GRANT **AGREEMENT NO.: 732174**

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

Orchestration and Reconfiguration Control Architecture

D3.3: Enhanced operational real-time SDR platforms

Revision: v.0.1

Work package	WP3
Task	Task 3.1, 3.2, 3.3
Due date	31/12/2018
Submission date	21/12/2018
Deliverable lead	NI
Version	1.0
Authors	Muhammad Aslam (IMEC), Wei Liu (IMEC), Joao F. Santos (TCD), Jonathan van de Belt (TCD), Walter Nitzold (NI), Clemens Felber (NI), Vincent Kotzsch (NI), Andrea Guevara (KUL), Martin

	Danneberg (TUD), Shahab Ehsanfar (TUD), Ana Martinez (TUD), Ahmad Nimr (TUD)
Reviewers	Andrea Guevara (KUL), Roberto Bomfin (TUD)

Abstract	This deliverable will include the Year 2 implementation results on the available SDR platforms and the risk analysis whether all scenarios defined in WP2 are possible for real-time prototyping in the testbed facilities. This deliverable will also include a plan for functionality to be implemented in the final year.
Keywords	real-time, SDR, data plane, PHY, MAC, higher layer, testbeds

Disclaimer

The information, documentation and figures available in this deliverable, is written by the ORCA (Orchestration and Reconfiguration Control Architecture) – project consortium under EC grant agreement 732174 and does not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

Confidential *- The information contained in this document and any attachments are confidential. It is governed according to the terms of the project consortium agreement*

Copyright notice

© 2017 - 2020 ORCA Consortium

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)*

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc

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 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.

In this deliverable the focus is on essential data plane SDR functionality to achieve low latency and high throughput operation by allowing real-time operation. Therefore, ORCA extends the current state-of-the-art SDR capabilities by designing data-plane SDR solutions on heterogeneous hardware platforms that can combine high data rates or low latencies with fast design cycles and high versatility.

This deliverable will include the ORCA Year 2 implementation results on the available SDR platforms and the risk analysis whether all scenarios defined in WP2 are possible for real-time prototyping in the testbed facilities. This deliverable will also include a plan for functionality to be implemented in the final project year.

© ORCA Consortium 2017-2020

TABLE OF CONTENTS

EXECUTIVE SUMMARY **3**
TABLE OF CONTENTS **4**
LIST OF FIGURES **7**
LIST OF TABLES **10**
ABBREVIATIONS **11**
1 INTRODUCTION **13**
1.1 Organization of the Deliverable 13
2 MMWAVE SYSTEMS **15**
2.1 Implementation results 15
2.1.1 PHY improvements 15
2.1.2 Higher layer protocol integration 21
2.1.3 Testbed integration 26
2.2 Relation to showcase 26
2.3 Risk analysis 26
2.4 Implementation plan 26
2.4.1 PHY improvements 26
2.4.2 Higher layer protocol integration 26
2.4.3 Testbed integration 27
3 SPECTRUM EFFICIENCY IN MULTI-CELL MASSIVE MIMO SYSTEMS **28**
3.1 Implementation results 29
3.1.1 PHY improvements 29
3.1.2 Higher layer protocol integration 32
3.1.3 Testbed integration 32
3.2 Relation to showcase 33
3.3 Risk analysis 33
3.4 Implementation plan 33
3.4.1 PHY improvements 33
3.4.2 Higher layer protocol integration 33
3.4.3 Testbed integration 33
4 FLEXIBLE PHYSICAL LAYER BASED ON GFDM **34**
4.1 Implementation Results 34
4.1.1 PHY improvements 36

