GRANT **AGREEMENT NO.: 732174**

Call: H2020-ICT-2016-2017
Topic: ICT-13-2016
Type of action: RIA

Orchestration and Reconfiguration Control Architecture

D3.5: Final operational real-time SDR platforms

Revision: v.1.0

Work package	WP 3
Task	Task 3.1, 3.2, 3.3
Due date	31/1/2020
Submission date	31/1/2020
Deliverable lead	NI
Version	1.0
Authors	Walter Nitzold (NI), Clemens Felber (NI), Juan Rafael Bolaños Madrigal (NI), Joao Santos (TCD), Yi Zhang (TCD), Wei Liu (IMEC), Xianjun Jiao (IMEC), Muhammad Aslam (IMEC), Jan

	Bauwens (IMEC), Andrea P. Guevara (KUL), Ahmad Nimr (TUD), Martin Danneberg (TUD), Ana Martinez (TUD)
Reviewers	Ingrid Moerman (IMEC), Roberto Bomfin (TUD)

Abstract	This deliverable includes the implementation results obtained in Year 3 on the final operational real-time SDR platforms with focus on the data plane. A mapping of the real-time prototype developments to the showcase scenarios defined in WP2 and detailed explanations of the single contributions including availability in the testbed facilities are described in this final deliverable for WP3.
Keywords	real-time, SDR, data plane, PHY, MAC, higher layer, testbeds

Disclaimer

Acknowledgment

This report has received funding from the EC under the grant agreement 731274.

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable:	R*	
Dissemination Level		
PU	Public, fully open, e.g. web	✓
CI	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to ORCA project and Commission Services	

** R: Document, report (excluding the periodic and final reports)*

EXECUTIVE SUMMARY

ORCA wants to accelerate flexible end-to-end network experimentation by making open and modular software and hardware architectures available that smartly use novel and versatile radio technology. More-specifically, real-time Software-Defined Radio platforms meeting the requirements in terms of runtime latencies, throughput, and fast reconfiguration and reprogramming, are needed to fulfil this overall objective.

This deliverable reports the ORCA Year 3 achievement on the data plane of final operational SDR platforms, with focus on millimeter wave systems, physical layer improvements and higher layer SDR integration. Each focus area may contain multiple implementations. For instance, in door channel experiments for Massive MIMO, GFDM transceiver enhancements are all achievement belonging to *PHY layer improvement*. Furthermore, this deliverable also describes the testbed integration of the final available SDR platforms and how they are mapped to the ORCA showcases.

In the first focus area this deliverable describes a millimeter wave system with a newly developed 26GHz antenna frontend to allow experiments in 5G pioneer bands. This system includes beams steering functionality which can be used in industrial environments.

The next focus area of this deliverable, PHY Layer Improvements, describes channel experiments for Massive MIMO in ultra-dense indoor scenarios, GFDM Transceiver enhancements towards 5G including flexible numerology and increased spectral efficiency, FPGA footprint reduction of DSSS accelerator to realize virtualization of IEEE 802.15.4, the validation of multi-antenna based 802.15.4 concurrent transmission scheme and lastly dynamic padding of I/Q streams which ensures isolation between virtual radios over unreliable connections.

In the last focus area, Higher Layer SDR Integration, this deliverable introduces the TAISC MAC engine extension to support multiple PHY instances, the computational resource scaling for dynamically allocating resources to end-to-end network slices and lastly the 5G GFDM PHY integration into the Multi-RAT platform supporting mixed numerologies and 5G-LTE interworking.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

API	Application Programming Interface
AFW	Application framework
BB	Base Band
BBU	Baseband Unit
BS	Base Station
BWP	Bandwidth part
CFS	Completely Fair Scheduler
CMCVT	Concurrent Multi-Channel Virtual Transceiver
C-MAC	Concurrent Multiple Access Control
CPU	Central Processing Unit
C-RAN	Cloud-RAN (Radio Access Network)
DC	Dual Connectivity
DCI	Downlink Control Information
DDC	Digital Down Converter
DL	Downlink
DUC	Digital Up Converter
DSP	Digital Signal Processing
eNB	Evolved NodeB
EPC	Evolved Packet Core
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GFDM	Generalized Frequency Division Multiplexing
GW	Gateway
HV	Hardware Virtualization
IP	Internet Protocol
L1-L2	Layer 1 – Layer 2
LTE	Long Term Evolution
LWA	LTE-WLAN Aggregation
LWIP	LTE WLAN Radio Level Integration with IPsec Tunnel
MAI	Multiple Access Interference
MAC	Medium Access Control
MAC	Multiple Access Control
MCS	Modulation and Coding Scheme

NOMA	Non-Orthogonal Multiple Access
NR	New Radio
OpenRAN	Open Radio Access Network
OTA	Over the Air
OTFS	Orthogonal Time Frequency Space
PDCP	Packet Data Convergence Protocol
PHY	Physical layer
PRB	Physical Resource Block
PS	Processing System
PSS	Primary Synchronization Sequence
RAT	Radio Access Technology
RRC	Radio Resource Control
RT	Real-Time
Rx	Receiver / Reception
SAP	Service Access Point
SCS	Subcarrier Spacing
SDN	Software-Defined Network
SDR	Software-Defined Radio
SDU	Service Data Unit
SIC	Successive Interference Cancelation
TCP	Transmission Control Protocol
TS	Terminal Station
TTI	Transmission Time Interval
Tx	Transmitter / Transmission
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink

1 INTRODUCTION

This deliverable focuses on the real-time software defined radio (SDR) platforms finalized and operational in Year 3 of ORCA. These platforms are described with focus on the DATA plane functionality. The CONTROL plane features developed in the Year 3 of ORCA are described in D4.5 [1]. Figure 1 shows the structure of this deliverable with its single functionalities and their mapping to testbeds and showcases.

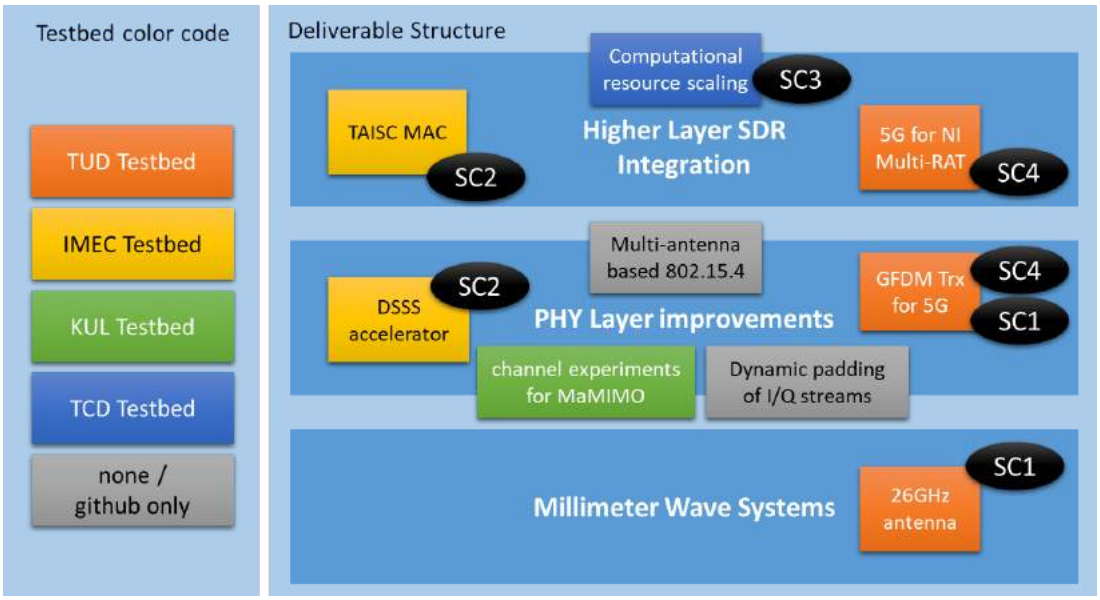

Figure 1 Deliverable structure with mapping to testbeds and showcases

The developments are grouped into three main chapters millimetre wave systems, physical (PHY) layer improvements and higher layer SDR integration. Each contribution is introduced in three aspects: motivation --- why it is developed, implementation --- how it is developed, and testbed integration --- how other people can use it. In the following the single contributions will be briefly described and mapped to the ORCA showcases which are described in detail in D2.5 [2].

In the Millimeter – Wave Systems chapter 2, the integration of the **GFDM PHY** with a new developed **26GHz antenna** frontend is described by TUD to allow experiments in 5G pioneer bands. This hardware with its beams steering functionality demonstrates in Showcase 1 how mmWave systems can be used in an industrial environment.

The PHY Layer Improvements chapter 3 describes the indoor **channel experiments for MaMIMO** done by KUL as a case study for broadband communications, in ultra-dense indoor scenarios with fixed users**.** In this experiment, the analysis is centred in the influence of the antennas array and the spectral efficiency**.** Further the **GFDM Transceiver** enhancements towards **5G** achieved by TUD, including flexible numerology and increased spectral efficiency of the multi-user scenario are described and integrated into Showcase 1 and Showcase 4. Next the FPGA footprint reduction of **DSSS accelerator** to realize virtualization of IEEE 802.15.4 by IMEC is introduced which is used in showcase 2 as physical layer of data plane that offers an API for real-time configuration. Subsequently the validation of **multi-antenna based 802.15.4** concurrent transmission scheme in simulation is described by IMEC. Lastly the development of **dynamic padding of I/Q streams** which ensures isolation between virtual radios over unreliable connections is represented by TCD and made available at github.

Chapter 4 - Higher Layer SDR Integration introduces first the **TAISC MAC** engine extension to support multiple PHY layer instances. This development, done by IMEC, is used in showcase 2, as MAC layer

for the data plane. Secondly TCD describes the **computational resource scaling** for dynamically allocating resources to end-to-end network slices, which operate on top of a virtualized network infrastructure comprised of hybrid SDR-SDN functionality, possessing RATs, switches and controllers implemented in software. This development is integrated into showcase 3. Lastly the TUD **5G** GFDM PHY integration into the **NI Multi-RAT** platform, done by NI, using ns-3 for higher layers by supporting mixed numerologies and 5G-LTE interworking is described in this deliverable and part of the showcase 4.

2 MILLIMETER – WAVE SYSTEMS

2.1 Integration of GFDM PHY with 26GHz antenna

2.1.1 Motivation

One of the goals of TUD for year 3 in the ORCA project was to develop mmWave frontends depicted in Figure 2, that are compatible to the TUD's SDRs. The idea of this new hardware is to allow experiments in 26 GHz bands using TUD's hardware and testbed.

Figure 2 26GHz mmWave frontends

One important application is the real time beam steering capability of TUD's hardware. This kind of development and experiment targets to serve as proof-of-concept for novel future wireless technologies that employ the mmWave bands. In particular, when the targeted use case requires mobility, which is likely the case in the future factory with robots, the beam tracking functionality plays a major role on the feasibility of the mmWave bands in such kind of application scenario.

Thus, this subsection describes the TUD's development involving the 26 GHz frontends and its integration to TUD's testbed.

2.1.2 Implementation Results

The 26 GHz mmWave front ends have been integrated to the GFDM implementation of TUD. The setup is depicted in Figure 3.

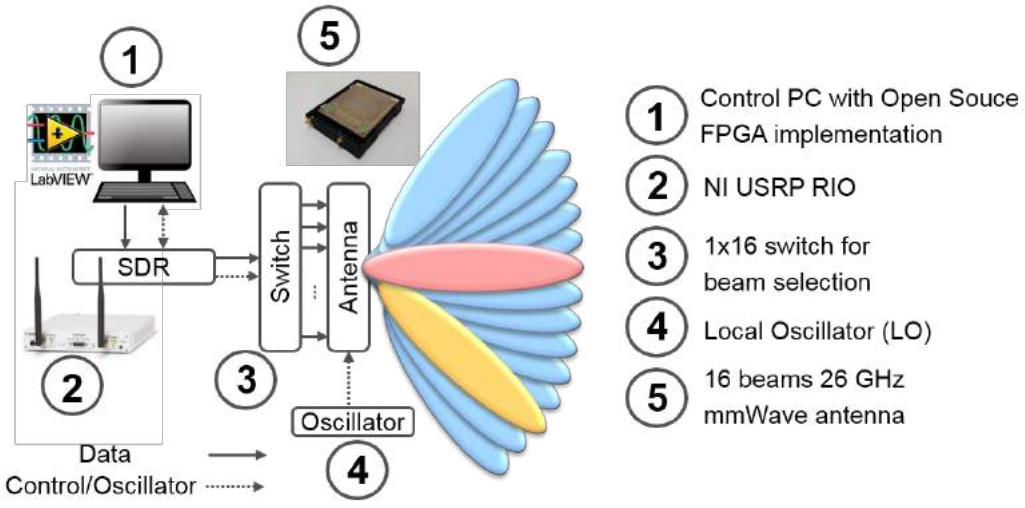

Figure 3 mmWave setup

1. Refers to the host PC. For example, it can execute the host side of the GFDM implementation of TUD, which is described in D3.1 [3]. This part of the system is responsible for sending the data down to the FPGA or reading it.
2. In order to allow a variety of experiments, we have different SDR platforms available in the testbed, which are USRP B200 mini, USRP 2953, USRP N210 and USRP 2920. Further information can be found in http://owl.ifn.et.tu-dresden.de/testbed/.
3. The 1x16 Mini-Circuits switch USB-1SP16T-83H is used as a beam selector for the mmWave antenna. The switch's input is connected to the SDR's output with carrier frequency of 2.4 GHz. The 16 switch's outputs are then connected 16 inputs of the multi-beam mmWave antenna of Figure 2.
   Since each input corresponds to a beam, we are able to select the antenna beam by selecting the switch's output port. The switch can be controlled by Python, Matlab, LabVIEW and other programming languages, information on software support and documentation can be found in https://www.minicircuits.com/softwaredownload/solidstate.html. With the collaboration of the partners of KUL, we use the USRP GPIO connection with a serial cable to select the switches' inputs or outputs. This solution allows fast beam switch for the real-time beam steering functionality. In particular, the minimum time transition between two beams is 15 us, while the alternative USB based solution is 2 ms.
4. We also employ a local oscillator that generates a signal of 11.8 GHz to feed the mmWave antenna, which is depicted in Figure 4. More information about this device is available in https://github.com/erainstruments/erasynth-docs/blob/master/erasynth-users-guide.pdf.

Figure 4 Signal generation (Oscillator)

5. Finally, the antenna depicted in Figure 2 is connected to all outputs of the switch and the local oscillator. Each input of the antenna corresponds to a beam, as depicted in Figure 1 below. Thus, this setup allows the beam steering algorithm to be performed at the FPGA level, which is suitable in the mobility use case, where the beam tracking needs to be achieved fast.

In the following, we describe our real-time beam steering algorithm solution that is integrated with TUD's GFDM implementation at the receiver side.

2.1.2.1 Real time beam steering data path focusing on RX side

The goal of this development is to integrate the GFDM implementation with the new mmWave frontends in a way that the beam alignment is performed automatically, even when mobility occurs. In the case where the channel changes due to mobility of blockage, the receiver automatically searches for a new beam that provides a better channel quality.

This section describes the beam selection algorithm working independent of the control channel, while the overall algorithm and control channel is introduced in deliverable D4.5 [1]. The block diagram of the FPGA implementation is depicted in the in Figure 5, where the new blocks added in this year are highlighted with blue color. Please note that the antenna and switch block are described in the previous

subsection.

Assuming a constant transmission, the easiest and most effective solution is to simply measure the channel quality by an energy computation that is done by summing the power of a given amount of received samples, by the "Energy Computation" block. According to the switch datasheet, the transition time of one switch port to another is 15us. Since the FPGA is set to work with a rate of 40 MCycles/s, we need to wait 40M * 15u = 600 cycles until we are sure that the selected port has changed. Experimentally, we have found that an amount of 1900 samples for energy computation works successfully. Therefore, considering both 600 waiting cycles plus 1900 samples to average, the amount of necessary time to probe one beam at RX is 2500/40M = 62,5 us.

The "Beam Selection" block compares the value provided by the "Energy Computation" block with a empirical threshold. If this value is greater than the threshold, it declares beam found and keeps this beam as long as this condition is kept. When there is mobility of the link or blocked, causing the energy to fall below the threshold, the "Beam Selection" block changes the switch output sequentially until it finds a beam whose energy is greater than the threshold again.

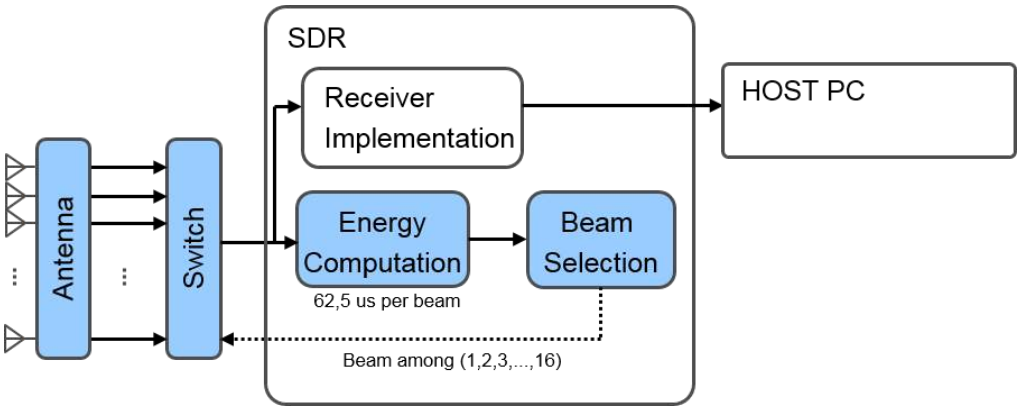

Figure 5 FPGA Beam Steering Diagram

2.1.2.2 Channel sounding algorithm

The goal of the channel characterization is to provide an intermediate step between the algorithm development and the real-time implementation. In particular, the channel sounding allows a precise verification of the assumptions when designing the beam steering algorithm. Since the collected data is in baseband, it contains all the RF impairments introduced by the antenna hardware which are reflected in the SNR and channel response.

