Real-time Emulation Methodologies for Centralized Radio Access Networks

Luis Felipe Ariza Vesga
Electric and Electronic Engineering
Universidad Nacional de Colombia
Bogota, Colombia
lfarizav@unal.edu.co

Raymond Knopp
Communication Systems
EURECOM
Biot, France
raymond.knopp@eurecom.fr

Sebastian Eslava Garzon
Electric and Electronic Engineering
Universidad Nacional de Colombia
Bogota, Colombia
jseslavag@unal.edu.co

***Abstract*—This work considers radio-network emulators for testing Centralized Radio Access Network (C-RAN) elements using a frequency-domain functional split (so-called split 7-1). The techniques are applicable to fourth and fifth generation cellular systems. They allow system validation and performance evaluation of base station or Distributed Unit (DU) equipment using a synthetic radio network instead of actual radio units. Also, they can be employed for testing during the development phase in a software-only environment or as real-time traffic stimulus for performance evaluation of not only the physical layer but also layer 2 procedures (e.g. MAC scheduling). Specifically, we propose real-time emulation methodologies built on top of the OpenAirInterface (OAI) platform employing general-purpose processors and fast Ethernet transport ports. We discuss software functions optimizations to emulate the multipath channel, enabling synthetic network scalability through frequency-domain processing and avoiding the need for abstraction models of the physical layer. Even though the IP-based synthetic network can accommodate 1-3 Radio Remote Units (RRUs) and up to 3 User Equipment (UEs) to work in real-time, it supports higher non-real-time scenarios. It can be parallelized on high-end multi-core computers to scale up to higher bandwidth and more UEs. Our proposal exhibits an improvement of one order of magnitude on generic x86 computers related to the average computation time of the multipath channel, compared to the existing time domain approach in downlink and uplink transmissions.**

***Index Terms*—Frequency Domain Analysis, Cloud Radio Access Network, Synthetic Network, Emulation Methodology.**

I. Introduction

There is currently no predefined scientific experimentation standard for development and testing of Centralized Radio Access Networks (C-RANs), in particular concerning validation and performance evaluation methodologies to reduce prototyping uncertainties and to achieve reproducibility, scalability, and applicability properties together [1]. Computer networks and mobile networks research approach these properties in three different ways: Real test beds, network simulations, and network emulations [2].

Except for real test beds which are difficult to scale, network simulations and network emulations are the most suitable means to prototype new mobile broadband protocols, algorithms, architectures, and services for the next generation cellular network. If a network simulator is analyzed, we can run a specific layer of the protocol stack quite accurately to accomplish good scalability and reproducibility, but with applicability issues. However, if a network emulator is chosen, the applicability is increased and we are able to capture the 3rd Generation Partnership Project (3GPP) standard-compliant real-world environments.

As noted, the network emulation is the most powerful option that may use either affordable software with general purpose hardware or expensive advanced physical equipment. Both may include physical antennas for Multiple Input Multiple Output (MIMO) technology, Commercial Off-The-Shelf User Equipment (COTS UEs), and Software-Defined Radio prototyping platforms (e.g. USRP, limeSDR, BladeRF), but prohibitively expensive in a large scale configuration.

If we place emphasis not only on optimizing software functions to emulate the multipath channel, but also simulating the channel model in terms of a frequency domain representation, we are able to decrease the signal processing complexity and run standard-compliant real-time scenarios in a software-only environment that prevents uncertainties of software-defined radio front-ends. It should be mentioned that the techniques considered here can be used to mix both, synthetic real-time network components, and real-time Radio Frequency (RF) hardware to create a hybrid emulation framework where the latter share the synchronization signal to the former. In this case, synthetic real-time network components are adapted to increase the network scalability and real-time RF hardware to test applications.

The paper is organized as follows: Section II explains the C-RAN architecture. Section III describes real-time network emulations in the frequency domain and scalability extensions. Section IV exposes the performance results. In the end we outline conclusions in section V.

II. Centralized Radio Access Network Architecture

This section highlights the C-RAN architecture depicted in Figure 1. It is composed of functional splits summarized in [3] and mapped by Next Generation Fronthaul Interface (NGFI), 3GPP, Small Cell Forum (SCF), and Next Generation Mobile Networks (NGMN) standard organizations. It has three different network segments called the fronthaul, the midhaul and the backhaul.

