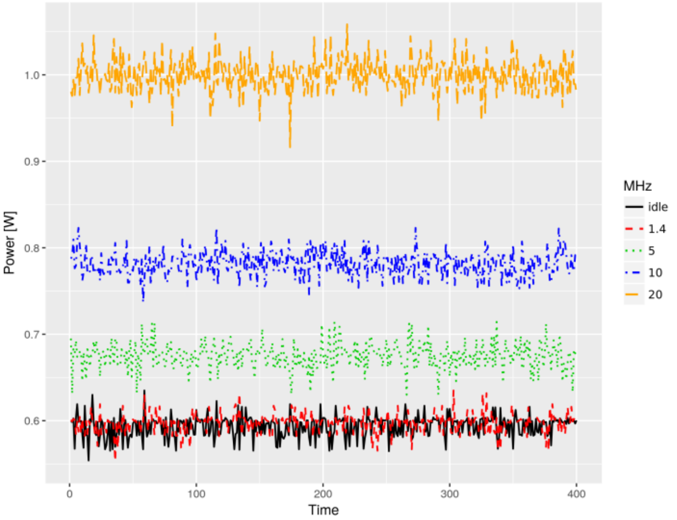
### B. Latency KPI

The latency has been calculated for both L1 and L2-L1 interface for all four supported BW configurations assuming the FSs where the L1 is placed in the eNB site. The total latency for the baseband processing and L2-L1 interface is given by:

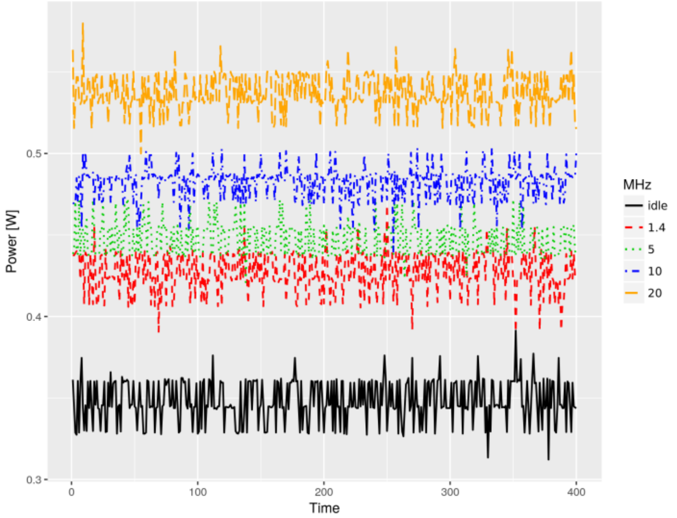
$$LAT_{TOTAL} = L1_{LAT\ (BB\ processing)} + L2L1_{LAT\ (interface)} \qquad (1)$$

The L2-L1 interface latency has a fixed value (i.e., ring buffer solution described in section III), hence $L2L1_{LAT} = 10$ milliseconds. The $L1_{LAT}$ is equal to the initialization time ($A_{LAT}$) plus the processing time counting from the first L2-input packet to the first output sample of the L1 inserted to the digital analog converter (DAC) ($B_{LAT}$); the calculated latencies are based on deterministic measurements of the L1 digital design:

$$L1_{LAT} = A_{LAT} + B_{LAT} \qquad (2)$$



5a. Power consumption of the eNB baseband processor featuring the L1.



5b. Power consumption of the eNB baseband processor without the L1.

Fig. 5. The impact of BW scaling on the power consumption of the Xilinx Zynq SoC is shown in both subfigures for two different FSs.

$B_{LAT}$ considers the encoding of the most demanding UE data sequence (i.e., up to 25 UEs, MCS up to 26, DL BW of 20 MHz) for the HWA L1 design (i.e., deterministic latency of the RTL architecture). Taking into account (1) and (2), the $LAT_{TOTAL}$ for each of the supported BW configurations was calculated as follows:

$$LAT_{TOTAL} = L1_{LAT\ (BB\ processing)} + L2L1_{LAT\ (interface)} = A_{LAT} + B_{LAT} + L2L1_{LAT\ (interface)},\ \text{thus:}$$

$$LAT_{TOTAL(1.4MHz)} = 2.74 + 1.62 + 10 = 14.36\ \text{ms}$$

$$LAT_{TOTAL(5MHz)} = 1.37 + 1.37 + 10 = 12.74\ \text{ms}$$

$$\mathrm{LAT}_{\mathrm{TOTAL(10MHz)}} = 0.68 + 1.36 + 10 = 12.04\ \mathrm{ms}$$

$$\mathrm{LAT}_{\mathrm{TOTAL(20MHz)}} = 0.34 + 1.3618 + 10 = 11.7018\ \mathrm{ms}$$

It should be noted that the end-to-end latency of the system could also be calculated for other specific operating scenarios and system configurations.

### C. *Throughput KPI*

The maximum LTE throughput that LENA is able to generate in simulation mode is of 68 Mbps (20 MHz) with ideal DL and UL channels and without considering the HWA L1. The same maximum achievable throughput was also verified through a series of tests for different MCSs when LENA's scheduler was configured in emulation mode, as depicted in **Error! Reference source not found.**. Therefore, emulation mode does not represent a bottleneck, as the maximum theoretical throughput can be achieved, despite the fact that ns-3 has been originally designed to work in simulation mode.