At the TX, a signal based on the well-known Chadoff-Chu synchronization sequence is sent with the mmWave frontends. At first, this sequence is created at the host PC. Then, using the streaming example of NI, this signal is transferred to the FPGA and then sent via the setup described in Figure 3. At the RX, the collected baseband signal is uploaded to the host PC, and then we perform the channel measurement according to the block diagram depicted in Figure 6.

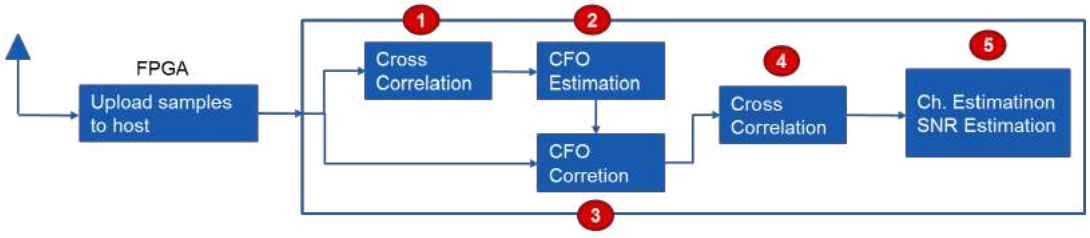

Figure 6 Channel sounding RX block diagram

We have performed the measurements 4 times, with 4 different angles between the transmitter and receiver. The angles are -15, 0, 15 and 30 degrees. For instance, angle = 0 means that the antennas are facing each other with no deviation. Angle = -15 means that the RX antenna is rotated in its own axis by -15 degrees, and so on. The antennas are separated by a distance of 60 cm with the same height, and the transmitted bandwidth was 10 MHz. The results demonstrated in Figure 7 show the SNR obtained for each beam combination position. Because this experiment was performed with the antennas being relatively close, we observe several side lobes, indicating that in this case, a less robust beam steering algorithm could work successfully. In addition, another important information is the measured SNR, which achieves 21 dB.

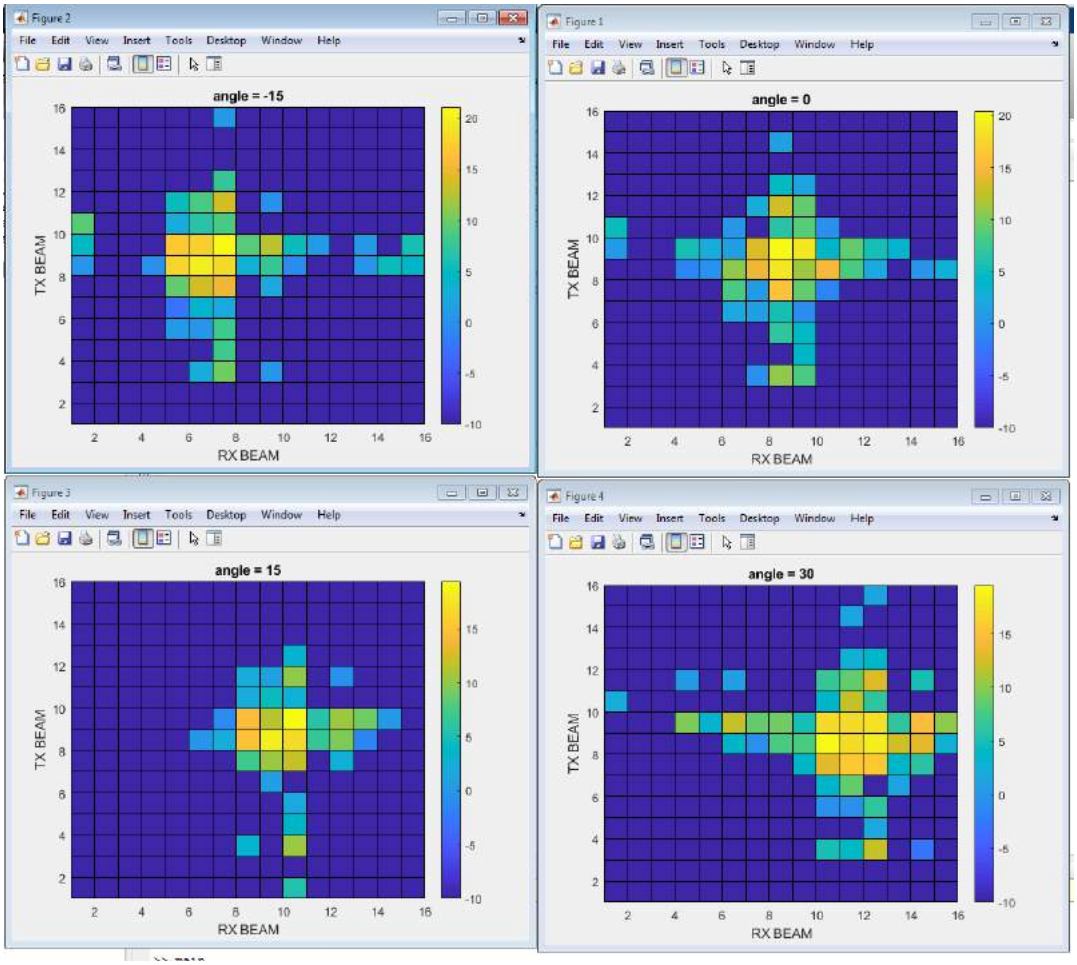

Figure 7 mmWave channel characterization

2.1.3 Testbed Integration

For year 3, TUD has included two pairs of the mmWave antennas in its testbed. That is, TUD's testbed is now equipped with four times the setup shown in Figure 3. In general, these antennas can be used with any SDR available in the TUD's testbed, including experiments with multiple users. Further information can be found in the testbed's webpage http://owl.ifn.et.tu-dresden.de/orca/mmwave26ghz/. In addition, the software related to the beam steering and channel sounding are accessible from the testbed.

3 PHY LAYER IMPROVEMENTS

3.1 Indoor channel experiments for MaMIMO

3.1.1 Motivation

In a massive MIMO (MaMIMO) system, the distribution of its antennas and topology impact the spectrum efficiency performance in the system. The main case of study theoretically and experimentally is the application of massive MIMO in broadband communications in outdoor scenarios. However, in ultra-dense scenarios the multipath components and the number of users to be served creates a more challenging case of study. During this year, the KULeuven massive MIMO incorporates a modular antenna design and allows multiple antenna topologies to be easily deployed indoor.

3.1.2 Implementation Results

A massive MIMO experiment was carried out to emulate an ultra-dense indoor scenario, where traffic sensors (or users) were deployed on the ground in a grid distribution with a separation of 25 and 30 cm, with the aid of 4 XY positioners. Each XY positioner holds a dipole antenna, that are controlled for two independent USRPs via the Single User MIMO Application Framework 1.1.

The dipoles antennas mounted on the positioners moved automatically and synchronously to 30 different positions every 30 seconds, with simultaneous and continuous signal transmission, more details about the testbed facilities are available in D6.5. The received uplink signal, generated by the USRPs, was collected, processed by a 64-antenna massive MIMO and controlled in real-time by LabVIEW Communications Base Station MIMO Application Framework 1.1.

One of the main characteristics of the 64 patch antennas at the base station is their modularity. This experiment was replicated under three different antenna topologies, please refer to D6.5 Section 2.2.4 to see the exact antenna placement:

- *Uniform Rectangular Array (URA),* in an array of 8 × 8 antennas.
- *Uniform Linear Array (ULA),* in an array of 1 × 64 antennas.
- *Distributed Uniform Linear Array (D-ULA),* 8 different arrays of 1 × 8 antennas each.

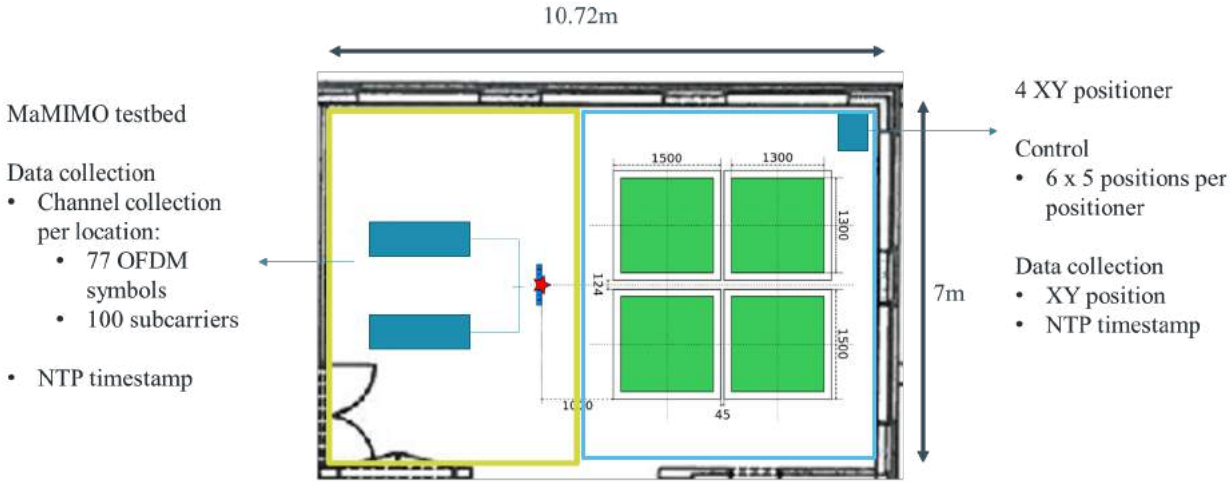

Figure 8 Indoor scenario deployment and data collection

The data collected in the Massive MIMO testbed is the channel information and the *Network Time Protocol (NTP)* timestamp of each channel sample. The XY controller saves also the *NTP* timestamp

and the XY position. In an off-line data processing, the channel information is associated to each user's position, via NTP timestamp, thus a virtual ultra-dense scenario with 120 users is studied. As an example, Figure 9 shows the channel gain received to antenna 1 during the experiment for 30 different locations, each dotted line represents the sample in where the NTP timestamp is collected. The channel collection between two NTP timestamps in the virtual scenario belongs to a different user.

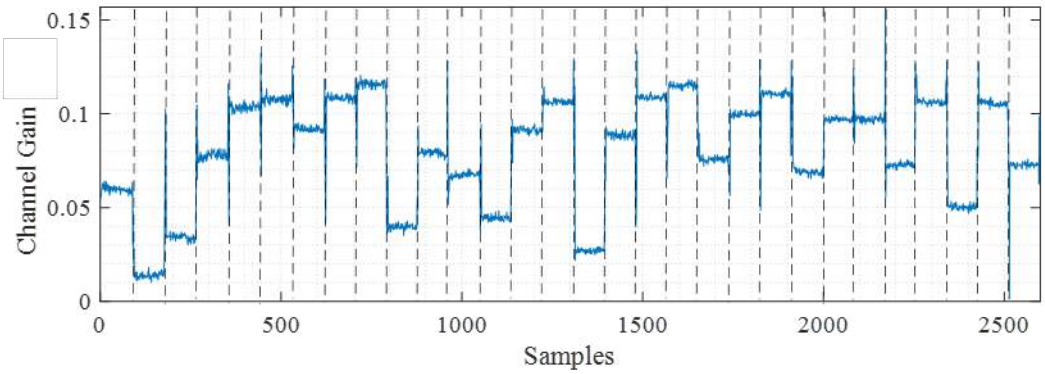

Figure 9 Channel gain collected in antenna 1 for different positions over time, and URA distribution

From the ultra-dense scenario described above, an analysis of the spectral efficiency based on the number of select users was carried out for the different antenna topologies. The distribution of the antennas as D-ULA increases the spectrum efficiency and supports a higher number of users, than the other antenna topologies. [4]

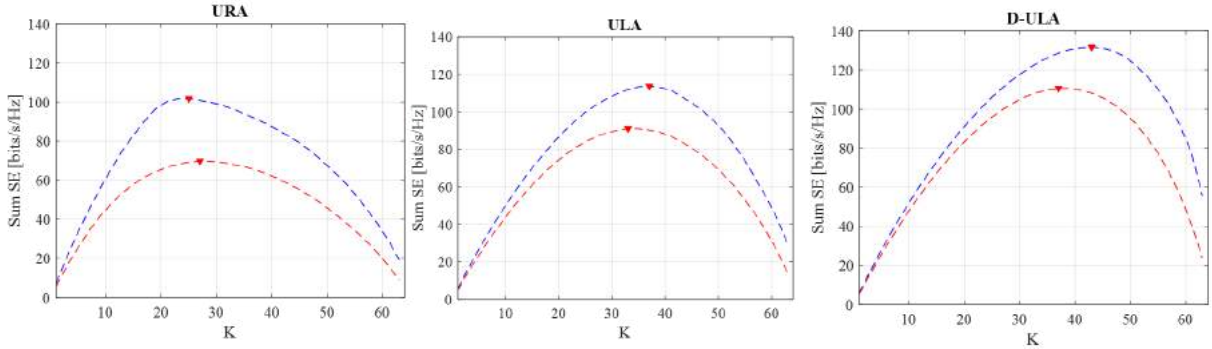

Figure 10 Sum SE comparison for optimal and random user grouping techniques, the number of selected users (K), and antenna topology

3.1.3 Testbed Integration

The NTP collection is implemented in the massive MIMO testbed, to collect its timestamp and save it along with the channel information. NTP is essential to create the virtual ultra-dense scenario, separating the information at each user position.

The XY positioner is deployed to dynamically configure user location in the experimental setup, more information is reported in D6.5. The access to the XY positioners for the aforementioned experiment is via an external controller; however, this access will be implemented through the massive MIMO testbed.

3.2 GFDM Transceiver enhancements towards 5G

3.2.1 Motivation

This section presents the transceiver enhancements of the TUD GFDM PHY implementation, including flexible numerology and increased spectral efficiency of the multi-user scenario. The research [5] pointed out that a new unified architecture model of the GFDM is more suitable for providing flexibility, including flexible numerology and flexible multi-user support. Figure 11 represents the core structure of the unified GFDM modem. As shown in the block diagram, the GFDM modem can be implemented very efficiently with the Discrete Fourier Transform (DFT) modules, especially when the K and M are radix-2 numbers, where the Fast Fourier Transform (FFT) module can be used.

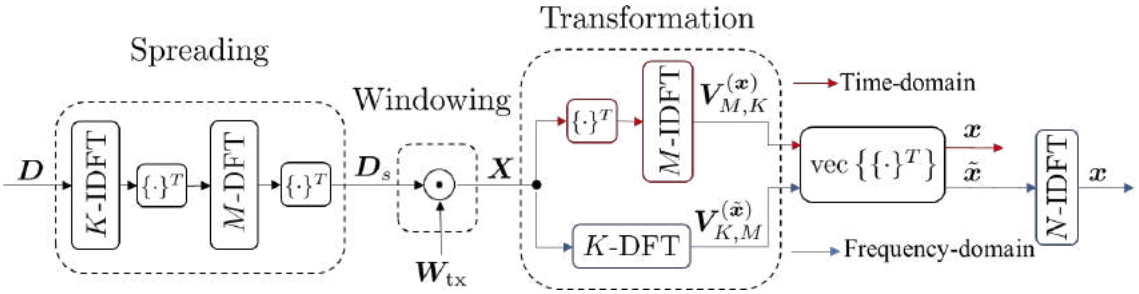

Figure 11 Unified Architecture of GFDM

In the following subsection, we will show the implementation result of this architecture and the feature of supporting flexible numerology and enhancing the efficiency for the multi-user case.

3.2.2 Implementation Results

3.2.2.1 Flexible numerologies

Figure 12 describes the block diagram of the GFDM radix-2 modem implementation. Compared with the original implementation, this one consumes less DSPs and has a more flexible range for choosing parameter M, which is essential for different numerologies.

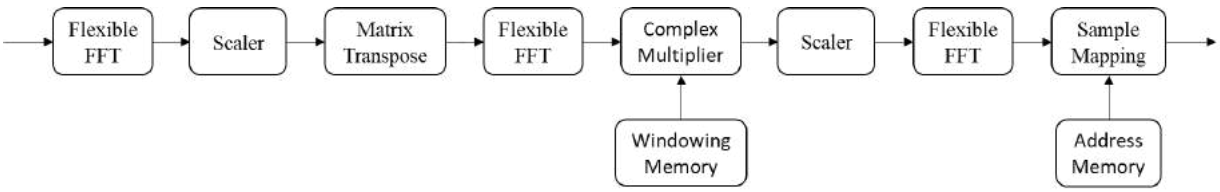

Figure 12 Block Diagram of GFDM Radix-2 Implementation

The hardware usage evaluated by the LabVIEW Communications Compile Worker is shown in Figure 13. Also, using this structure, one can generate different waveforms, including GFDM time-domain signal, GFDM frequency domain signal, OFDM signal, OTFS signal, DFT-Spread-OFDM signal, etc. In the next sections, we indicate that one can easily add 5G-like features to the GFDM PHY using the new implementation.

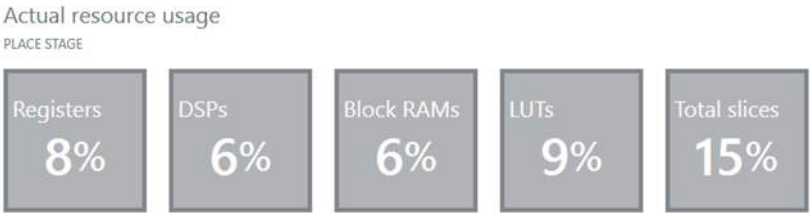

Figure 13 Hardware Usage of the Implementation of GFDM Radix-2 Modulator

As well known, the GFDM localizes subcarriers by performing circular windowing. These localized subcarriers will be transmitted in different subsymbols. By changing the parameter K and M, one can easily change the numerology of the transmission. Additionally, if the orthogonal window is used, changing parameter K and M while using fixed $N = KM$ is equivalent to changing subcarrier spacing.