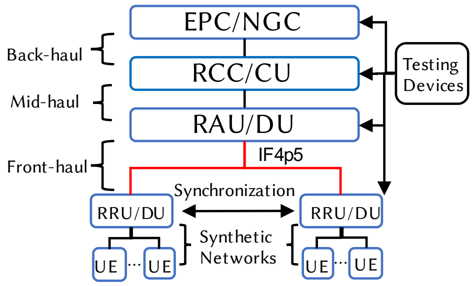

Fig. 1. C-RAN architecture using NGFI / 3GPP terminologies.

Previous efforts on this topic are the following. First, the network slice prototype for C-RAN architecture using the FlexCRAN SDN controller and the IF4P5 functional split to validate the feasibility of handling network slices on-demand [4]. Second, the C-RAN experimental evaluation of capacity and latency based on functional splits and USRPs to enable Remote Radio Units (RRUs) scalability [5]. They do not consider the scalability of real-time synthetic networks in a software-only environment, which is a novel research topic introduced here. It takes advantage of signals exchange in the frequency domain, and software function optimizations of the multipath channel to reduce the signal processing complexity.

A. Functional splits

The functional splits consider a Base-band functionality redistribution into data centers to solve MIMO scalability, performance, and capacity demands of the radio access network. Proposed methodologies are focused on the IF4P5 functional split [6], where the input and the output of the Orthogonal Frequency Division Multiplexing (OFDM) symbol generator transmits and receives compressed resource elements in the frequency domain. In our configuration, the Radio Aggregation Unit (RAU) or equivalently Distributed Unit (DU) functionalities are moved to the Radio Cloud Center (RCC) or equivalently Centralized Unit (CU) entity.

B. Mobile fronthaul, midhaul and backhaul

The C-RAN has 3 different network segments called the fronthaul, the midhaul and the backhaul [7]. The backhaul communicates the RCC to the Core Network within a distance range of 40-200 km, the midhaul connects the RCC to the RAU within a distance range of up to 10 km, and the fronthaul links the RAU with the RRU within a distance range of 0-20 km. We handled backhaul and fronthaul through S1 and IF4P5 interfaces utilizing 1 Gigabit Ethernet ports.

III. Real-time emulation extension in the frequency domain

The following subsections will discuss our extensions to enable real-time (The Transmission Time Interval (TTI) $<$ 1ms) network emulations for the C-RAN in the frequency domain. First, we explain Single Instruction Multiple Data (SIMD) optimizations. Second, we detail the implementation of two Gaussian random number generators to simulate the thermal noise in the receiver hardware, that is time-consuming in OpenAirInterface (OAI). Third, we describe the physical frame structure in the frequency domain. Fourth, we analyze the channel frequency response to reduce the computational complexity of the OFDM chain. Finally, we depict the scenario configuration for synthetic networks scalability and the organization of C-RAN entities.

A. Single Instruction Multiple Data

We took advantage of the processor architecture by selecting Streaming SIMD Extension (SSE) and Advanced Vector Extension (AVX2) instructions to rebuild software functions of the multipath channel model. We chose single-precision floating-point instead of Double-precision floating-point data types to boost 2-fold the network emulation speed.

B. Gaussian random number generators

There are some well-studied implementations such as Wallace, Ziggurat [8] and, Box-Muller [9] which use recursive, rejection, and transformation methods from the best to the worst performance in speed correspondingly.

From above methods, we implemented Ziggurat, SSE Ziggurat, and AVX2 Ziggurat that provide a better performance in terms of the Chi-square metric (This metric indicates how observed data fits expected data) compared to corresponding Box-Muller methods. Afterward, we implemented Box-Muller, SSE Box-Muller, and AVX2 Box-Muller generators that pose a better performance in terms of the average computation time (ns/samples) compared to corresponding Ziggurat methods as shown in Table I. Finally, the fastest AVX2 Box-Muller method was selected.

Generator	Samples	Chi-Square	Average computation time (ns/samples)
Box-Muller		9.99e+05	290
SSE Box-Muller	1e+06	9.99e+05	74.5
AVX2 Box-Muller		9.75e+05	37.4
Ziggurat		1e+06	220
SSE Ziggurat	1e+06	1e+06	78
AVX2 Ziggurat		1e+06	39

TABLE I
Chi-Square and average computation time metrics for Box-Muller and Ziggurat methods.

C. Physical frame structure

We use numerology 0 [10], 15 KHz subcarrier spacing, and 5 MHz of bandwidth. We restrict 5 MHz to reduce the computational complexity and to demonstrate the principle. Figure 2 represents the slot structure for frequency and time domains. The changes we introduced forced us to fix the basic call procedure composed of initialization, receiving system information, random access and RRC connection setup, between UEs and the evolved Node B (eNB) described in [11].