© ORCA Consortium 2017-2020

4.1.1.1 Automatic Gain Control 36
4.1.1.2 RF Control Module 37
4.1.1.3 Scrambling 38
4.1.1.4 Interleaving 38
4.1.1.5 Higher Order QAM Mapper / De-mapper 38
4.1.1.6 CRC 38
4.1.1 Higher layer protocol integration 39
4.1.2 Testbed integration 41
4.2 Relation to showcase 41
4.3 Risk analysis 41
4.4 Implementation plan 41
4.4.1 PHY improvements 41
4.4.2 Higher layer protocol integration 44
4.4.3 Testbed integration 44
5 HYBRID FPGA PLATFORM INCL. FLEXIBLE MAC 45
5.1 Implementation results 45
5.1.1 PHY improvements 45
5.1.2 Higher layer protocol integration 51
5.1.3 Testbed integration 51
5.2 Relation to showcase 52
5.3 Risk analysis 52
5.4 Implementation plan 52
5.4.1 PHY improvements 52
5.4.2 Higher layer protocol integration 52
5.4.3 Testbed integration 53
6 RADIO VIRTUALISATION ON DEMAND 54
6.1 Implementation results 54
6.1.1 PHY improvements 54
6.1.2 Higher layer protocol integration 56
6.1.3 Testbed integration 58
6.2 Relation to Showcase 58
6.3 Risk analysis 58
6.4 Implementation plan 58
6.4.1 PHY improvements 59
6.4.2 Higher layer protocol integration 59

6.4.3 Testbed integration 59
7 NS-3 BASED PROTOTYPING PLATFORM FOR RAT INTERWORKING 60
7.1 Implementation results....60
7.1.1 PHY improvements 60
7.1.2 Higher layer protocol integration....60
7.1.3 Testbed integration 78
7.2 Relation to showcase 78
7.3 Risk analysis 78
7.4 Implementation plan 78
7.4.1 PHY improvements 78
7.4.2 Higher layer protocol integration....79
7.4.3 Testbed integration 79
8 CONCLUSIONS 80
9 REFERENCES 81

LIST OF FIGURES

Figure 1: Phase noise measurements from different mmWave frontends. 15

Figure 2: Signal Processing block diagram of mmWave Transceiver with highlighted phase compensation steps. 16

Figure 3: Schematic visualization of two-step compensation principle.....17

Figure 4: Measured SINR gain without and with phase compensation algorithm. 18

Figure 5: Measurement Head to characterize different beams and codebooks. 18

Figure 6: Characterization results for a single beam ((a) and (b)) from the azimuth codebook as well as the coverage of the combined codebook ((c) and (d)). 19

Figure 7: 3D Beam Pattern and Heatmap of complete 2D beam codebook.....20

Figure 8: 16x16 Butler Matrix prototype.....21

Figure 9: Beam steering algorithm inputs and outputs. 22

Figure 10: Flow chart of Parameterized Exhaustive Search Algorithm. 23

Figure 11: Flow chart of measurements combination procedure. 24

Figure 12: Example beam pattern set with Nprob=16.....24

Figure 13: Manual beam steering configuration. 25

Figure 14: Integration of mmWave with testbed. 26

Figure 15: Scenario selection under different levels of cooperation. 29

Figure 16: Average UL SE for different SIR thresholds and combining vectors. 30

Figure 17: Average UL SE comparison for different combining vectors when PM-MMSE $\Gamma = 5dB$ and its computational cost. 30

Figure 18: Average spectral efficiency per cell versus the number of active antennas, for different cooperation agreement and under different types of combining vectors in a system. ...31

Figure 19: Overlook of the symmetric two-cell Massive MIMO outdoor scenario. 32

Figure 20: Robot demonstrator – connecting two robots over a low latency link. 34

Figure 21: Evaluation set-up using a real-time channel emulator. 34

Figure 22: Packet error [%] rate as a function of the path loss [dB]. 36

Figure 23: Automatic Gain Control state machine. 37

Figure 24: RF Control Module. 37

Figure 25: Higher Order QAM Mapper / De-mapper. 38

Figure 26: Cyclic Redundancy Check. 39

Figure 27: Latency as a function of the packet rate between a UDP sending application and an application receiving UDP packets on the same Windows 2012 server host machine, without network or PHY. 39

Figure 28: Latency as a function of the packet rate between a UDP sending application and an application receiving UDP packets on the same host machine, including the FPGA bypassing the

PHY signal processing on the FPGA. This figure depicts the minimal achievable latency with the hardware set-up, without any signal processing.....40

Figure 29: Latency of different PHY implementations. 40

Figure 30 Zak transform. 42

Figure 31: Unified architecture for TD and FD GFDM processing. 43

Figure 32: The impact of turnaround time on latency during TDD operation in a D2D scenario. 45