The new implementation of the GFDM PHY supports a full range parameterization of K and M, which can be changed during run-time. This flexibility provides flexible numerology at the transmitter and the receiver. The numerology in terms of subcarrier spacing is defined in the current 5G standard.

3.2.2.2 Multiple User support

In the conventional GFDM, the data for different users are allocated in the data matrix $\boldsymbol{D}$, which means a full-size GFDM modulation needs to be performed in downlink and uplink, although the data matrix is sparse. Furthermore, sample-wise control in the generated GFDM vector is not available, which impairs the spectral efficiency. Our new research reveals that one can perform the allocation in the frequency domain after the frequency domain GFDM modulation. In this way, only a small size modulation for the non-zero part is needed for each user, and the spectral efficiency is improved. As shown in Figure 14, one can see that the modulated non-zero samples are better allocated in the frequency domain.

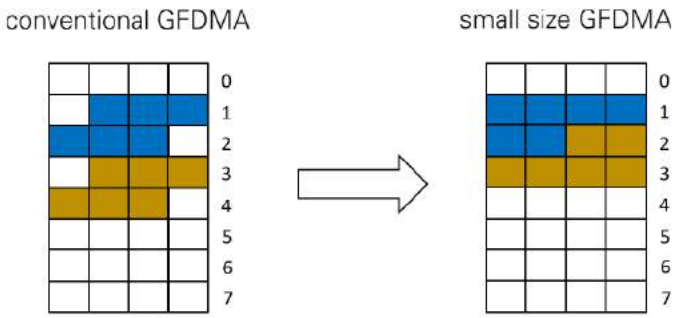

Figure 14 Comparison of Conventional GFDMA & Small Size GFDMA

Figure 15 presents the modulation scheme of the implemented GFDMA. With the new implementation, one can easily change the modulation scheme to perform the small size modulations for different users and use the mapping module at the last stage to perform the new allocation scheme.

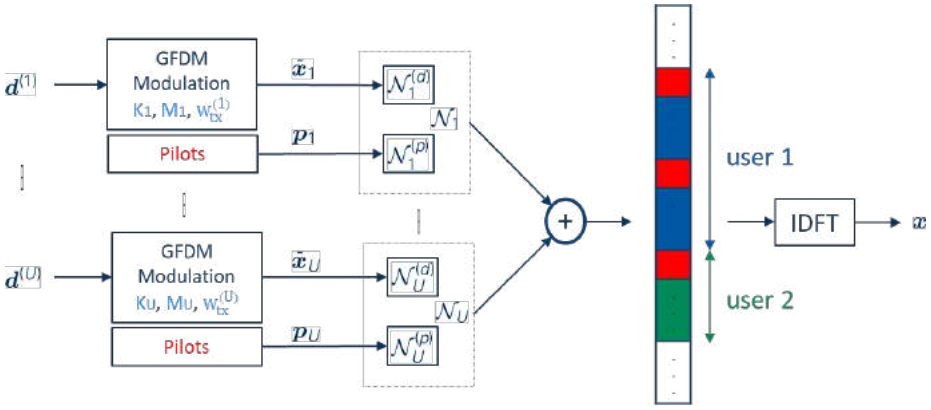

Figure 15 Flexible GFDMA Model

For the Time Division Multiple Access (TDMA), we implement a MAC layer on top of the current GFDM PHY implementation. Using the new MAC layer at the base station allows it to schedule the users into different time slots with different priorities, also it can receive data or transmit data from or to the users. Figure 16 represents a demonstration of the TDMA multi-user functionality. The robot arms (ID3, ID4) are using a D2D link, and the base station (ID0) schedules the transmitting time slots for the robot arms. The base station also forwards the control data for the AGV (ID5) from the remote controller and the processed data for the Air-Hockey table (ID2) using different time slots.

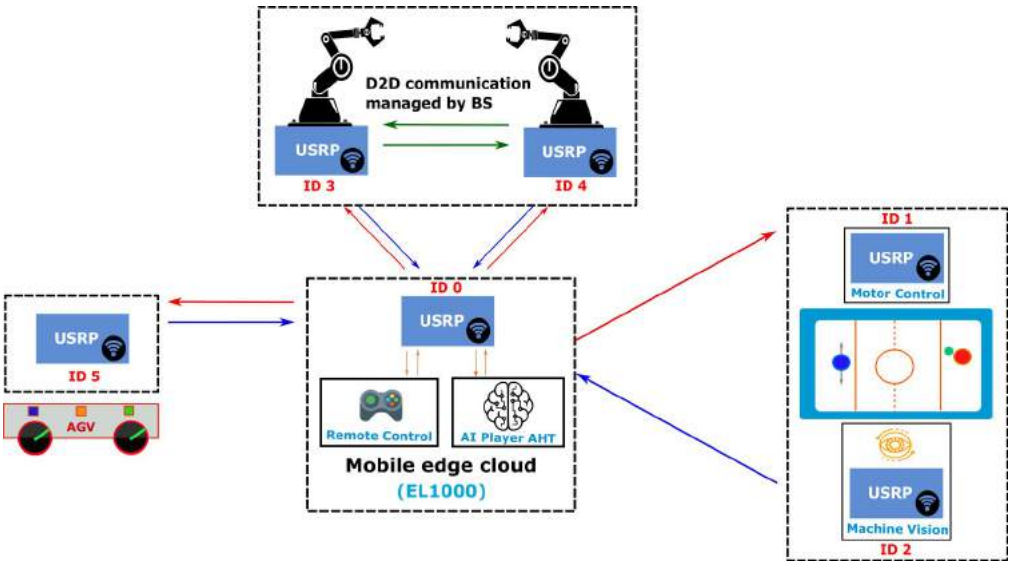

Figure 16 Demonstration of Multi-User Support

3.2.3 Testbed Integration

The development shown is this section is available in the testbed, and the multi-user system can be accessed via the public git repository, https://fusionforge.zih.tu-dresden.de/projects/flexiblegfdmphy/. Documentation and references regarding the GFDM development are also found in the testbed webpage http://owl.ifn.et.tu-dresden.de/GFDM/.

3.3 FPGA footprint reduction of DSSS accelerator to realize virtualization of IEEE 802.15.4

3.3.1 Motivation

The number of end-devices connected in wireless communication standards/protocols in Internet of Things (IoT) paradigm has proliferated significantly, most of the commercial state-of-the-art wireless Gateways (GW) or Access Points (AP) , offers single Channel (CH) radio link (i.e., only a single end-device can communicate with the GW at a time). The quality of service of a single channel GW can be further improved by employing a multi-channel transceiver, which is capable of transmitting/receiving data on different radio CHs simultaneously. Thanks to SDR, we have the opportunity to implement a prototype of such a multi-channel GW, however the existing implementations use a dedicated Physical (PHY) for each channel. This simple approach is not only inefficient in terms of hardware utilization, but also underutilizes the potential processing capability of an SDR. For example, a modern SDR containing FPGA that can process the data at a rate far higher than the wireless standard used by IoT device.

In this deliverable, we introduce an SDR based Concurrent Multi-Channel Virtual Transceiver (CMCVT) which applies hardware virtualization (HW) on programmable hardware part of an SDR. The benefits of the CMCVT are twofold; (1) it utilizes the same physical PHY layer to generate multiple virtual transceivers, making it highly efficient in terms of hardware resources, (2) the high processing capability of a modern SDR is exploited by implementing the PHY layer on the FPGA part of an SDR. More details are given in the next section.

3.3.2 Implementation Results

The block diagram that visualizes the main differences between the traditional way of implementing a multi-channel transceiver and the proposed approach is given in Figure 17. As depicted in Figure 17 (a), the existing designs assign a separate PHY layer to each channel, while in our design, the same PHY layer is timely shared among multiple channels. We apply the overclocking concept in our design, resulting our PHY layer runs at N×CLK bb, where N and CLK bb represent the number of channels and a standard compliant BaseBand (BB) clock rate, respectively. It is obvious that each PHY layer can be further decoupled into TX PHY and RX PHY layers. The PHY layers of the CMCVT are detailed in the subsequent sections. To shape our concept into real time system, we have selected the IEEE 802.15.4 standard, but the concept is applicable to any kind of PHY. Furthermore, the standard specifies three different PHY modes (i.e., 20kbps, 40kbps, 250kbps) where each mode operates in a particular frequency band. We have chosen the PHY layer of data rate 250kbps functioning in 2.4GHz band. It is worth noting that our concept is wireless-standard agnostic.

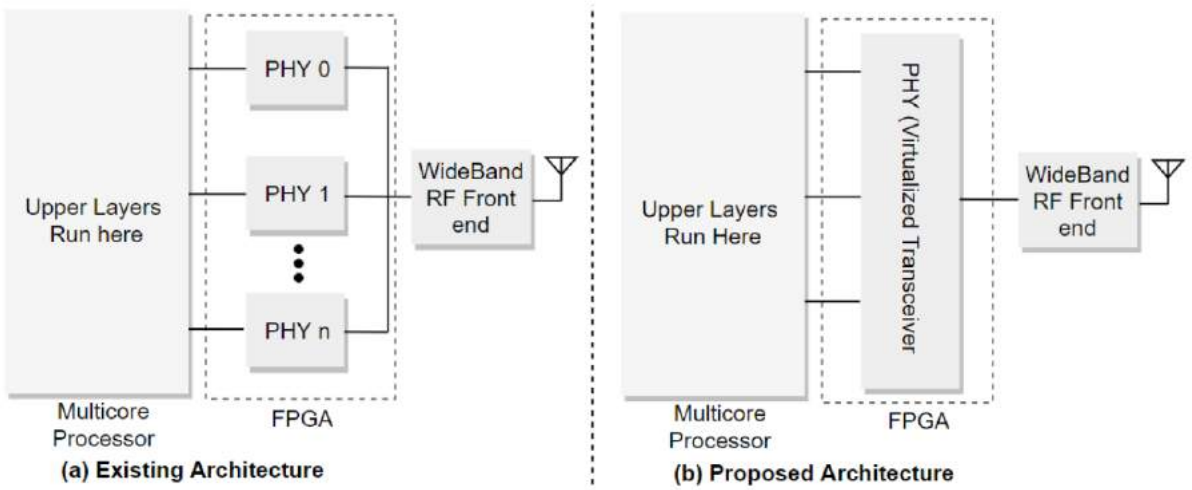

Figure 17 A block diagram showing a general comparison between (a) the traditional way of implementing a multi-channel transceiver and (b) the proposed approach

3.3.2.1 Physical layer of the multi-channel virtual transmitter

A detailed diagram of the Multi-Channel Virtual Transmitter (MCVT) is highlighted in Figure 18. The PHY layer of MCVT implemented in FPGA is broadly composed of Digital Up Converter (DUC) bank, and Baseband Processing Unit (BPU). The working flow of the MCVT is as

follow:

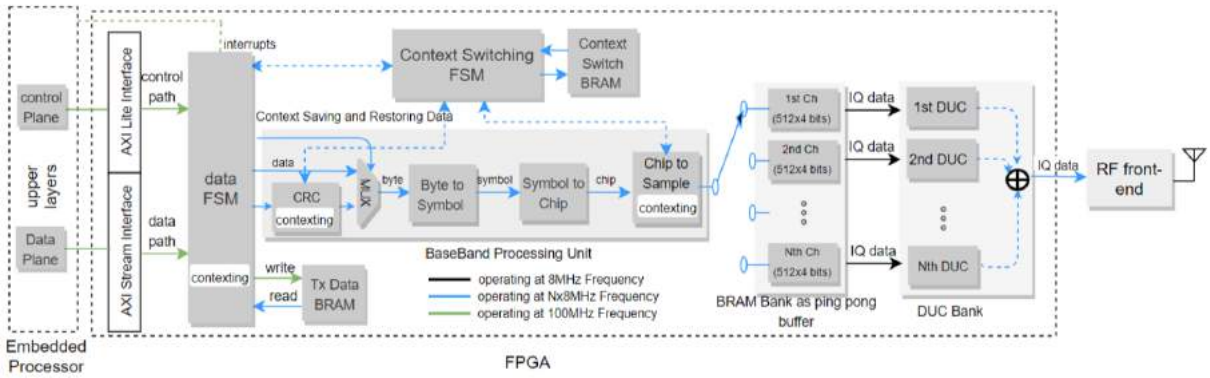

Figure 18 A detailed diagram showing modules involved in the multi-channel virtual transmitter

1. The Medium Access Control (MAC) layer running on embedded Processor System (PS) configure the BPU, and RF front-end (as indicated by control path in Figure 2). For instance, sampling frequency, and bandwidth of RF front-end, the number of potential Channels (CHs) on which data is to be transmitted in BPU, etc.
2. Then, the MAC layer starts sending data to BPU through DMA (indicated by data path in Figure 18). The BPU keeps storing the data into RAM, until it receives the corresponding data of all the CH(s) defined in step (1).
3. Finally, the BPU initiates the transmission, and by generating interrupts, it informs the MAC layer about the status of transmission.

We leverage multitasking, pipelining, and multi-clock domains in virtualizing the BPU. These techniques are detailed in the following sub-sections.

3.3.2.1.1 Multitasking

Multi-tasking is initially introduced in a CPU, the key challenge here is context switching, which comes down to storing-restoring memory, pipeline and internal states of a program. Our design in hardware is faced with similar problem. Therefore, we introduce a context switching Finite State Machine (FSM) and Block Random Access Memory (BRAM) in our design. Furthermore, we observe only three modules require context switching: data FSM (due to internal states), CRC (due to memory) and Chip to Sample converter (due to delay), as shown in the previous figure.

3.3.2.1.2 Pipelining

The pipelining has helped us in reducing the clock overhead caused by context saving and restoring operations. An example of pipelining for N channels is illustrated in Figure 19, where horizontal axis represents the time and vertical axis reflects the modules of the BPU. At "t0", M0 module begins processing the data of CH1. After a unit time (ie at "t2", it is hereinafter referred to as tick), M0 switch to CH2, meanwhile M1 module is enabled for CH1. During the 3rd third tick, all the modules are busy in processing the data of consecutive channels. The output of M3 is stored into BRAM which decouples the BPU from DUC bank. The M3 generates 256 symbols after each tick and the width of each symbol is 4 bits (i.e., LSB 2 bits belong to I sample, and MSB 2 bits belong to Q sample). There is a separate BRAM for each channel, having the capacity to store 512 symbols. The BRAM behaves as a ping pong buffer which means the RAM can accept new symbols for 257-512 locations while sending out the symbols stored in the first 256 locations or vice versa. This configuration avoids the related DUC from the processing of wrong IQ symbols.

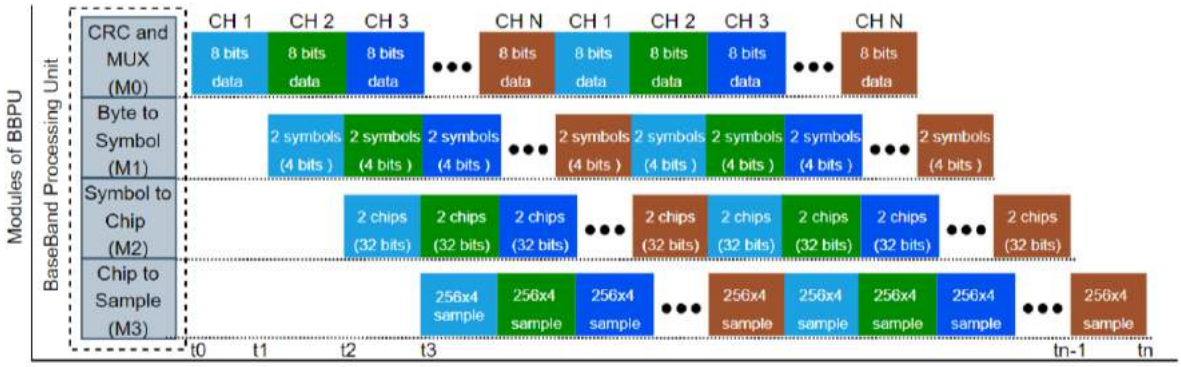

Figure 19 Applying pipelining in the Multi-Channel Virtual Transmitter

3.3.2.1.3 Multi-clock domains

To meet with critical constraints of the specific standard, the MCVT uses three different clocks; (1) the clock specified for control and data paths (100MHz in our design), (2) the BPU runs at N×CLKbb (CLKbb is 8MHz in our design), and (3) DUC bank reads the data from ping pong BRAM at CLKbb rate. Due to the presence of multi-clock domains, the MCVT is prone to metastability. To mitigate the metastability situation, we have introduced 2-flop synchronizer and dual port BRAM for single-bit, and multi-bits data signals, respectively.

3.3.2.2 Physical layer of the multi-channel virtual receiver

Unlike MCVT, turning the Multi-Channel Virtual Receiver (MCVR) from concept into reality was a challenge for us. Almost all the modules constitute the MCVR needs context saving and restoring, adding extra complexity in the architecture. A detailed diagram of the MCVR is elaborated in Figure 20, where the corresponding PHY layer incorporates BPU and Digital Down Converter (DDC) bank is realized in FPGA. It is worth noting that the HV is only applied on BPU of MCVR. Likewise, the MCVR

benefits from multi-clock crossing, pipelining and multitasking, which together alleviate the clock overhead added during context saving and restoring.

The decoding flow of the MCVT is as follows:

1. After configuration of PHY layer and RF front-end from the upper layer (indicated by control plane in the Figure 4), the DDCs of related CHs in DDC bank first shift, down sample and then write the incoming wideband signals to the BRAMs.
2. The BPU begins decoding the samples reading from first BRAM (indicated by 1st CH in Figure 4).
3. After a tick (i.e., reading 8 samples), the sample reading pointer switches to the second BRAM, meanwhile context-switching FSM concurrently performs context saving for the current CH and restoring for the subsequent CH.

For upper layer to recognize that the incoming packet belongs to which CH, the BPU appends one extra byte with the decoded data representing the CH number.