D. Channel Frequency Response

The Orthogonal Frequency-Division Multiple Access (OFDMA) technique, is based on a set of orthogonal subcarriers, some of them allocated to each User End (UE). The

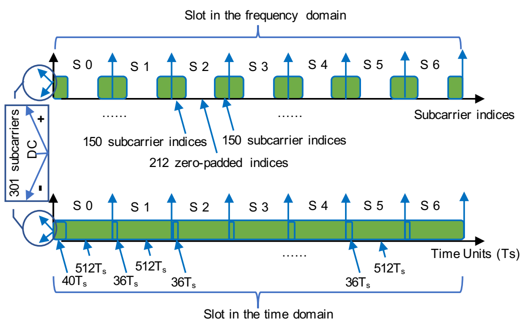

Fig. 2. Slot structure in time and frequency domains (Normal CP, 7 symbols/slot, and 5 MHz of bandwidth). Top, frequency domain. Bottom, time domain.

transmitter chain is composed of modulation, subcarrier mapping, Inverse Fast Fourier Transform (IFFT), adding Cyclic Prefix (CP), Digital to Analog Converter (DAC), and RF (Up-conversion) modules that takes a bits stream to deliver signals in the time domain to the antenna. The receiver chain follows the inverse order to acquire signals from the antenna in the time domain and deliver a bits stream after passing through RF (Down-conversion), ADC (Analog to Digital Converter), removing CP, Fast Fourier Transform (FFT), demapping, and demodulation modules. Essentially bypassing the FFT, IFFT, and CP functions and employing the Channel Frequency Response (CFR), we enabled the frequency domain analysis.

The channel frequency response model extracts details about how signals behave over the path, considering obstacles, noise, and attenuators in a real environment. The equation (1) describes the received signal r(m) in terms of the transmitted signal s(m) convoluted with the Channel Impulse Response (CIR) h(m) and the receiver noise n(m) (m is the number of samples and $*$ the convolution operator).

$$r(m) = s(m) * h(m) + n(m) \tag{1}$$

The CIR is represented by the Tapped Delay Line (TDL) model [12]. It simulates multiple echoes of the same transmitted signal. The model expressed by Equation (2) is divided into two steps. First, it generates the channel state vector **a** that outlines the reduced set of taps of different paths between the transmitter and the receiver, and then Sinc interpolates the channel impulse response.

$$h(m) = \sum_{l=0}^{N_p-1} \mathbf{a}[l] sinc(m - F_s(l+\beta)\Delta\tau_d - 0.5F_s\tau_{max}) \tag{2}$$

In the Equation (2), N_p is the channel path number, F_s represents the sampling frequency, β describes a real number to ensure the continuity of the envelope of $h(m)$, and τ_{max} expresses the maximum allowable delay in the channel.

Applying linearity and shifting properties of the Discrete Fourier Transform to equation (2), we accomplish the equation (3) that belongs to the CFR.

$$H[k] = \sum_{i=0}^{N_p-1} \mathbf{a}[l](j\sin(2\pi\frac{k}{N}m_l) - \cos(2\pi\frac{k}{N}m_l)) \tag{3}$$

Where N is the sampling rate, m_l=$F_s(l+\beta)\Delta\lambda_d$+$0.5F_s\tau_{max}$ and $\Delta\tau_d = \frac{\tau_{max}}{N_p}$.

Equation (3) decreases the computational complexity of convolutions to simple multiplications as noted in equation (4).

$$R(k) = S(k).H(k) + N(k) \tag{4}$$

E. Scenario configuration for synthetic networks scalability

Figure 3 shows an scenario of 2 RRUs, 2 Remote Radio Heads (RRHs), 1 RCC, and 1 Evolved Packet Core (EPC) for an indoor Distributed Antenna System (DAS) with full centralization at the RCC. Each RRU has 1 layer 1 (L1) instance. The RCC has 1 L1 instance and 1 layer 2 (L2) instance with 2 Component Carriers (CCs).

The RCC has two southbound IP-based IF-devices (One per CC). Each RRU has 1 northbound IP-based IF-devices to reach the RCC and one RF-device to reach the RRH.