Figure 33: Block diagram of (a) the conventional TDD and (b) the proposed BBHD-RFFD approaches. 46

Figure 34: Experimental setup to capture IQ samples from AD9361. 47

Figure 35: The comparison of self-interference (visualized by PSD), between FDD and TDD-WOC modes of AD9361, under the following settings: (a) Tx and Rx ports are directly connected and DC tracking is disabled; (b) Tx and Rx ports are directly connected and DC tracking is enabled; (c) Tx and Rx ports are connected through a power splitter with DC tracking enabled; (d) Tx and Rx ports are connected to two antennas, placed orthogonal to each other.....48

Figure 36: Experimental setting for turnaround time and receiver sensitivity measurements. ...49

Figure 37: Measured Rx power (in dBm) vs AD9361 Tx attenuator.....51

Figure 38: Receiver sensitivity measurements.....51

Figure 39: The timed buffer consumes the received IQ samples and generates the windows that the radio hypervisor requires. The same process happens in the opposite direction.....55

Figure 40: Delay measurements for different virtual radio configurations. All the measurements were made using a real bandwidth of 2 MHz and an overall FFT size of 8192 bins, which corresponds to the worst case scenario in Section 6.1.2.1.....56

Figure 41: An example of the steps involved in negotiating the use of resources and allocating a new virtual radio. XVL (blue) stands as a wrapper around the radio hypervisor (grey).....57

Figure 42: The provisioning delay to fulfil a virtual radio request. The time taken for creating virtual radios is proportional to the number of allocated FFT bins, but not to the overall IFFT size of the radio hypervisor. 58

Figure 43: Illustration of virtual radios with virtual RF front-ends and virtual radio function stacks. 59

Figure 44: Envisaged Platform for Multi-RAT experimentation 60

Figure 45: Overview about the generalized NI L1-L2 API. 61

Figure 46: Simplified communication between L1 and L2 based on API message types. 62

Figure 47: PHY SAP General Message Structure.....62

Figure 48: LTE PHY SAP Message Definitions. 63

Figure 49: LTE PHY SAP Message Sequence Chart.....64

Figure 50: API Implementation Diagram (eNB example).....65

Figure 51: USRP-2974 Hardware Device.....66

Figure 52: Overview of the LTE EPC simulation model [27]. 67

© ORCA Consortium 2017-2020

Figure 53: Synchronization between CPU and FPGA clock. 68
Figure 54: NI L1-L2 API implementation in the ns-3 LTE module (based on [27]). 69
Figure 55: ns-3 LTE eNB Implementation Diagram of NI API. 69
Figure 56: New ns-3 LTE eNB and UE Object hierarchy. 70
Figure 57: New ns-3 WiFi AP and STA Object hierarchy. 71
Figure 58: LTE WiFi Multi-RAT Prototyping System. 71
Figure 59: Multiuser Scheduling main elements. 72
Figure 60: High Level MAC Architecture. 73
Figure 61: Integration process of the C++ based ns-3 scheduler into LabVIEW. 74
Figure 62: Shared Library Interface (SLI) declares the functions, parameters and return values of the C-functions. 74
Figure 63: Scheduler architecture and interfaces. 75
Figure 64: RR Algorithm (left) vs. TDMT algorithm (right). 76
Figure 65: 5G-NR MAC PDU assembling. 77
Figure 66: 5G-NR LabVIEW MAC in LTE AFW 2.2. 77
Figure 67: LabVIEW 5G MAC demo at Globecom Conference 2018. 78

LIST OF TABLES

Table 1: Network performance metrics for all the study scenarios, with 64 active antennas. 32

Table 2: Channel Model A Indoor Office Rec. ITU-R M.1225. 35

Table 3: Channel emulation parameter. 35

Table 4: Transmission parameter. 36

Table 5: Summary of GFDM equations. 43

Table 6: Configuration parameters for modulator. 44

Table 7: Demodulator configuration, FD equalization. 44

Table 8: Self-Interference measured by power difference in the 4 cases. 49

Table 9: The average TT measured under three duplex operation modes of AD9361with a standard deviation of around 1µS. 50