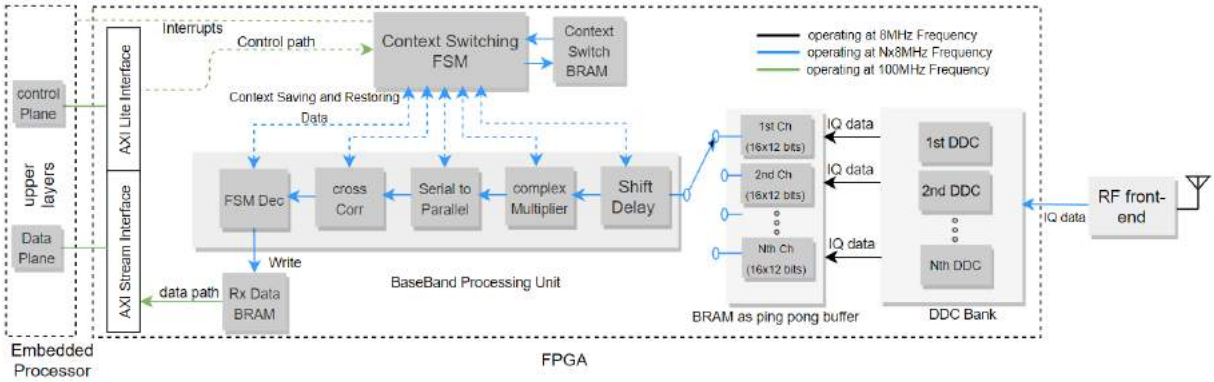

Figure 20 A detailed diagram showing all the modules involved in realization of the Multi-Channel Virtual Receiver.

3.3.2.3 Results and Discussions

Like every design in FPGA, our BPUs has logic and memory parts. The logic part of a design is mapped on Look-up Table (LUT) and Flip-Flop (FF), whereas the memory part is placed on RAM (better explained in the subsequent section). Instead of directly comparing the hardware utilization of CMCVT with its existing CMC-TRX, MCVT and MCVR are separately compared against their corresponding conventional approaches. Table 1 depicts the comparison of hardware utilization efficiency of our MCVR against Conventional Multi-Channel Receiver (CMCR), while Table 2 illustrates the comparison of our MCVT against Conventional Multi-Channel Transmitter (CMCT). The tables only contain LUTs and FFs (i.e., they only include logic parts of the respective designs). The efficiency shown in the tables represents the relatively improved hardware resources of the CMCVT, calculated as follows:

$$Efficiency = \left(\frac{HW_{conv} - HW_{our}}{HW_{conv}} \right) \times 100$$

Where HW_conv and HW_our represent the hardware consumed by conventional and our approaches, respectively and can be LUT, FF or RAM. It can be seen from tables below that the relative efficiency is increased as the number of concurrent channels increase. Below 2 parallel CHs, our approach provides either negative (in the MCVR case) or slightly improved efficiency (in the MCVT case). This is expected, because our approach involves extra logic for context storing-restoring overhead (e.g., context-switching FSM in Figures 2&4). However, the most part of the extra logic do not change as the

number of parallel CH increases, resulting to significantly improved efficiency. For instance, the MCVR saves 82.95% FFs and 67.51% LUTs as compared to CMCR when it is operated in 8 parallel CH mode. Likewise, the MCVT (as shown in Table 2) reduces 81.87% FFs and 66.04% LUTs.

CHs	Resource	CMCR	MCVR	EFFICIENCY
1	LUTs	854	854	0
	FFs	1073	1073	0
2	LUTs	1708	2128	-24.59
	FFs	2146	1412	34.20
4	LUTs	3416	2168	36.53
	FFs	4292	1432	66.64
8	LUTs	6832	2220	67.51
	FFs	8584	1464	82.95

Table 1 A Comparison of Hardware Utilization Efficiency of Conventional Multi-Channel Receiver and Our MCVR under the combined Logic Setting.

CHs	Resource	CMCT	MCVT	EFFICIENCY
1	LUTs	272	272	0
	FFs	442	442	0
2	LUTs	544	532	2.21
	FFs	884	565	36.09
4	LUTs	1088	619	43.11
	FFs	1768	593	66.46
8	LUTs	2176	739	66.04
	FFs	3536	641	81.87

Table 2 A Comparison of Hardware Utilization Efficiency of Conventional Multi-Channel Transmitter and Our MCVT under the combined Logic Setting.

3.3.3 Testbed Integration

The implementation is available on git repository, users with interests can contact the repository owner to have access and test it on the ORCA facility. The repository host full stack Zigbee solution, which is also offered in ORCA open call. More details are elaborated on the testbed documentation https://doc.ilabt.imec.be/ilabt/wilab/tutorials/taisc_on_zed.html.

3.4 Multi-antenna based 802.15.4 concurrent transmission scheme

3.4.1 Motivation

This Section introduces a Non-Orthogonal Multiple Access (NOMA) scheme to support concurrent transmission of multiple IEEE 802.15.4 packets. Not like collision avoidance Multiple Access Control (MAC), concurrent transmission supports Concurrent-MAC (C-MAC) where packet collision is allowed. The communication latency can be reduced by C-MAC because a user can transmit immediately without waiting for the completion of other users' transmission. The big challenge of concurrent transmission is that error free demodulation of multiple collided packets hardly can be achieved due to severe Multiple Access Interference (MAI). To improve the demodulation performance with MAI presented, we introduce an architecture with multiple switching antennas sharing a single analog transceiver to capture spatial character of different users. Successive Interference Cancellation (SIC) algorithm is designed to separate collided packets by utilizing the spatial character.

3.4.2 Implementation Results

In this section, we describe a co-channel concurrent transmission scheme which is based on switching antennas. In our design, multiple antennas share a single RF transceiver, and capture spatial character of transmission from different users. Together with SIC operation, multi-user signals can be demodulated successfully utilizing spatial character of different users. IEEE 802.15.4 O-QPSK physical layer is taken as an example. The scheme can also be applied to other system, where multiple users sharing the same set of spreading sequence, such as the Physical Random Access Channel (PRACH) in 4G/5G and GPS system. In the rest part of this section, we will first introduce traditional 802.15.4 transceiver and interference model of concurrent transmission; then we propose the NOMA scheme based on antenna switching and SIC procedure; finally, we show the Packet Error Rate (PER) simulation result of multi-user concurrent transmission under different array size.

3.4.2.1 Traditional 802.15.4 transceiver and interference model

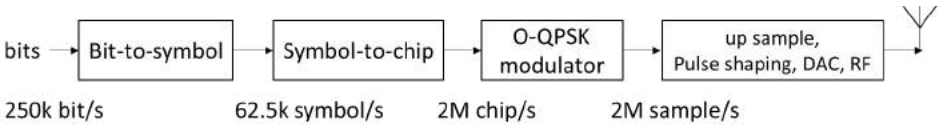

Figure 21 IEEE 802.15.4 O-QPSK 250kbps transmitter

The figure above shows the processing steps of transmitter defined in 802.15.4-2015 standard [6]. Every 4 bits, also known as a symbol, are mapped to a length 32 chip sequence by "Table 12-1" in the standard. The mapping is also shown in the table below.

Data symbol	*Chip values (c0 c1 … c30 c31)*
0	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
2	0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
3	0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
4	0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1
5	0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0
6	1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1
7	1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1
8	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
9	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1
10	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1
11	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
12	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0
13	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1
14	1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 0 1 1 0 0
15	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 0

Table 3 Symbol-to-chip mapping for IEEE 802.15.4 2450/2380 MHz frequency bands

Notice that the first 8 chip sequences are cyclic shifted version of the first chip sequence. The last 8 chip sequences are cyclic shifted version of the 9th chip sequence. So, 16 chip sequences actually are generated by 2 basic chip sequences. This leads to severe interference as you will see later analysis.

The receiver performs inverse processing of transmitter. The core step is determining which chip sequence has been transmitted by transmitter. To detect the sequence index, following correlation test method can be used

$$\mathrm{i}= \underset{i=0\ldots15}{\operatorname{argmax}}\{\boldsymbol{s}_i^{\mathrm{H}} \times \boldsymbol{r}\}$$

where Si is 32x1 vector representing the i^{th} chip sequence defined in the standard (O-QPSK modulated version); r is 32x1 vector representing synchronized incoming I/Q sample sequence. Superscript H means conjugate transpose. The formula correlates input signal with 16 predefined chip sequences, and the one that generates the biggest correlation result implies the transmitted sequence.

The correlation test method above will be unreliable if another time shifted chip sequence is transmitted concurrently by another user. For example, in Figure 22 an interferer signal has relative time difference around 8us (4 chips) in parallel to target signal, and both user transmit chip sequence 5. In this case, not only local chip sequence 5 but also local chip sequence 4 will generate strong correlation value. Because from the beginning time of target sequence, interferer signal segment becomes part of chip sequence 4 although interferer transmits chip sequence 5 as well.

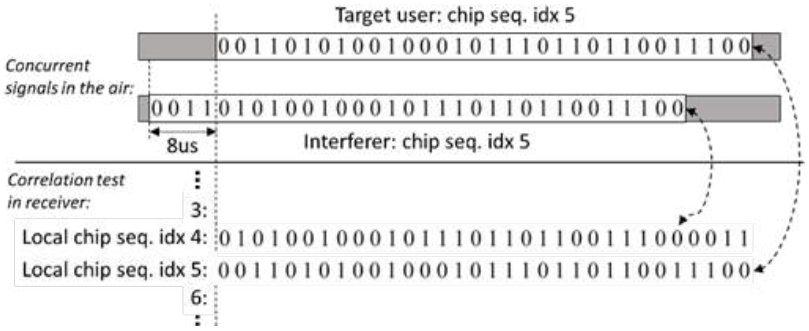

Figure 22 IEEE 802.15.4 interference and correlation test model

In theory, sequence 4's correlation value is weaker than sequence 5's because interferer only gives partial sequence, and target user's sequence 5 can still be identified correctly. In practice, the mistaken probability of correlation test is high, because the correlation value of chip sequence 4 is also big.

3.4.2.2 Antenna switching based NOMA scheme

According to the analysis of the previous section, multi-user signals are Non-Orthogonal because there is only one set of chip sequence in the IEEE 802.15.4 standard and all users use the same sequence set for modulation instead of Multiple Accessing (MA). Unlike in the traditional CDMA system, different users are assigned with different sequences for MA. Because of this, separation of different user signals becomes difficult for IEEE 802.15.4 concurrent transmission. Considering that different users are usually located in different locations in real deployment, an antenna array can be used to capture this spatial character. The different spatial array responses will render the same set of chip sequence to different sets for different users at different locations. Figure 23 shows how to achieve this via switching antennas and a single RF receiver. In our design, antenna switching rate is the same as chip rate. In this example, three users are transmitting concurrently, and multi-user receiver is equipped with 4 switching antennas. There are three separated baseband processing branches to extract bits of three users.

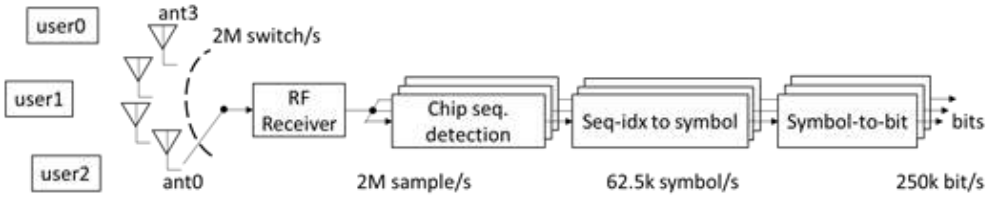

Figure 23 Antenna switching based multi-user receiver

Signal model at chip sequence detection of proposed multi-user receiver is depicted in Figure 24.

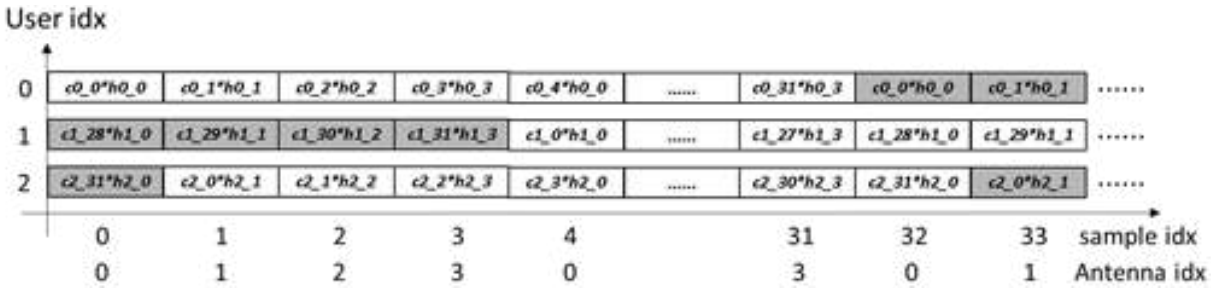

Figure 24 Signal model of multi-user concurrent transmission

Y axis indexes user. X axis indexes received I/Q sample and corresponding antenna (remember that antenna switching rate is the same as baseband chip rate). In general, multi-user transmissions are asynchronous, and different user signals have different time shifts at receiver. Although we draw signal of different users separately with different user indexes in Figure 24 actually they are superposed at the point after RF receiver and before chip sequence detection. The task of chip sequence detection is identifying different chip sequences transmitted by different users. In Figure 24, the series of white block represent a complete chip sequence of a user, and gray blocks are chip from adjacent symbol of the same user. cn_j is I/Q sample of the nth user's the jth chip, where n={0,1,2} and j={0,1,…,31}. hn_k is complex channel gain between the nth user and the kth antenna. n={0,1,2}, k={0,1,2,3}. With this model, the chip sequence detection for user n will be:

$$i = \underset{i=0\ldots15}{\operatorname{argmax}}\{\boldsymbol{s}_{n\_i}^{\mathbf{H}} \times \boldsymbol{r}_n\}$$

Where $\boldsymbol{r_n}$ is 32 x 1 vector:

$$\boldsymbol{r_n} = \begin{bmatrix} r_{q(n)+0} \\ r_{q(n)+1} \\ \vdots \\ r_{q(n)+31} \end{bmatrix}$$

representing the corresponding segment of incoming I/Q sequence of user n. q(n) is the starting I/Q

sample index of a chip sequence for user n. In this example, for user 0 q(0)=0 which means I/Q sample index is 0~31; for user 1 q(1)=4 which means I/Q sample index is 4~35; for user 2 q(2)=1 which means I/Q sample index is 1~32.

So, the segment $\boldsymbol{r_n}$ contains a complete chip sequence from the nth user and interference from other users. $\boldsymbol{s_{n\_i}}$ is not the standard chip sequence anymore, instead it is the antenna switching scrambled version of original chip sequence:

$$\boldsymbol{s_{n\_i}} = \begin{bmatrix} s_{i\_0} * h_{n\_[q(n)+0] mod\ 4} \\ s_{i\_1} * h_{n\_[q(n)+1] mod\ 4} \\ \vdots \\ s_{i\_31} * h_{n\_[q(n)+31] mod\ 4} \end{bmatrix}$$

Where $\boldsymbol{h_{n\_k}}$ is complex channel gain between the nth user and the kth antenna; $\boldsymbol{h_{n\_k}}$ is the unique spatial character captured by switching antenna. This spatial scrambling generates different chip sequence sets for different users; thus it will reduce the multi-user interference which is very strong in the single antenna receiver as analysed before.

By using SIC in NOMA scheme, decoding some users' bits and then subtracting reconstructed signals of these users from the total received signal will improve the detection performance of left users. Figure 25 shows the proposed SIC procedure for 802.15.4 multi-user receiver.

In a SIC iteration, if some decoded packets pass CRC check, corresponding signals will be reconstructed and subtracted. Even if there isn't any decoded packet passing CRC check, we propose reconstructing signal based on bits decoded from signal that has strong correlation value, because that implies high chance to have more correct bits. This is called non-ideal reconstruction. If this happens in an iteration, the subtracted signal will be recovered in the next iteration by the inter-iteration store-restore operation (restore_flag in Figure 25), so that more decoding attempts can be performed on original signal. In this way, the chance of successful decoding is improved.

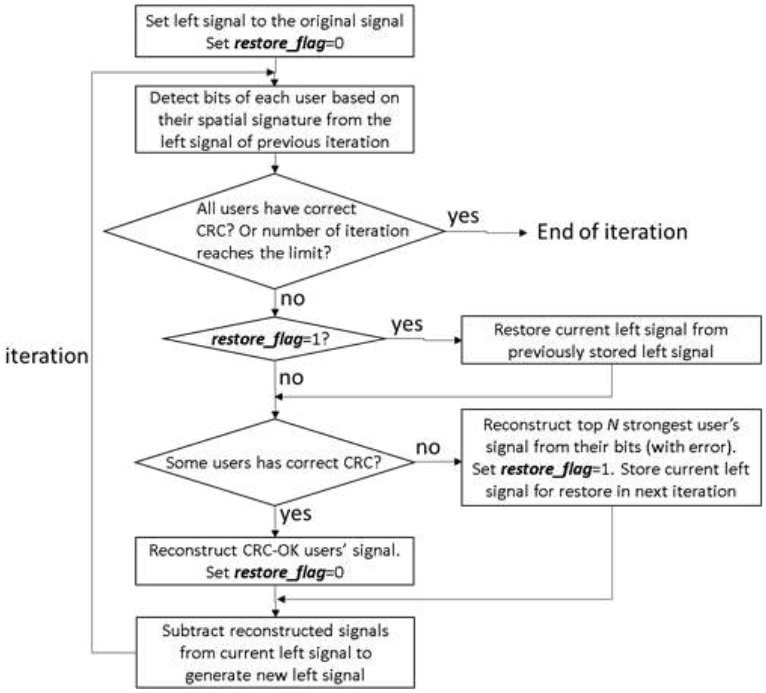

Figure 25 SIC procedure for IEEE 802.15.4 multi-user receiver

3.4.2.3 Simulation results

To evaluate the antenna switching based NOMA SIC algorithm, numerical simulation is carried out involving a model of uniform circular array composed by 8 antennas. Corresponding simulation topology is shown in Figure 26. Multiple users are located at specific directions with equal angle spacing. Line of Sight (LoS) propagation and far-field plane wave model are used to calculate the multiple complex channel gains. The arrival time of signals from different users are distributed randomly in the time interval of 16us (time duration of chip sequence) to model the asynchronous behaviour of concurrent users. For the best performance, parameter N ("Reconstruct top N strongest …" in Figure 25) is set to 2 in our case. The maximum number of iterations is set to the number of concurrent users. Each simulation result is the averaging of multiple simulation results on direction angle 0, 9, 18, …, 45 degrees. Thanks to the symmetric array, 0~360 degree ergodic tests are not necessary.