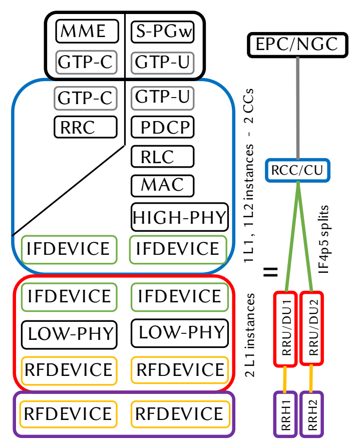

Fig. 3. C-RAN deployment scenario for scalability.

IV. Performance Results

In this section, we present results employing the C-RAN setup for time and frequency domains. We evaluate the performance with the simplest scenario (Composed of 1 EPC, 1 RCC, 1 RRU, and 1-3 UEs) and the scalability increasing the number of RRUs. In a software-only environment network emulation we employ an 8th generation Intel Coffee Lake core I7-8700x6 (Ubuntu 16.04, kernel 4.4.0-133-lowlatency) computer to emulate RRUs and UEs, a 2nd generation Intel core I5-2400x2 (Ubuntu 14.04, kernel 4.4.0-31-generic) computer to emulate the RCC, and a 3rd generation Intel core i7-3700x4 (Ubuntu 16.04, kernel 4.13.0-45-generic) computer to emulate the EPC. In addition, we consider a USRP B200mini-i, two VERT2450 antennas, a Samsung Galaxy S8, and a universal subscriber identity module to evaluate the accuracy of the system-level emulation. The hardware is depicted in Figure

4 and the network emulation parameters of the experiments are summarized in Table II.

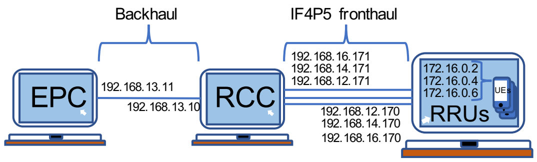

Fig. 4. Hardware. It is composed of 3 PCs: 1 for the EPC, 1 for the RCC, and 1 for RRUs.

Parameter	Single-UE Value / USRP B200mini-i
Band	7
Transmitter gain	90 dB
Receiver gain	120 dB
Transmitter power	15 dBm
Working Mode	FDD
Cyclic Prefix	Normal
Interface compression	A-law
System Bandwidth	5 MHz
Transmission Mode	1 SISO
Multipath Channel Model	AWGN / Rayleigh 1

TABLE II
NETWORK EMULATION PARAMETERS.

The following are the metrics we employed to evaluate the performance, the accuracy of the system-level network emulation, and the scalability.

A. Maximum user throughput

We obtained the maximum user throughput of the Physical Uplink Channel (PUSCH) and the Physical Downlink Channel (PDSCH), stimulating the C-RAN with User Datagram Protocol (UDP) traffic generated from UE and the EPC, and measuring the maximum user success rate respectively (The maximum Modulation and Coding Scheme (MCS) for uplink is 18 and downlink 28). The traffic is generated by the network performance measurement tool called iperf [13] and the maximum user throughput is visualized by the tool called OAI T-tracer.

	MCS	Time	Frequency	USRP B200mini-i	TS 36.213 [14]
Downlink	28	17.5 Mb/s	17.5 Mb/s	17.5 Mb/s	18.336 Mb/s
Uplink	18	8.43 Mb/s	8.43 Mb/s	8.43 Mb/s	9.144 Mb/s

TABLE III
MAXIMUM USER THROUGHPUT (5 MHZ OF BANDWIDTH, AWGN CHANNEL MODEL).

The maximum user throughput values are fulfilled in Table III. All cases use transmission mode 1 and have the same maximum user throughput for downlink and uplink transmissions. Despite the maximum user throughput is reproducible, acceptable errors of 4.56% in downlink and 8.47% in uplink are observed compared to theoretical specifications.

B. Downlink Block Error Rate

The PDSCH Downlink Block Error Rate (BLER) is displayed in Figure 5 for some MCSs. In the frequency domain, we extended the unitary Monte Carlo dlsim simulator to allow the measurement of the BLER. The BLER value is very similar for both domains in all MCS cases, nevertheless, in the frequency domain, the number of erroneous blocks is slightly higher. It is justified because the frequency domain neither implements CP functions nor inhibits the inter-symbol interference.