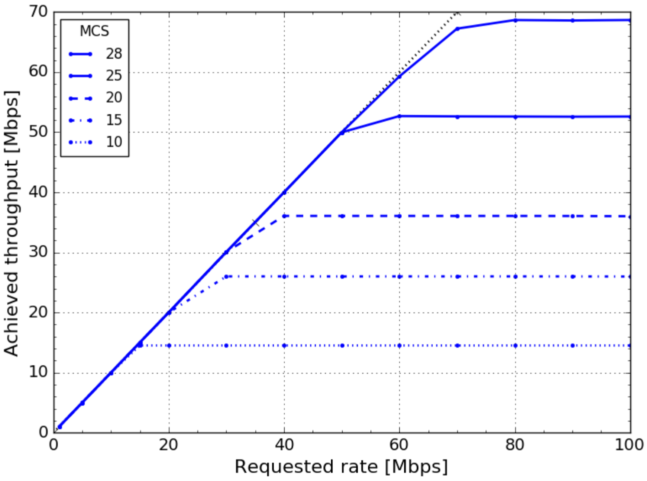


Fig. 6. Achieved throuhgput vs. requested rate with BW = 20 MHz.

### D. *Reconfiguration time KPI*

In this section we calculated the time required to reconfigure the HW components, SW applications and HWA functions for the different FSs when a BW adaptation is applied. The BW adaptation affects the reconfiguration time of different subsystems as follows:

- RFIC (AD9361): up to 1 second, validated according to on-board measurements under different configurations.
- Baseband processor (Xilinx Zynq XC7Z045 SoC): a few ms according to calculations based on RTL simulations.
- LENA: a few hundreds of milliseconds according to estimated calculations that consider a procedure similar to the RRC handshake used to change the transmission mode.

## VI. CONCLUSIONS

In this paper, we have presented an extension to the ns-3 LTE/EPC network simulator/emulator (LENA) that allows working in real-time, by using an SDR physical layer implementation, which aims at combining the flexibility of a simulator with the accuracy of a hardware prototype. In doing this, we designed the system to enable the possibility of distributing the communication functions across different elements of the network as it is made in NFV scenarios. Thanks to an emulated full network protocol stack, the prototype can be used to realistically validate novel 5G solutions and evaluate relevant KPIs from the low-layer protocols up to application level. To this end, we presented the evaluation of the energy, latency, throughput and reconfiguration time KPIs.

## ACKNOWLEDGMENT

This work was supported by the European Commission in the framework of the H2020-ICT-2014-2 project Flex5Gware (Grant agreement no. 671563) and also partially supported by the Generalitat de Catalunya (2014 SGR 1551 and 2014 SGR 1397) and by the Spanish Government under project TEC2014-58341-C4-4-R and TEC2017-88373-R.

## REFERENCES

[1] S. S. Kumar, R. Knopp, N. Nikaein, D. Mishra, B. R. Tamma, A. A. Franklin, K. Kuchi, R. Gupta, FLEXCRAN: Cloud radio access network 42prototype using OpenAirInterface, in: 2017 9th International Conference on Communication Systems and Networks (COMSNETS), 2017, pp. 421–422.

[2] Ghizlane Mountaser, Maria Lema Rosas, Toktam Mahmoodi and Mischa Dohler, "On the feasibility of MAC and PHY split in Cloud RAN", in the Proceeding of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), 19-22 March 2017, San Francisco, USA..

[3] Nicola Michailow, Vincent Kotzsch and Douglas Kim, "Real-time Prototyping of 5G Software Defined Networks: Part 2", IEEE ComSoc White Papers, 23 January 2017.

[4] Virtualization for small cells: Overview, Small Cell Forum White Paper (106.06.01), Tech. rep. (June 2015).

[5] Nicola Baldo, Marco Miozzo, Manuel Requena-Esteso, and Jaume Nin-Guerrero, "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, Miami, FL, USA, pp. 293-298.

[6] N. Bartzoudis et al., "Energy footprint reduction in 5G reconfigurable hotspots via function partitioning and bandwidth adaptation", in the Proceedings of the 5th International Workshop on Cloud Technologies and Energy Efficiency in Mobile Communication Networks (CLEEN 2017), 22 June 2017, Turin, Italy.

[7] O. Font-Bach, N. Bartzoudis, A. Pascual-Iserte, M. Payaró, L. Blanco, D. López, M. Molina, "Interference Management in LTE-based HetNets: a Practical Approach", Transactions on Emerging Telecommunications Technologies (Wiley), Vol. 26, Issue 2, Pages 195-215, February 2015.

[8] R. Muñoz et al., "The CTTC 5G End-to-End Experimental Platform: Integrating Heterogeneous Wireless/Optical Networks, Distributed Cloud, and IoT Devices", in IEEE Vehicular Technology Magazine, vol. 11, no. 1, pp. 50 – 63, March 2016.