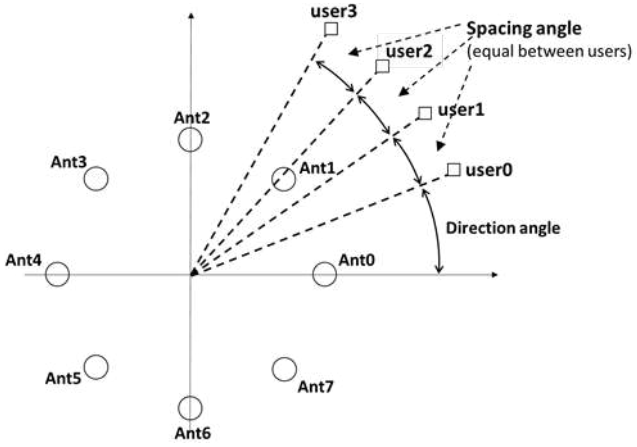

Figure 26 Simulation setup of antenna array and multiple users

Figure 27 and Figure 28 give the simulated PER results under different array diameter, number of user, spacing angle, detection method and power differences. In the result figures, "1 seq" or "2 seq" refers to single chip sequence detection or dual chip sequence detection accordingly; "XXdB power diff" means that received multi-user signal strengths (relative level) is distributed randomly between 0dB to XXdB. Larger dB value means bigger chance to have a stronger user for the successful decoding in the 1st SIC iteration. Zero dB means the worst case where all users give the same received power at multi-user receiver. This is confirmed in the simulation result figures: bigger user power difference supports more concurrent users. Five users can transmit concurrently achieving 1% PER under the worst case of 0dB power difference. When array diameter is 16cm, spacing between user needs to be not less than 20 degrees. With a 32cm diameter array, spacing angel can be as small as 10 degrees.

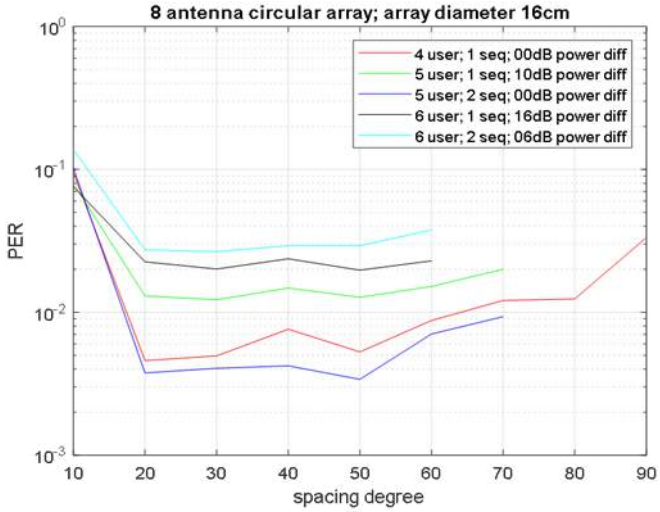

Figure 27 Average simulation result of 16 cm diameter array

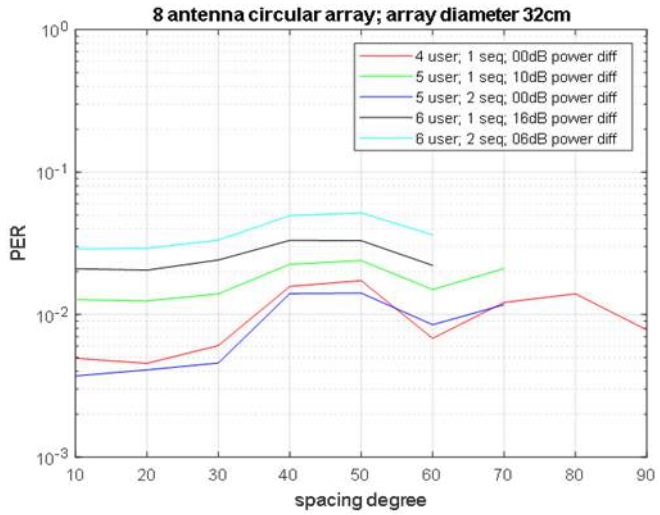

Figure 28 Averaged simulation result of 32 cm diameter array

3.4.3 Testbed Integration

This work is currently only verified in simulation, there is no matching deployment in testbed, also the work requires multi-antenna setup that are not yet available. In future if needed, IoT gateway designed on an SDR can be equipped with multiple antennas, and the algorithm presented in this section can be applied.

3.5 Dynamic padding of I/Q streams to ensure isolation between virtual radios over unreliable connections

3.5.1 Motivation

The radio virtualisation mechanisms of technology-agnostic radio hypervisors are ultimately multiplexing operations, where each virtual radio has access to a fraction of the overall radio resources of the physical radio. Current technology-agnostic radio hypervisors employ frequency-domain radio virtualisation mechanisms, which partition the bandwidth of a physical radio's RF front-end (B) into spectrum chunks, abstracted to the virtual radios in the form of the virtual bandwidth (b_i) of their virtual

RF front-end, i = 1, ... , N. The virtual radios interact with their virtual RF front-end as they would with a real RF front-end, sending/receiving streams of IQ samples. The radio virtualisation mechanism ensures isolation between the virtual radios at the radio resources level, using non-overlapping radio resources and including the necessary guard bands and/or intervals. However, due to the synchronicity of the multiplex operation, virtual radios that do not produce or consume the appropriate number of IQ samples at the precise timing may compromise the operation of the entire system.

3.5.2 Implementation Results

The multiplexing of the IQ samples generated by multiple virtual radios occurs on a per-window basis, where the radio hypervisor collects a certain number of IQ samples (W) from/to the virtual radios over a time period (τ_{vv}) of duration equal to the ratio between W and the sampling rate of the physical radio (f_s), i.e., $\tau_{vv} = W/f_s$. Only when in possession of W IQ samples, the radio hypervisor is able to multiplex/demultiplex the IQ samples from/to multiple virtual radios. For frequency-domain radio virtualisation mechanisms, each virtual radio is expected to produce/consume a given number of IQ samples (w_i) within τ_{vv} , corresponding to a fraction of W equal to the ratio between its virtual bandwidth b_i and the total bandwidth B:

$$w_i = W \cdot \left\lceil \frac{b_i}{B} \right\rceil .$$

On a RANaaS platform, the virtual radios send/receive streams of IQ samples from/to different tenants. It is likely that the virtual radios will not always consume/produce the exact number of w_i IQ samples during every multiplex window, i.e., generate samples at an expected nominal rate $r_i = w_i/\tau_{vv}$. Instead, during multiplex window j, virtual radio i will present an effective rate ($r\_eff_{i,j}$), which may vary for each window and can be inferior to the expected nominal rate ($r\_eff_{i,j} \leq r_i$). Such effective rate may be the result of (i) the multiple access scheme of RATs which intermittently transmits frames, e.g., WiFi and Bluetooth; (ii) the behaviour of software radios and baseband units which cannot process IQ samples at the expected nominal rate; or (iii) RAN slices that stopped operating but were not yet de-allocated from the RANaaS platform.

To prevent individual virtual radios from affecting the processing performance of the entire system, we must isolate the processing of their streams of IQ samples and ensure that the appropriate number of W IQ samples are available to the radio hypervisor at every τ_{vv} . We can achieve such isolation through dynamic padding of the streams of IQ samples, i.e., with the inclusion of a number of null samples ($v_{i,j}$) to the stream of the i-th virtual radio during the j-th multiplex window whenever $r\_eff_{i,j} < r_i$:

$$v_{i,j} = w_i - r_{\mathrm{eff}_{i,j}} \cdot \tau_W = w_i - \frac{w_i \cdot r_{\mathrm{eff}_{i,j}}}{r_i} = w_i \cdot \left(\frac{r_i - r_{\mathrm{eff}_{i,j}}}{r_i} \right)$$

The dynamic padding limits the effect of a virtual radio presenting a $r\_eff_{i,j} \leq r_i$ to its own RAN slice, without delaying the production/consumption of multiplex windows. If such padding were not added, then the radio hypervisor would need to wait until a full window of W IQ samples was available. This implies that the virtual radios would not be independent in terms of the processing of their streams of IQ samples, i.e., the virtual radios would not be isolated at the radio processing level. The impact of this delay would accumulate over successive windows, deteriorating the performance experience of all virtual radios over time, as illustrated in Figure 29, where we show a numerical analysis where we consider a constant $r\_eff_{i,j}$ for every window. Through the application of the dynamic padding of IQ samples, such delay is nullified, which allows virtual radios to operate independently of one another regardless of the number of produced and/or consumed IQ samples.

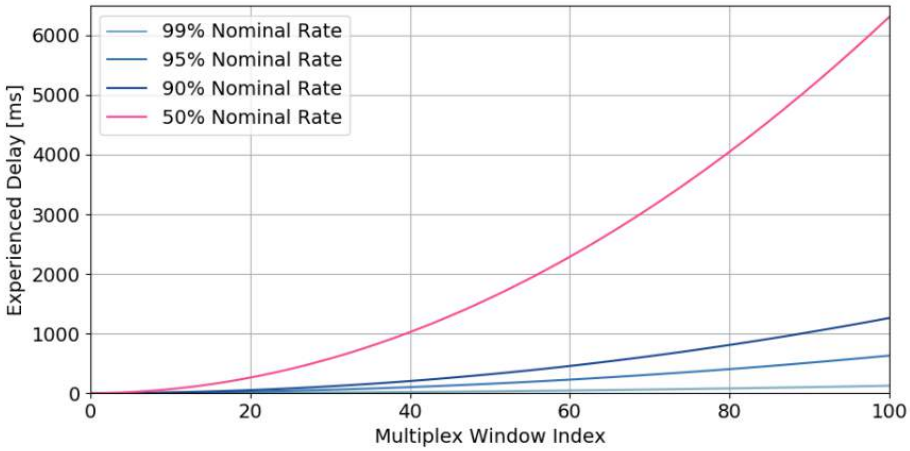

Figure 29 How different effective rates would affect the experienced delay over time, if dynamic padding were not applied. In this scenario, we have a radio hypervisor at f_s = 200 KHz and W = 1000 samples, with two virtual radios at r_i = 100 KHz and w_i = 500 samples, while one of the virtual radios manifests an effective rate shown as a percentage of the expected nominal rate

3.5.3 Testbed Integration

We have integrated our dynamic padding of IQ samples with HyDRA, which not only allowed the virtual radios to be independent in terms of their radio processing, but also enabled HyDRA to instantiate and terminate virtual radios on the fly. This effort led do a journal extension of a previous IEEE ICC paper, submitted to the IEEE Transactions on Network and Service Management (TNSM). In addition, the current state of the implementation can be found on the following public repository: https://github.com/maiconkist/gr-hydra.

4 HIGHER LAYER SDR INTEGRATION

4.1 TAISC MAC engine extension to support multiple PHY layer instances

4.1.1 Motivation

The original TAISC concepts only took into account a single PHY and MAC layer to perform medium activity. This concept was extended in order to (on the fly) install multiple MAC protocols, still only using a single PHY layer. Therefore, only a single MAC protocol could be active at each given moment. Through the addition of the SDR research track, it is now possible to transmit/receive on 8 different channels simultaneously on a single device, TAISC needs to be extended in order to fully make use of the new features offered by PHY.

4.1.2 Implementation Results

The original architecture of TAISC is given in Figure 30 . Multiple MAC protocols are installed as radio binaries, but only a single PHY instance is present.

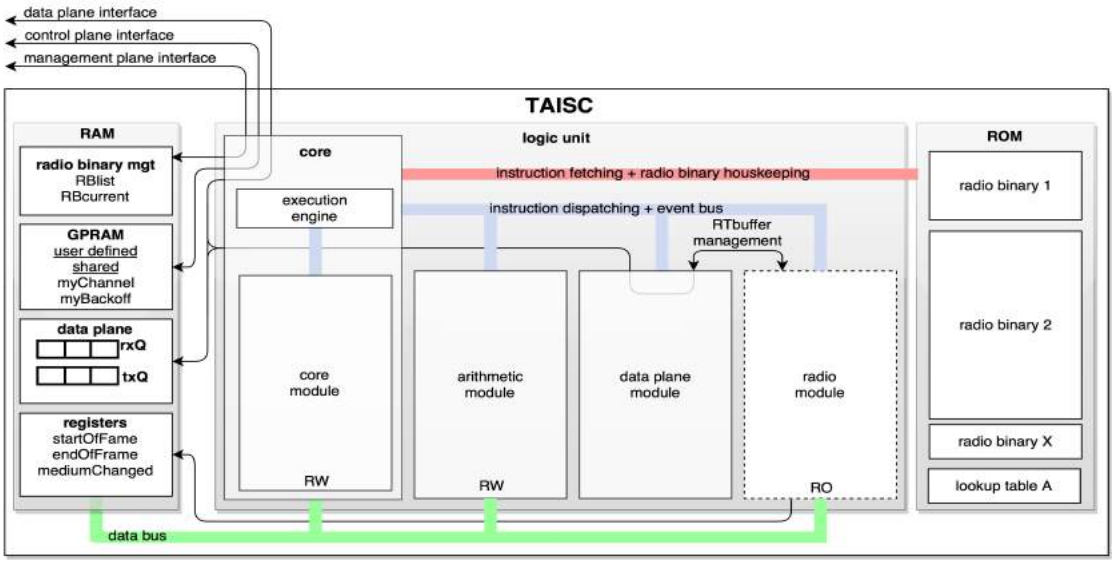

Figure 30 Original TAISC concept, multiple MAC protocols (radio binaries) are installed but only one PHY instance

The modification of TAISC is intended to use of the multi-channel virtual transceiver, by registering each virtual transceiver to a separate radio module under a more generic “SDR radio module”. This is necessary since the virtual modules still share some logic between them. When performing medium access the MAC developer should select which interface needs to be configured for either TX or RX capabilities, the new architecture is illustrated in Figure 31.

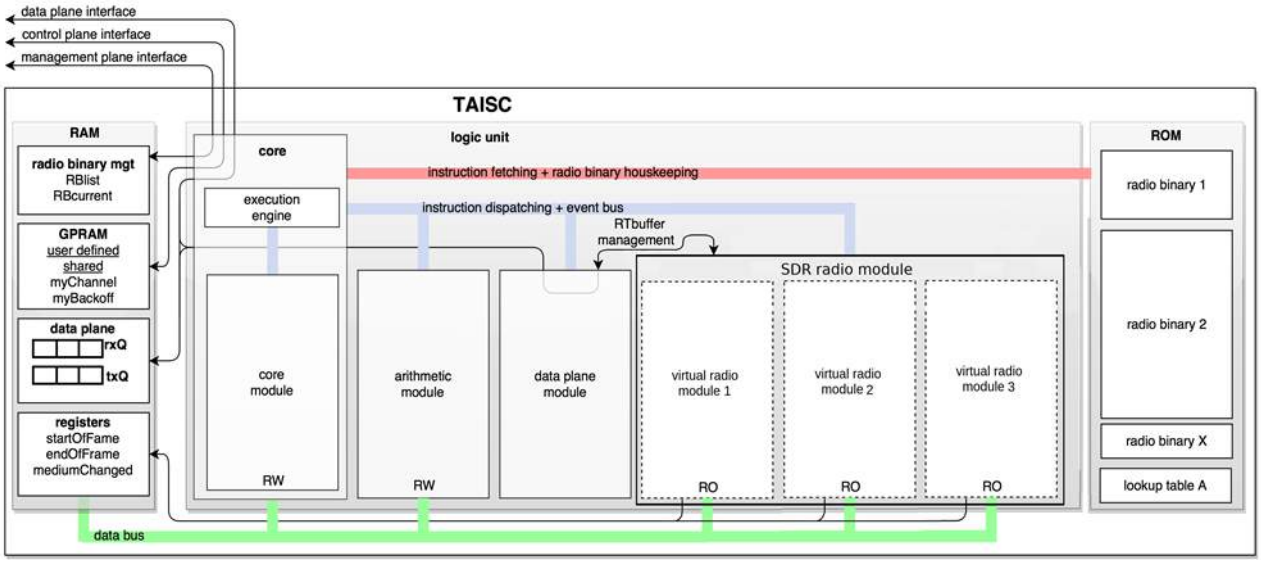

Figure 31 The new TAISC architecture, allows the instantiation of multiple PHY instances as virtual radio modules under a single SDR radio module

For the TAISC core, there weren't any deep changes required since these new modules are registered the same way as any other modules. The development effort was split into two tasks: (i) creating the new modules, and (ii) create a novel MAC protocol which can make use of the multi-channel transceiver. The new radio modules make use of the SDR PHY programming interface and wraps the API instructions into time-annotated TAISC instructions. To transmit data, a MAC developer can use the following flow:

_setChannel(R1,X); // Select virtual interface x (in range 0 - 8)

_loadFrame(R1,0,currentTxFrame); // Load the frame into PHY layer x

_tx(R1,0); // Transmit the frame

In order to send data on multiple virtual interfaces simultaneously, the MAC developer should first load the to be transmitted data on a per interface basis, and afterwards give a single transmission signal:

_setChannel(R1, A); // Select virtual interface A (in range 0 - 8)

_loadFrame(R1,0,currentTxFrame_A); // Load the frame into PHY layer A

_setChannel(R1, B); // Select virtual interface B (in range 0 – 8)

_loadFrame(R1,0,currentTxFrame_B); // Load the frame into PHY layer B

...