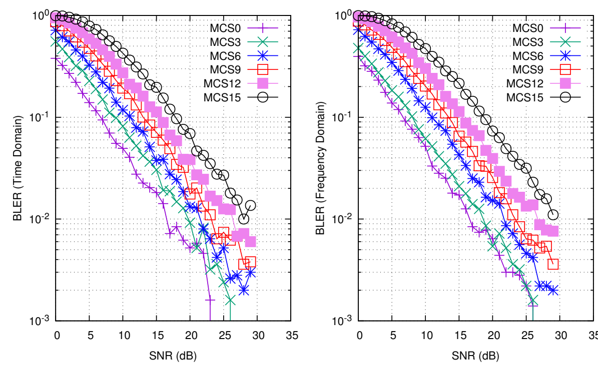

Fig. 5. Downlink Block Error Rate for different MCSs (5 MHz of Bandwidth, 5000 subframes, transmission mode 1, and Rayleigh channel model (1 tap)).

C. Average computation time

The third performance metric is the average computation time of software functions to emulate the multipath channel. We considered the OAI performance profiler among other tools because it marginally impacts OAI emulators in terms of time. Table IV shows the average computation times for some important channel functions. The reported gain is defined as the average computation time in the frequency domain divided by the average computation time in the time domain.

Channel function	Time Domain (μs)	Frequency Domain(μs)	Gain
Downlink multipath channel	45.58	9.774	4.66
Uplink multipath channel	46.048	11.356	4.06
Downlink DAC	19.303	13.815	1.4
Uplink DAC	19.487	13.99	1.39
Downlink receiver rf	500.288	37.062	13.49
Uplink receiver rf	494.876	36.867	13.42
Downlink ADC	18.52	2.304	8.04
Uplink ADC	18.489	2.122	8.71
ADC/DAC (USRP)	0.03255	n/a	n/a
Uplink Channel	596.232	72.758	8.19
Downlink Channel	596.833	78.811	7.57
Uplink PRACH Channel	n/a	219.202	n/a

TABLE IV
AVERAGE COMPUTATION TIMES IN TIME AND FREQUENCY DOMAINS. C-RAN ARCHITECTURE, 5 MHZ OF BANDWIDTH, 10000 FRAMES, AWGN CHANNEL MODEL, AND 5 MB OF IPERF TRAFFIC.

In the time domain, the average computation time of uplink and downlink channels (The channel computation time is the time spent by the ADC, DAC, multipath channel, and RF Gaussian noise functions), takes more than 1 ms which is clearly a non-real-time emulation. However, our proposal in the frequency domain takes near 150 μs, which allows real-time emulations. That is possible because we reduced the computational complexity in the frequency domain considering neither convolutions nor some functions of the OFDM chain. Also, optimizing software functions of the channel model by means of AVX2 single-precision floating-point instructions.

The improvements increased one order of magnitude the average computation time of the multipath channel in the frequency domain compared to the time domain as depicted in Figure 6. Notwithstanding, we required a new uplink physical Random Access channel (PRACH) with a small average computation time of 219 μs, that happens once for each attached UE. However, how much does the frequency domain solution scales?. This question is analyzed in the next subsection.

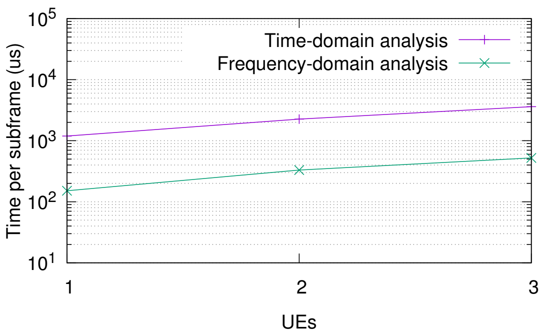

Fig. 6. System-level emulator benchmark of the average computation time.

D. Synthetic network scalability

We extended OAI to support multiple synthetic networks in a software-only environment. Results are shown in Table V. The channel computation time of the USRP B200mini-i is obtained from the manufacturer.

RCCs	RRUs	UEs	Time Domain μs	Frequency Domain μs	B200mini-i μs
		1	1193.065	151.66	0.0651
1	1	2	2262.5	330.972	N/A
		3	3614.066	523.208	N/A
		1	1280.8	162.25	N/A
1	2	2	2215.87	358.2	N/A
		3	N/A	622.164	N/A
		1	N/A	148.072	N/A
1	3	2	N/A	397.556	N/A
		3	N/A	546.334	N/A

TABLE V
AVERAGE COMPUTATION TIMES FOR SYNTHETIC NETWORK SCALABILITY.

In the frequency domain, we were able to run a real-time emulation with UEs and RRUs up to 3. Denser scenarios are possible, but not in real-time. Conversely, in the time domain, the emulation was restricted to less than 3 RRUs.