_setChannel(R1, N); // Select virtual interface N (in range 0 - 8)

_loadFrame(R1,0,currentTxFrame_N); // Load the frame into PHY layer N

_tx(R1,0); // Transmit the frame on interfaces A,B,...,N

For now if the MAC developer chooses to listen by issuing the `rx` command, it will start listening for incoming packets on all interfaces simultaneously. Incoming packets are stored into the internal SDR PHY buffer and can be collected by TAISC following the default receive flow:

```
if(waitForTrigger(1, _TAISC_EVENT_radio_rxFrameAvailable(R1))){
        // A
        _grabFrame(R1,0);

        if(waitForTrigger(1, TAISC_EVENT_dataplane_rxFrameAvailable)){
                rxTrigger(1);
        } else {
                rxTrigger(0);
        }
}
```

To test these concepts a novel TDMA protocol was created which is able to transmit or receive data on multiple channels simultaneously in a single TDMA time slot. This allows several new use cases, e.g. send an alarm signal to multiple networks on different channels, with minimal latency.

4.1.3 Testbed Integration

The result of this work consists of a single TDMA protocol using multiple PHY's and/or channels of the multi-channel virtual transceiver. This is offered in the w-ilab.t portable testbed. In the future, the implementation be further optimized to have fully independent MAC/PHY combinations running in parallel. This has a higher complexity since MAC protocols can interrupt one another, and possibly break time-sensitive MAC level operations.

A current research track is comparing several approaches to either safely and fairly divide CPU resources between MAC/PHY combinations, or to have multiple CPU's or CPU cores each running a dedicated TAISC instance.

4.2 Computational resource scaling with Linux containers based on the traffic requirement of network slices

4.2.1 Motivation

Network slicing in end-to-end Telecom networks largely involves Cloud-RAN (C-RAN) technology, which is a mobile network architecture where the Baseband Units (BBUs) of a group of base stations are separated from the Remote Radio Heads (RRHs), and processing is moved to a centralised BBU pool in a cloud-based computing centre. Network Function Virtualization (NFV) can be implemented at the C-RAN central office, with the support of dynamic resource allocation. This enables the flexible assignment of different amounts of computational resources to different BBUs. In this way, computational resources for each BBU can be tailored to the requirements of applications in real time, in order to optimize overall resource utilisation. It is hard to achieve this dynamic allocation with Application-specific Integrated Circuits (ASICs). Therefore, more flexible BBU implementations in software have emerged to bring flexibility and dynamicity. Software-Defined Radio (SDR) enables the softwarization of the BBUs and allow Linux containers in the cloud to be used to host these BBUs.

We design and implement an SDR-based prototype to dynamically allocate computational resources in a Linux-container based BBU pool for C-RAN. The objective is to demonstrate its benefit in terms of

an increase in the performance of high throughput applications, e.g. High Definition (HD) video streaming. We use LXD container technology to achieve NFV on the Linux-based BBU-pool system.

4.2.2 Implementation Results

We use a quad-core ubuntu Virtual Machine where LXD containers are hosted to implement CPU resource scaling. We use the lxc-cgroup API for dynamic CPU allocation. This API can manage the number of assigned CPUs, limit the memory usage and the access to physical system devices through terminal commands.

As the first step we simulate CPU loads using a tool called “stress-ng”. With this tool we emulate the CPU load to be an approximation of a process that receives and elaborates data, such as the baseband processing of BBUs. The tool “stress-ng” can accurately load a particular number of CPUs for a given percentage or emulate certain usage of the container memory and I/O operations. CPU scheduler also plays a key role in our experiments. By default, Linux kernel uses the Completely Fair Scheduler (CFS), which has a load-balancing policy that tries to assign the same CPU time to all processes.

In the first experiment we run one LXD container and initially assign it with only one CPU. After the container has started, we spawn two processes which both consume 90% of the CPU time, with “stress-ng”. As expected, the CPU is overloaded to 100%, as shown in Figure 32 (by running “htop” command). Therefore the CPU time of each process is squeezed to be 50%. Much lower than the normal CPU time (90%). Depending on the type of processes, we expect that the usage of a lower CPU time than normal leads to instability of the applications running in the container. At this point we assign a new CPU and, since the Linux scheduler has a load-balancing policy, the two processes are assigned to the two different CPUs separately. Figure 33 shows the screenshot of the “htop” command inside the container after the additional CPU allocation. The results show that the two CPUs now both have around 90% usage.

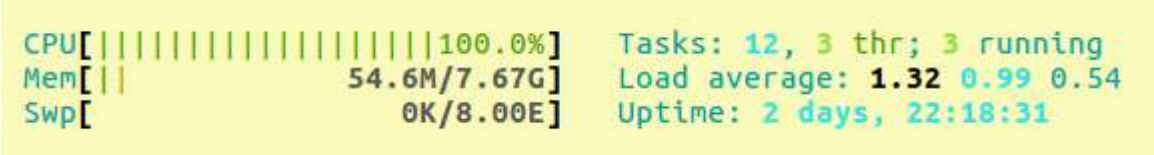

Figure 32 CPU utilization when one CPU is running two 90% processes

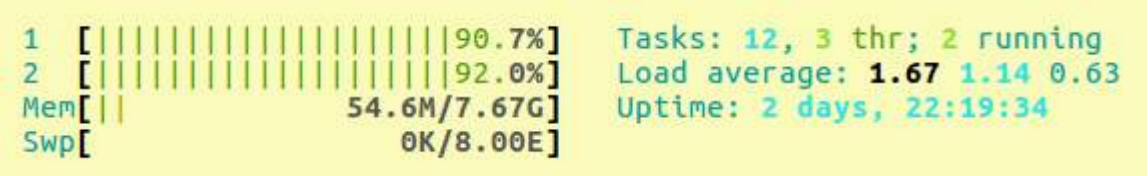

Figure 33 CPU utilization when two CPUs are running two 90% processes

We also test the switching time of the Linux container system when applying the additional CPU resource allocation. We build our Linux containers on Dell PowerEdge R440 servers with Ubuntu 18.04 installed. Table 4 shows the average time to allocate new CPUs in 3 different scenarios (i.e., adding 1, 2, or 3 CPUs at a time). These time periods are very short (around 12 ms), no matter how many CPUs are added at a time. This means a very fast CPU scaling system can be applied to a communication network. The short switching time 12ms allows real-time reconfiguration with minimum interruption to the existing network traffic.

Number of CPUs added	Time (ms)
1	12.029
2	12.033
3	12.060

Table 4 Dynamic CPU allocation time

In the second experiment, we install two SDR-based LTE eNodeBs with srsLTE software to one LXD container and then stream an HD video through the air-interface to two UEs. When the experiment starts at Time 0, the container is allocated only one CPU, and the MCS index of the LTE eNodeBs is increased every 20 seconds, meaning that the computational load of the container is increased every 20 seconds. At the 80th second one more CPU is assigned to this container. Figure 34 shows the performance of frame dropping of the video transmission over the time. It can be seen that from the 20th seconds the frame dropping becomes non-zero due to the overload of the CPU, while on the 80th second it becomes stable (no new frame is dropped) again when we add one more CPU to the container. Therefore, the performance of the SDR system has been improved when one more CPU is added to the system.

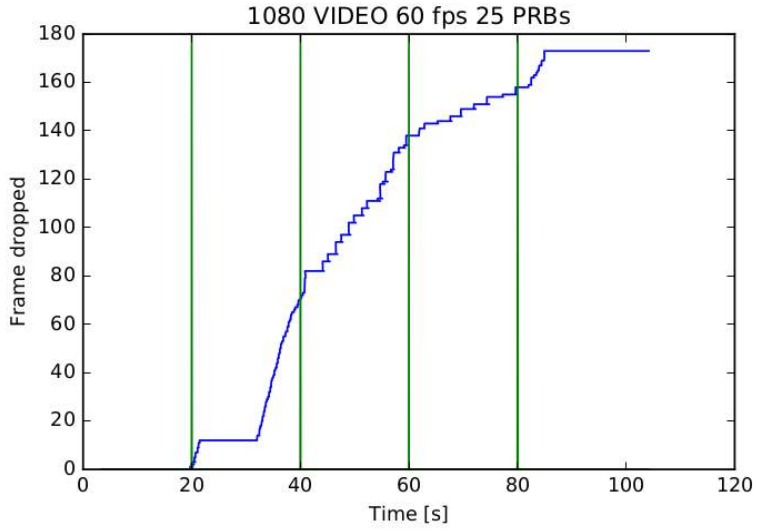

Figure 34 Frame dropped over time when transmitting HD video through SDR-based eNodeB

4.2.3 Testbed Integration

This work has been implemented based on TCD's Iris SDR testbed where all the baseband processing is located in virtual machines in a cloud-based server cluster. Linux containers are implemented in the virtual machines in the cloud. OpenStack is installed to manage the virtual machines. The aforementioned computational resource scaling technology helps improving the performance of the SDR and cloud system when more resource is needed as per the demands. On the other hand, the dynamic resource scaling technology is also capable to help saving energy when not so many CPU resources are actually needed if the traffic demand is low. Reducing the number of CPUs assigned in real time saves the energy consumption of the system.

Besides, this technology has been extended by ORCA open-call project ELASTIC. The project is successfully completed and reported to the ORCA consortium. The ELASTIC project implements an auto-scaling method to add and reduce CPU resources in an automatic way and tests the performance remotely through Fed4FIRE tool on our Iris SDR testbed.

4.3 5G GFDM PHY integration into Multi-RAT platform

4.3.1 Motivation

The goal in Year 3 was to further enhance and complete the NI Multi-RAT experimentation platform with a real-time 5G path supporting flexible numerologies. With this, 5G-LTE interworking experiments are possible using the dual connectivity (DC) functionality. The overview of the final Multi-RAT platform is shown in Figure 35.

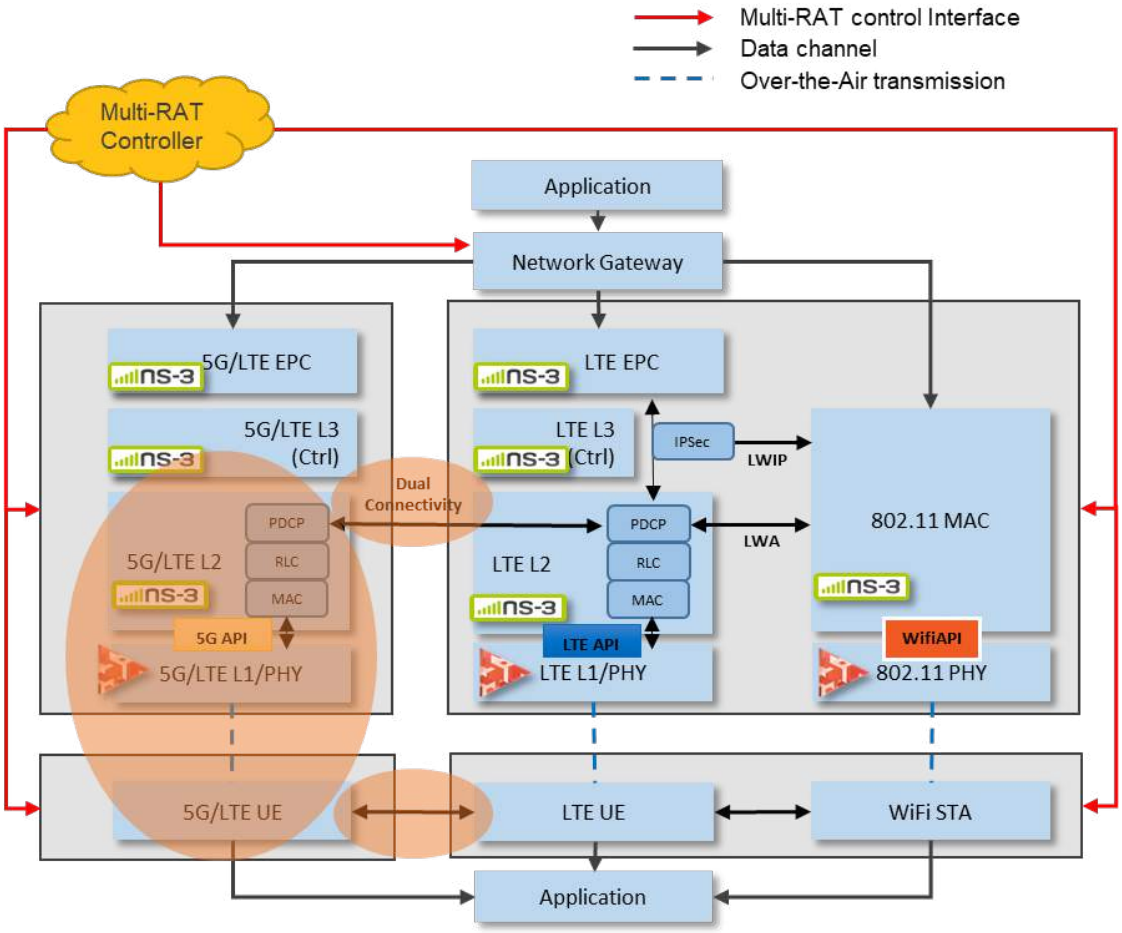

Figure 35 Overview of the final Multi-RAT platform

To showcase the capabilities of this platform the setup will be extended by an industrial robot control application including RAT control and monitoring functionality. This will be demonstrated in the final implementation of showcase 4, see D2.5 [2].

4.3.2 Implementation results

In collaboration with Technical University Dresden (TUD) in ORCA Year 3 the 5G path for the Multi-RAT platform was established consisting of the following three main components (visualized in Figure 36):

- ns-3 upper layer protocol stack (green)
- NI L1-L2 API for 5G path (blue)
- GFDM flexible physical layer supporting 5G flexible numerology (red)

Figure 36 shows these components integrated to a full stack 5G path including ORCA partner responsibilities. The GFDM Transceiver enhancements towards 5G are already described in section 3.2. The subsequent sections will focus on the interfacing of the physical layer via NI L1-L2 API [7] and the ns-3 [13] upper layer protocol stack integration and adaption.

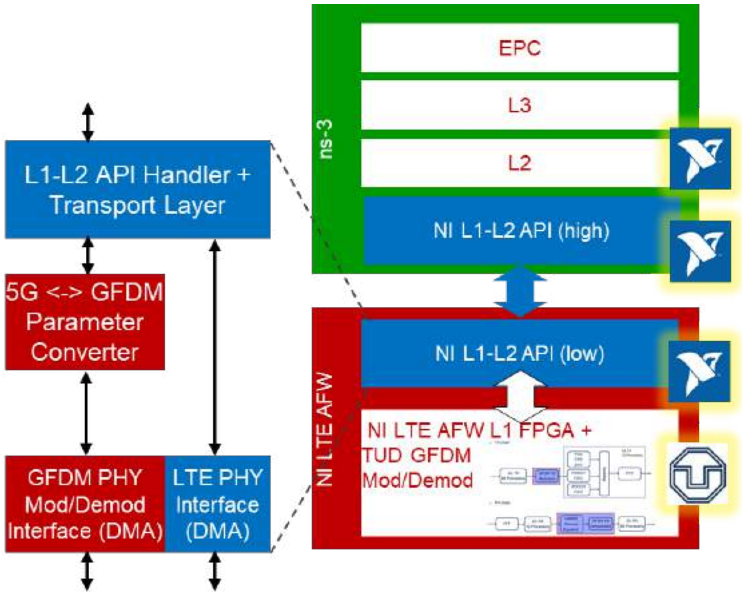

Figure 36 5G GFDM integration approach

4.3.2.1 NI L1-L2 API for 5G

The current physical layer implementation (PHY) supporting flexible numerology based on the NI LTE Application framework (AFW) 2.5 [8] was extended with modules for encoding/decoding based on GFDM principles. Figure 37 shows a high-level block diagram of the physical layer modules including GFDM extensions (blue).

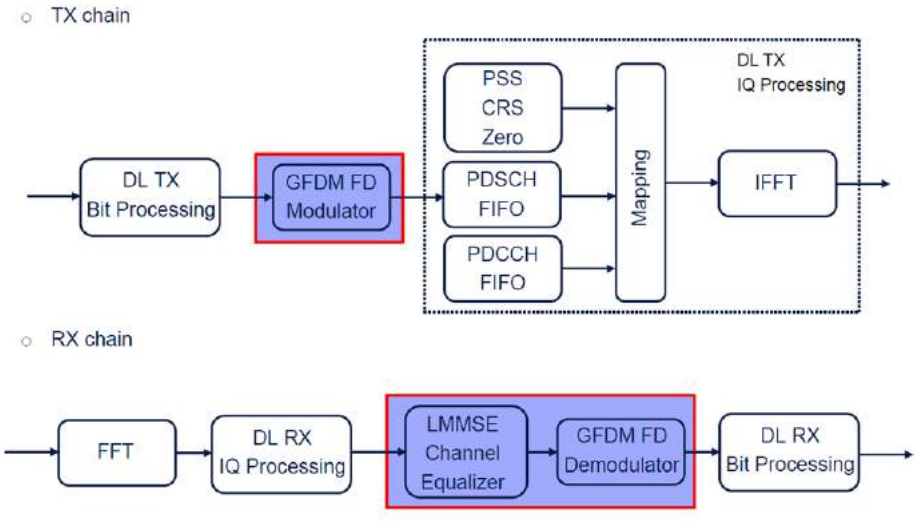

Figure 37 GFDM Implementation diagram. Blue denotes changes/modules implemented

Given this construction for the physical layer, the design concept for the L1-L2 API is to have the original LTE functionality described in D3.3 [9] plus extensions related to the application of 5G flexible numerology which are described in this deliverable. Figure 48 provides the general idea of the messaging architecture.

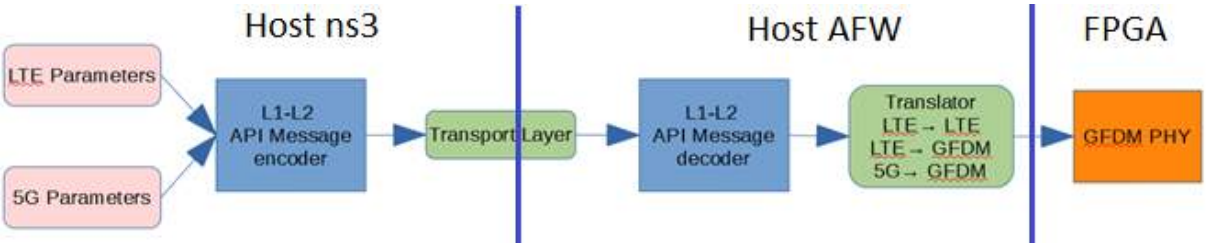

Figure 38 Message architecture implementation diagram

It is visible from the previous diagram that the L1-L2 API has 2 different chains that merge before the transport layer in ns-3. In general, the API will generate the LTE messages (this API was not modified and thus it is operational independent of the 5G API part) as well as the 5G related messages, both will be written into the transport layer, which in this case are Linux named pipes, after the proper processing and serialization.

Through the transport layer, the messages will arrive in the NI LTE Application framework (AFW) and are decoded one at a time on the RT CPU (Host). If there is any 5G message, it will be sent and received first, followed by any relevant LTE-related messages, depending on the case that is being evaluated. The intention of following this messaging scheme is to not interfere with the LTE API messages and the LTE implementation. Based on the information of the received LTE/5G API messages the LTE/GFDM FPGA configuration will be generated using a translation unit (Translator or Parameter Converter) which transfers 5G parameter to GFDM PHY parameters e.g. K and M.

If the L1-L2 API is to be used in 5G/GFDM mode, the operation stays the same as for LTE, but with some key extensions. To provide LTE backward compatibility, the LTE messages are not modified and remain the same as described in ORCA D3.3 [9], but four new messages are introduced to support 5G flexible numerology in uplink (UL) and downlink (DL):

- PHY_5G_DL_TX_CONFIG_REQ
- PHY_5G_DL_RX_CONFIG_REQ
- PHY_5G_UL_TX_CONFIG_REQ
- PHY_5G_UL_RX_CONFIG_REQ

Each of them configures the specific part of the communication link, and each shares the same sub message 5G_TX_CONFIG. Hence the complete L1-L2 API message definition was extended as described in the subsequent figure.

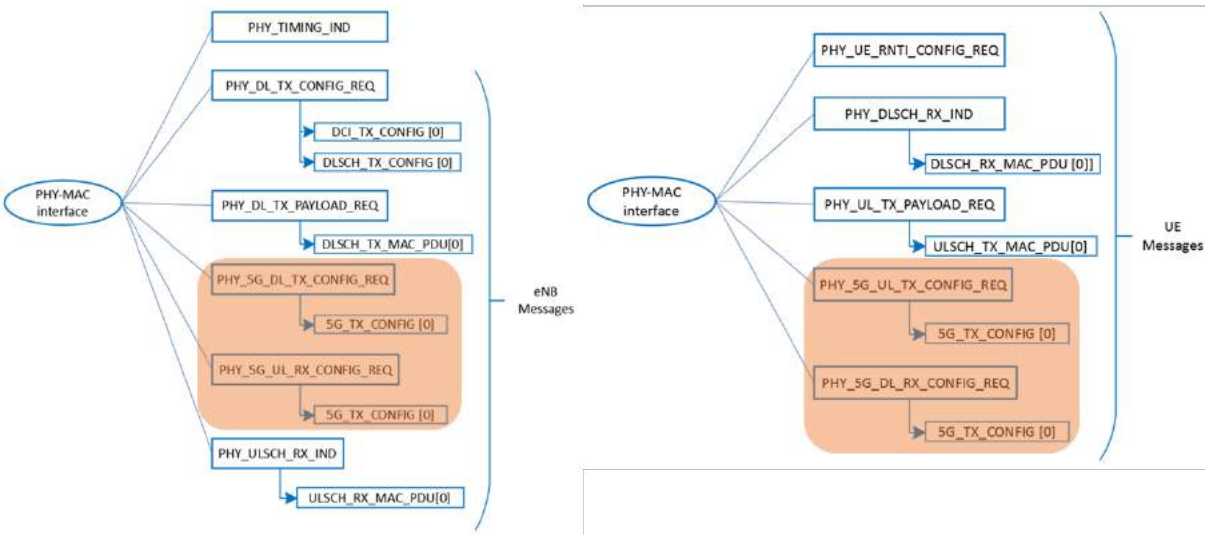

Figure 39 L1-L2 API messaging scheme for eNB and UE, 5G enhancements highlighted

We see that this new message structure keeps the original LTE structure intact and thus should become completely compatible with it, only needing to have the additional messages disabled (orange). This allows for an additional level of freedom when looking at how to implement the message itself. While permitting the message to be extended such that different 5G-NR PHY parameters can be added as they become necessary, the structure is also transparent to the underlying PHY layer. Those parameters are encapsulated by message PHY_5G_##_##_CONFIG_REQ with sub message 5G_TX_CONFIG_REQ which is described in detail in Table 5:

PHY_5G_##_##_CONFIG_REQ	message type ID	U16	0x4###
	reference ID	U16	Linear Counter
	instance ID	U8	0
	msg. body length	U24	8
	SFN	U16	{0…1023}
	TTI	U8	{0…9, 0…19}
	#of sub-messages	U8	1
	Confirmed mode	U8	{0, 1}
	Reserved	U24	0
	For any number of sub messages this applies		
	Sub-msg. type	U8	{0} - Since there is only one defined
	Parameter set ID	U8	{0,1)
	Parameter set body length	U24	11
	Parameter set body	<length>	
5G_TX_CONFIG	sub message of PHY_5G_##_##_CONFIG_REQ		

	SCS	**U32**	**0: 15kHz, 1: 30kHz, 2: 60kHz, 3: 120kHz**
	Reserved	U56	This space is 7 bytes. For 32-byte word alignment.

Table 5 5G message for the configuration of the GFDM PHY

Please note, this new sub message is intended for the current implementation of the GFDM 5G PHY layer provided by TUD. As such, it was defined that the only modifiable parameter of this PHY layer implementation is the subcarrier spacing (SCS). This message can be easily expanded to contain more 5G parameters as the PHY layer becomes more flexible and complex, or a different 5G PHY will be integrated.

As already mentioned, the GFDM PHY requires parameters K and M to apply different SCS configurations. The definition of K, M and other GFDM related parameters can be found in [10]. The subsequent table shows how the L1-L2 SCS value maps to K and M:

SCS in kHz	**μ (3GPP 38.211)**	**SCS L1-L2 API value**	**GFDM K, M**
15	0	0	K=64, M=1
30	1	1	K=32, M=2
60	2	2	K=16, M=4
120	3	3	K=8, M=8
240	4	Not supported	Not supported

Table 6 Parameter mapping for different subcarrier spacings

To support the configuration of a variable SCS provided by the 5G GFDM PHY the aforementioned NI L1-L2 API extensions have been implemented into the lower part of the NI L1-L2 API part, Figure 36. To realize this new message decoders, message handlers and parameter conversion units have been developed and integrated into the LV-based PHY layer and PHY service access point (SAP) implementation. As an example, the uplink Tx configuration on UE is shown in Figure 40. Blocks with the green/turquoise header are related to the 5G GFDM configuration.

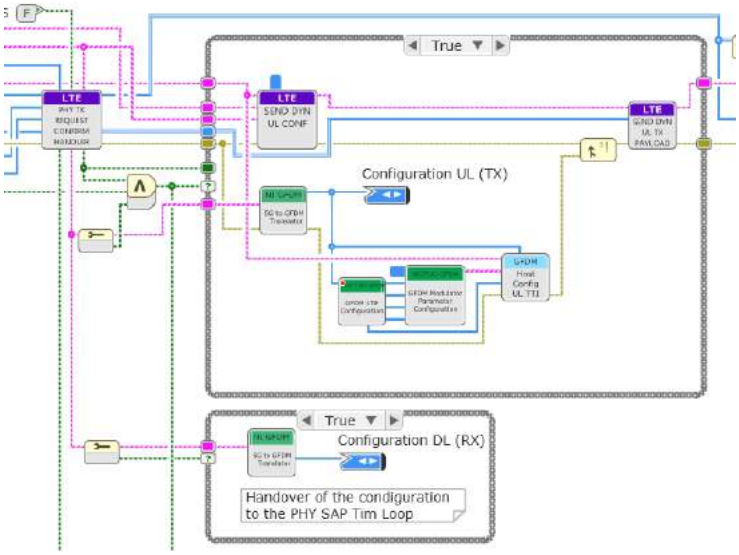

Figure 40 NI L1-L2 API low part for UL Tx configuration on UE

4.3.2.2 ns-3 upper layer protocol stack

As described earlier the goal of the API implementation is to connect physical layers like the NI LTE Application Framework PHY also with external open source stacks providing the MAC and higher layer functionality. For the Multi-RAT platform, the ns-3 simulator is used as it includes a rich set of modules that provides functionalities for testing wireless communication system, e.g. LTE and WiFi. In D3.1 a basic description can be found how the NI API is implemented in ns-3 WiFi module to connect to the NI 802.11 Application Framework [11]. In D3.3 a description of the API implementation for the ns-3 LTE module towards NI LTE Application Framework [8] can be found.

Since 5G is an evolution based on LTE, the ns-3 part for the 5G API extension is based on the ns-3 LTE API developed in ORCA Year 2. The defined goal for Year 3 to allow the ns-3 stack to connect also to other PHY modules using the same API e.g. the TU Dresden 5G Transceiver is realized with the enhancement of the whole Multi-RAT platform towards 5G as outlined in the motivation section. To realize this, we evaluated two different options for the higher layers.

- 5G-LENA ns-3 module
- LTE ns-3 module

5G-LENA is a New Radio (NR) network simulator developed by CTTC, designed as a pluggable module to ns-3. The simulator is the natural evolution of LENA, the LTE/EPC Network Simulator. The development started from the mmWave module created by NYU, and it incorporates fundamental PHY-MAC NR features aligned with NR Release 15 TS 38.300. The first release of the simulator is dated February 2019 [12]. An overview about the 5G-LENA system architecture is visualized in Figure 41.

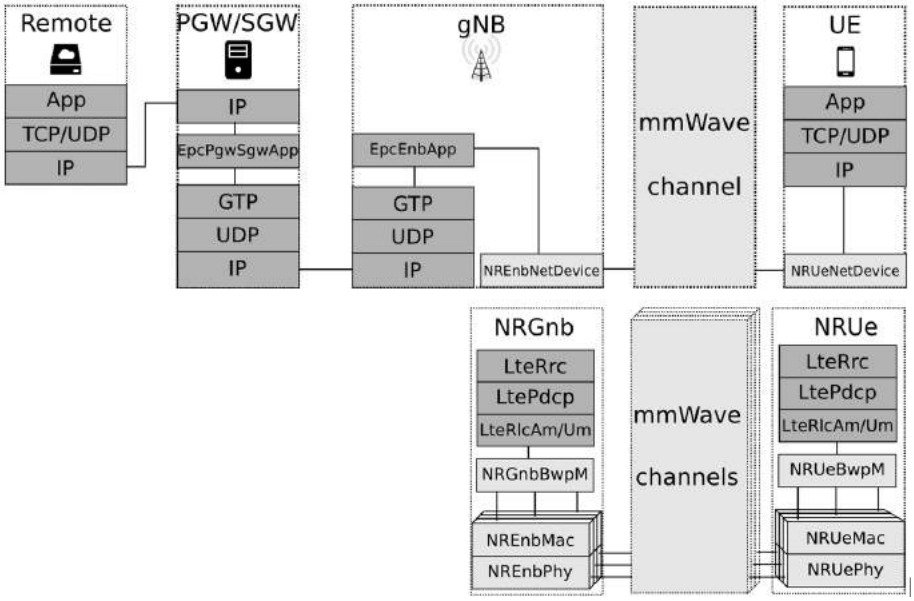

Figure 41 ns-3 5G-LENA overview (https://5g-lena.cttc.es/features/).

During analysis of the 5G-LENA code base, it turned out that RRC Real protocol which is important for separated gNB and UE instances was not implemented. 5G-LENA is still in development phase. That's why some higher layer features are missing. To clarify, RRC Real protocol encapsulates the RRC messages to an PDCP SDU, which will be forwarded through the entire protocol stack. The implemented RRC Ideal protocol doesn't support this functionality and makes use of function calls instead. Within ORCA Year 3 we achieved first steps to enable RRC Real which might be a good base line for the ns-3 community to continue.

Another challenge is the NR frame structure. Figure 42 compares the LTE with the NR frame structure in time domain. Further respective ns-3 function calls are outline in order to understand complexity of the two different implementations.

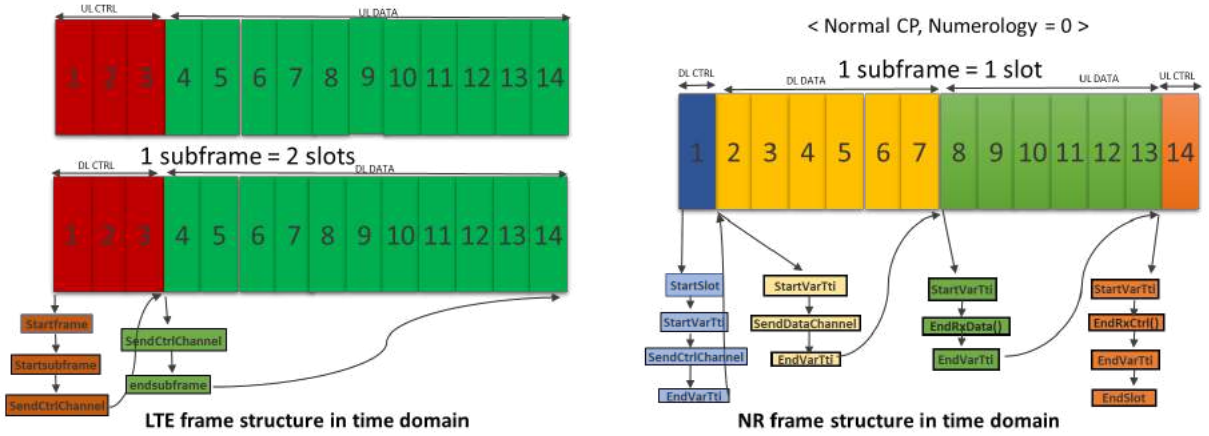

Figure 42 LTE vs. NR frame structure

Since the 5G GFDM PHY physical layer implementation is based on LTE frame structure, a variable TTI (VarTti, variable Transmission Time Interval) as it is defined in 5G and implemented in 5G-LENA cannot be applied. Further the ns-3 protocol processing on time intervals < 1ms will be a challenge for a 4 Core Intel i7 CPU which is used for this platform including DMA transfers to the physical layer. This may change in the future where more powerful SDR platforms become available.

To mitigate the risk for the entire ORCA Multi-RAT platform, we focused on the second approach using the **LTE ns-3 module** [13] as higher layer protocol stack. Extensions towards the integration of 5G GFDM PHY including mixed numerology with the focus on a runtime configurable subcarrier spacing are implemented as described in the subsequent paragraphs. Another advantage is the re-usability of the dual connectivity (DC) functionality for the 5G path which was developed in ORCA Open Call 2 for extensions.

The definition of the extensions for the NI L1-L2 API was already discussed in section 4.3.2.1. The main contribution here is the implementation of the new 5G message encoders, decoders and handlers into the *NiLtePhyInterface* class. As introduced in D3.3 the *NiLtePhyInterface* class represents the convergence layer between the ns-3 control and payload messages that need to be translated into the L1-L2 API messages using the *NiApiHandler* class. Both implementations are extended towards 5G functionality.

Important for real time functionality is the message sequence and the message timing. As an example, for the downlink configuration at eNodeB, the extended message sequence is described as follows:

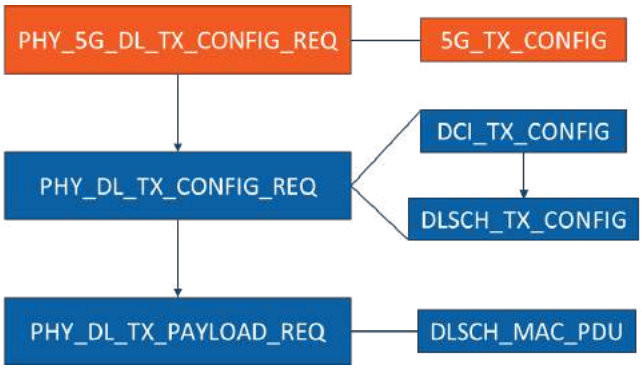

Figure 43 L1-L2 API – Message sequence for 5G DL TX configuration

Meeting real-time requirements is the key to perform successful over-the-air (OTA) end-to-end experiments with real-time FPGA PHY implementations. Here the configuration timing of the physical layer is very important. Thus, the L1-L2 API messages need to be sent from ns-3 upper layers with correct timing and order. The subsequent diagram shows the relation between the achieved ns-3 and PHY-FPGA processing using the NI L1-L2 API to transfer timing and configuration values.

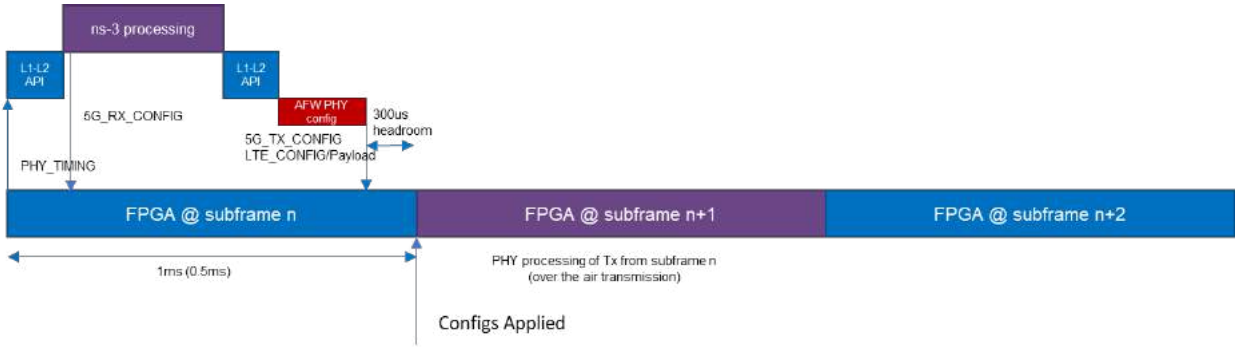

Figure 44 ns-3 and PHY-FPGA configuration timing

Another important change which was implemented into ns-3 LTE MAC layer is a new scheduling algorithm that is executed by the MAC scheduler module. The MAC scheduler assigns physical resources to UEs and was adapted for the 5G GFDM PHY. The implementation which is currently used for LTE executes a round robin (rr) algorithm. With the 5G GFDM implementation, there is an additional challenge in the resource allocation since the encoder can only handle multiples of 4 for the resource block group (RBG) allocations. Additionally, the middle 3 physical resource blocks (PRBs) which are overlapping with the primary synchronization sequence (PSS) can't be allocated otherwise it would corrupt the communication link, as a result it would not recover from synchronization loss. As such, modifications are required in the ns-3 MAC scheduler. In total 80 out of 100 resource blocks can be allocated for the physical layer implementation. A new scheduling algorithm was derived from the original round robin (rr) implementation in order to keep the original ns-3 LTE scheduler operational and unchanged. It assigns the resource block allocation according to the 5G GFDM PHY requirements dynamically. With that a reliable operation is ensured.