V. CONCLUSIONS

We successfully implemented affordable and real-time emulation methodologies in the frequency domain for synthetic networks as a prototyping framework to rapid proof-of-concept and time-to-market designs in a software-only environment. It provides a realistic system validation and performance evaluation of real-time protocols at higher layers, assesses protocol correctness and traffic congestion, and creates traffic stimulus for scheduling algorithms applied to the C-RAN.

We reduced one order of magnitude the average computation time of the multipath channel in the frequency domain compared to the time domain. The cost in time we need to pay is 219 μs related to the uplink PRACH channel. This event occurs only during the random access of the basic call procedure, after the initialization and the system information reception. Further optimizations can be achieved with AVX512 instructions and more threads in high-end multi-core computers.

Our proposal in the frequency domain provides a better performance related to the time domain, which makes it a useful network emulation choice to improve the applicability and the scalability for real-time C-RANs. Also, it preserves the same functionality for preparing and processing signals going to and coming from RRUs in a 3GPP standard-compliant round trip network emulation.

REFERENCES

[1] B. B. Romdhanne, "Large-scale network simulation over heterogeneous computing architecture," Ph.D. dissertation, Telecom ParisTech, 46, Rue Barrault - 75634 Paris, France, December 2013.

[2] B. B. Romdhanne, N. Nikaein, R. Knopp, and C. Bonnet, "Openairinterface large-scale wireless emulation platform and methodology," in *Proceedings of the 6th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, ser. PM2HW2N '11. New York, NY, USA: ACM, 2011, pp. 109–112.

[3] C.-Y. Chang, N. Nikaein, R. Knopp, T. Spyropoulos, and S. Kumar, "FlexCRAN: A flexible functional split framework over ethernet fronthaul in Cloud-RAN," in *ICC 2017, IEEE International Conference on Communications, May 21-25, 2017, Paris, France*, Paris, FRANCE, 05 2017. [Online]. Available: http://www.eurecom.fr/publication/5155

[4] S. Costanzo, I. Fajjari, N. Aitsaadi, and R. Langar, "A network slicing prototype for a flexible cloud radio access network," in *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2018, pp. 1–4.

[5] F. Giannone, H. Gupta, K. Kondepu, D. Manicone, A. Franklin, P. Castoldi, and L. Valcarenghi, "Impact of ran virtualization on fronthaul latency budget: An experimental evaluation," in *2017 IEEE Globecom Workshops (GC Wkshps)*, Dec 2017, pp. 1–5.

[6] R. Knopp, N. Nikaein, C. Bonnet, F. Kaltenberger, A. Ksentini, and R. Gupta, "Prototyping of next generation fronthaul interfaces (ngfi) using openairinterface," EURECOM," Technical Report, 2017, white paper. [Online]. Available: http://www.openairinterface.org/?page_id=1695

[7] ITU, "Transport network support of imt-2020/5g," TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU," Technical Report, 2018. [Online]. Available: https://www.itu.int/dms_pub/itu-t/opb/tut/T-TUT-HOME-2018-MSW-E.docx

[8] G. Marsaglia and W. W. Tsang, "The ziggurat method for generating random variables," *Journal of Statistical Software, Articles*, vol. 5, no. 8, pp. 1–7, 2000. [Online]. Available: https://www.jstatsoft.org/v005/i08

[9] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *The Annals of Mathematical Statistics*, vol. 29, no. 2, pp. 610–611, 06 1958. [Online]. Available: https://doi.org/10.1214/aoms/1177706645

[10] 3GPP, "Physical channels and modulation," 3rd Generation Partnertship Project (3GPP), Technical Specificacion (TS) 38.201, 1 2018, version 15.0.0. [Online]. Available: https://goo.gl/HnLFdw

[11] R. Wang, Y. Peng, H. Qu, W. Li, H. Zhao, and B. Wu, "Openairinterface an effective emulation platform for lte and lte-advanced," in *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2014, pp. 127–132.

[12] F. Kaltenberger, "Low-complexity real-time signal processing for wireless communications," Ph.D. dissertation, Vienna University of Technology, Vienna, Austria, May 2007.

[13] J. Dugan, S. Elliott, B. A. Mah, J. Poskanzer, and K. Prabhu. (2019) iperf - the ultimate speed test tool for tcp, udp and sctp. [Online]. Available: https://iperf.fr

[14] 3GPP, "Evolved universal terrestrial radio access (e-utra):physical layer procedures," 3rd Generation Partnertship Project (3GPP), Technical Specificacion (TS) 36.213, 10 2018, version 15.3.0. [Online]. Available: https://goo.gl/C8ePeU