As a proof of operation, SCS switching was performed during runtime which relates to a bandwidth part (BWP) adaptation defined in 3GPP. The results are shown in Figure 45. It can be seen that only one physical transport block was not decoded successfully during the SCS switching operation because of re-configuration of the physical layer. This leads to an application level packet loss of two packets which is acceptable for an experimentation platform because it's a packet loss rate of 0.0001 % over the complete run-time.

Level	Packet loss w/o SCS switching	Packet loss w/ one-time SCS switching
High Level (APP Pkt Lost)	1	2
Low Level (PHY CRC Fails)	0	1
Negative Confirmations	0	0

Ns-3 packet configuration:

- 20000 packets
- 3 ms interval
- 1500-byte size
- SCS =1 (K=32,M=2)
- Switch to SCS=2 (K=16, M=4)

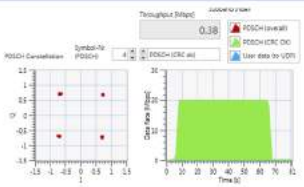

Figure 45 5G SCS runtime switching

4.3.2.3 Multi RAT end-to-end evaluation

To assess the performance of the ns-3 Multi-RAT interworking platform with attached 5G GFDM, NI LTE and 802.11 physical layers we conducted measurements for end-to-end throughput. The setup is comprised of a remote host node that acts as a client application which sends UDP packets with payload data in downlink direction over the different RATs. The corresponding server application that represents the packet sink for these measurements resides in the 5G GFDM, LTE or WIFI UE node. The UDP packets which are sent from the client are injected into ns-3 with the help of a tap bridge. This tap bridge appears to the Linux host as a normal Ethernet interface with an IP address. An external arbitrarily configurable UDP packet generation application is used to generate traffic. Similarly, on the server side of the transmission, the packets are output from the UE node through a tap bridge to an external server sink application that calculates the end-to-end throughput. To assess whether the whole system can convey an envisaged throughput, we alter packet sizes as well as inter packet intervals in the traffic generation of the client application.

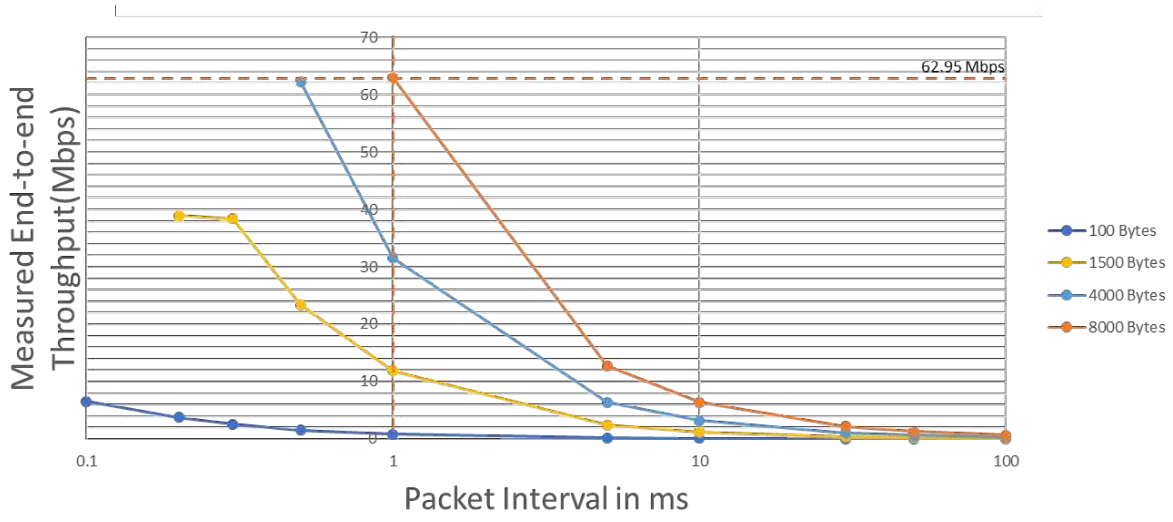

Figure 46 WIFI DL throughput for varying packet sizes and inter-packet intervals

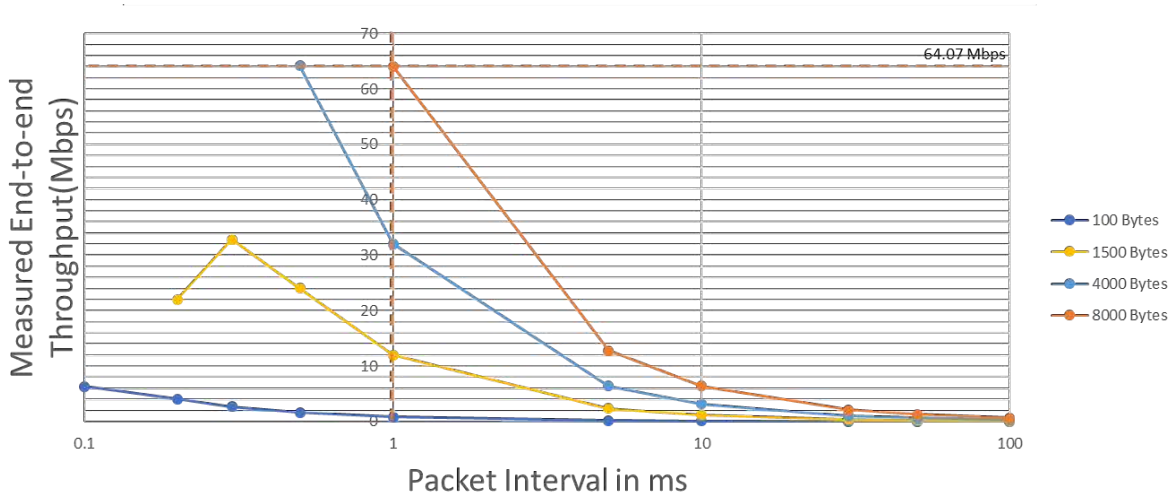

Figure 47 LTE DL throughput for varying packet sizes and inter-packet intervals

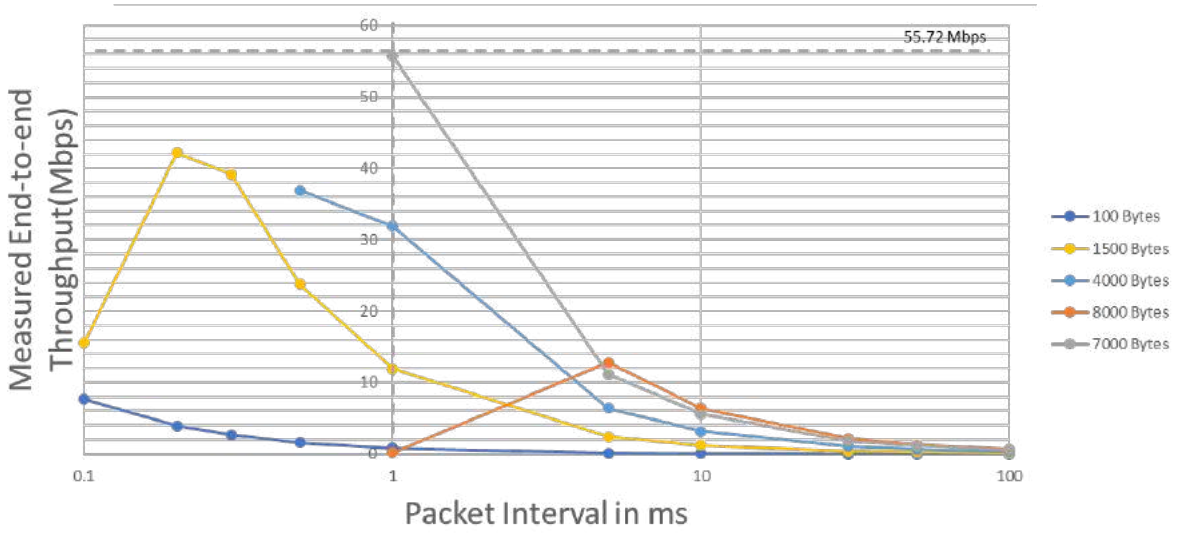

Figure 48 5G GFDM DL throughput for varying packet sizes and inter-packet intervals

In general, for WiFi the throughput for all considered packet sizes follows the theoretical calculations depending on the packet transmission interval. For small packet sizes such as 100 Bytes, the packet interval can be reduced to 0.1 ms. This represents a use case where packets of small size need to be conveyed with low latency. The prototyping platform can convey the theoretical end-to-end throughput with only negligible deviation for small packet transmission intervals. When increasing packet size, the packet transmission interval cannot arbitrarily lower to yield higher throughputs, because this higher amount of payload packets doesn't fit to the maximum physical layer transport block size which is 9422 bytes per ms. The system saturates at around 63 MBit/s. If one lowers the packet transmission interval further, packet losses start to occur which in turn also lower the throughput in a non-deterministic way.

For LTE, the behaviour is similar, and the achievable end-to-end throughputs approach the theoretical limit of 75 MBit/s for an LTE configuration with 20 MHz bandwidth and highest modulation and coding scheme (MCS). A small gap between theoretical and achieved throughput can still be observed but this is due to the header overhead that is added for ns-3 control content and through the different layers of the full LTE stack implementation.

The 5G GFDM results are identical to LTE for packet sizes up to 7000 bytes because similar downlink control information (DCI) configurations are used. As introduced earlier the 5G GFDM physical layer is restricted to an allocation of 80 resource blocks which leads to a theoretical limit of 60 MBit/s for 20 MHz bandwidth and MCS 28. Similar to LTE a small gap between theoretical and achieved throughput can be observed due to the aforementioned reasons.

The throughput curves for WiFi, LTE and 5G GFDM look similar as expected for an end-to-end throughput measurement where the PHY and higher layers of the respective standard are only abstractions for data transfer. The measurements show that the prototyping platform can achieve throughput that is close to theoretical limits. As a summary the maximum throughput evaluated on this platform is described in Table 7.

	WIFI		**LTE**		**5G GFDM**
Max. Throughput [Mbps]	62.95		64.07		55.72
Packet size [bytes]	4000	8000	8000	4000	7000
Interpacket interval [ms]	0.5	1	0.5	1	1

Table 7 Maximum throughput for different RATs evaluated on the Multi-RAT platform

4.3.3 Testbed integration

To conclude, the development of a complete Multi-RAT platform for re-search on interworking techniques was successfully achieved during ORCA project, see Figure 49.

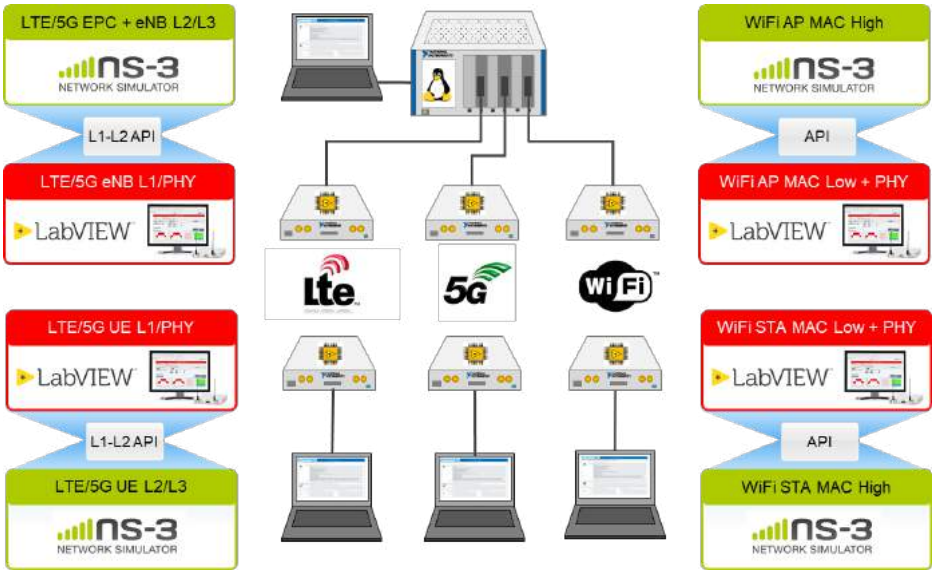

Figure 49 Final Multi-RAT platform deployed into TUD testbed

The software was made available at https://github.com/ni/NI-ns3-ApplicationExample and is accessible for everyone. It includes open call extensions like LWA/LWIP and DC. An extensive getting started guide helps experimenters to setup their one Multi-RAT scenario easily. The Multi-RAT platform was deployed to TUD testbed (http://owl.ifn.et.tu-dresden.de/orca/). The page http://owl.ifn.et.tu-dresden.de/NI/MultiRat-Coord.html is the main entry point for this platform.

5 CONCLUSIONS

In conclusion, this deliverable reports on data plane functionality for the final operational real-time SDR platforms. The devolvement's achieved in Year 3 of ORCA are covering 26GHz millimetre wave systems, various PHY layer improvements and different higher layer SDR integrations. Most of them are implemented to the final showcases described in D2.5. ORCA enhances the current state-of-the-art SDR capabilities by further improving data-plane SDR solutions on heterogeneous hardware platforms that can combine high data rates or low latencies with fast design cycles and high versatility. Therefore, each of the partners has extended and finalized the functionalities which are listed as follows:

- TUD – In year 3 a mmWave frontend was developed and integrated to an USRP based SDR allowing experiments with 5G 26 GHz bands. Furthermore, transceiver enhancements of the GFDM PHY implementation towards 5G, including flexible numerology and increased spectral efficiency were made.
- KUL – For a theoretically and experimentally case study the KULeuven massive MIMO setup was enhanced by incorporating a modular antenna design which allows multiple antenna topologies easy to deploy indoor to support virtual single and multi-cell channel experiments. As a result of the presented experiment, it is proven that the distribution of the antennas in an indoor scenario allows a higher spectral efficiency,
- IMEC – In this year a multi-antenna based concurrent transmission scheme is proposed, and the multi-channel virtual transceiver based on the DSSS accelerator is finalized with optimized FPGA footprint. The virtual transceiver has been integrated with TAISC, and the full stack communication based on virtual transceiver is used in showcase 2.
- TCD – To ensure isolation between virtual radios over unreliable connections dynamic padding of I/Q streams was implemented and integrated with HyDRA. Further computational resource scaling for dynamically allocating resources to end-to-end network slices, which operate on top of a virtualised network infrastructure comprised of hybrid SDR-SDN functionality, possessing RATs, switches and controllers was implemented in software.
- NI – In collaboration with TUD the Multi-RAT experimentation platform was enhanced and completed with a real-time 5G path supporting flexible numerologies using the NI L1-L2 API with 5G extensions and ns-3 as upper layer protocol stack. With this, 5G-LTE interworking experiments are possible using the dual connectivity (DC) functionality. Further the Multi RAT platform was evaluated for end-to-end data transmission.

Additionally, a brief description was given on how the aforementioned implementations were integrated in the third year ORCA showcases and in the partners' testbeds.

REFERENCES

[1] ORCA Deliverable D4.5, Final operational SDR platforms with end-to-end capabilities, Jan 2020.
[2] ORCA Deliverable D2.5, “Final Evaluation of the Showcases”, Jan 2020.
[3] ORCA Deliverable D3.1, First operational real-time SDR platforms, Dec 2017.
[4] “MaMIMO User Grouping Strategies: How much does it matter?” Guevara, Andrea P., De Bast, Sibren, Pollin, Sofie. 53rd IEEE Asilomar Conference on Signals Systems and Computers, 3-6 Nov 2019, Pacific Grove CA, USA.
[5] A. Nimr, M. Chafii and G. Fettweis, "Unified Low Complexity Radix-2 Architectures for Time and Frequency-Domain GFDM Modemin", in *IEEE Circuits and Systems Magazine* (MCAS), Dec 2018.
[6] IEEE 802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks. https://standards.ieee.org/standard/802_15_4-2015.html, Dec 2019.
[7] ORCA Functionality: Generalized API, https://orca-project.eu/wp-content/uploads/sites/4/2017/09/ORCA_2.2-08-v1.1.pdf, Dec 2018.
[8] LabVIEW Communications LTE Application Framework 2.5, Manual, Oct 2018.
[9] ORCA Deliverable D3.3, “Enhanced operational real-time SDR platforms”, Dec 2018.
[10] A. Nimr, S. Ehsanfar, N. Michailow, M. Danneberg, D. Zhang, H. Rodrigues, L. Mendes and G. Fettweis, Chapter: “Generalized Frequency Division Multiplexing: A Flexible Multi-carrier Waveform,” in “Multiple Access Techniques for 5G Wireless Networks and Beyond,” Springer Verlag, 2018, ISBN: 978-3-319-92090-0.
[11] LabVIEW Communications 802.11 Application Framework 2.1, Manual, Mar 2018.
[12] 5G-LENA simulator, https://5g-lena.cttc.es, Dec 2019.
[13] ns-3 Model Library, Manual, https://www.nsnam.org/docs/models/html/index.html, Dec 2019.