© ORCA Consortium 2017-2020

ABBREVIATIONS

ADC	Analog Digital Converter
API	Application Programming Interface
AP	Access Point
BBHD-RFFD	BaseBand processing unit operating in Half Duplex mode and analog Radio Frequency front-end operating in Full Duplex mode
BPU	Baseband Processing Unit
BS	Base Station
CPU	Central Processing Unit
DAC	Digital Analog Converter
D2D	Device to Device
DL	Downlink
DSSS	Direct-sequence spread spectrum
FD	Full Duplex
FFT	Fast Fourier Transform
FPGA	Field-programmable gate array
GUI	Graphical user interface
GFDM	Generalized Frequency Division Multiplexing
IP	Intellectual Property
IQ	In-phase and Quadrature
LoS	Line of sight
MAC	Medium access control
MaMIMO	Massive MIMO
MIMO	Multiple-input multiple-output
NLoS	Non-Line of sight
NR	New Radio
OFDM	Orthogonal frequency-division multiplexing
PSS	Power Saving Scheme
RAM	Random Access Memory
RAT	Radio Access Technology
RF	Radio Frequency
Rx	Receiver
SPI	Serial Peripheral Interface Bus
TAISC	Time Annotated Instruction Set Computer

© ORCA Consortium 2017-2020

TDD	Time Division Duplexing
TDMA	Time-division multiple access
TT	Turnaround Time
TTI	Transmission Time Interval
Tx	Transmitter
UE	User Equipment
UDP	User Datagram Protocol
USRP	Universal Software Radio Peripherals
UL	Uplink
ZF	Zero Forcing

© ORCA Consortium 2017-2020

1 INTRODUCTION

ORCA wants to accelerate flexible end-to-end network experimentation by making open and modular software and hardware architectures available that smartly use novel versatile radio technology. More-specifically real-time Software Defined Radio (SDR) platforms meeting the requirements in terms of runtime latencies, throughput, fast reconfiguration and reprogramming are needed to fulfil this overall objective. Further ORCA will offer experimental facilities, with SDR devices incorporating relevant software and hardware building blocks that allow easy design, implementation and programming, while also achieving low runtime delay allowing end-to-end networking experimentation.

In this deliverable the focus is on essential data plane SDR functionality to achieve low latency and high throughput operation by allowing real-time operation. Therefore, ORCA extends the current state-of-the-art SDR capabilities by designing data-plane SDR solutions on heterogeneous hardware platforms that can combine high data rates or low latencies with fast design cycles and high versatility. Flexible end-to-end reconfiguration facilities are described in detail in D4.3 [1].

This deliverable will include the ORCA Year 2 implementation results on the available SDR platforms and the risk analysis whether all scenarios defined in WP2 are possible for real-time prototyping in the testbed facilities. This deliverable will also include a plan for functionality to be implemented in the final project year.

1.1 Organization of the Deliverable

The remaining sections of the document are organized as follows:

- Section 2 describes improvements made to the 60 GHz mmWave system and the plans for the 28 GHz mmWave antennas to be used with USRPs. NI shows a concept to improve link level quality with enhanced phase noise compensation and the exemplary characterization of a 2D codebook for beam steering for the 60 GHz mmWave system. In addition, TUD describes how the system is integrated to the testbed.
- Section 3 describes two methods to improve the spectral efficiency in a multi-cell Massive MIMO system. The first method proposes a new combining vector to reduce computational cost without sacrificing spectral efficiency, in a single network with two-cells. While the second method considers the coexistence of two networks under different levels of cooperation and its impact to the spectral efficiency.
- Section 4 describes the improvements of the TUD's GFDM framework. In particular, the latency of GFDM PHY was measured, demonstrating that the system is suitable for low-latency applications. In addition, details on general improvements are provided, e.g, RF control and algorithms. Finally, the plan for year 3 is described.
- Section 5 describes IMEC's SDR physical layer improvement regarding Tx/Rx turnaround time, the feature is integrated in the upper layer, and it is critical for latency performance of a wireless link.
- Section 6 describes TCD´s radio virtualisation on demand, supporting the creation of radio slices on the fly.
- Section 7 describes the real-time L1/L2 API implementation for NIs SDRs as well as for ns-3 and the addition of a 5G MAC Protocol implementation

© ORCA Consortium 2017-2020

Each section is then subdivided into: (i) summary of the implementation results obtained during Year 2 and these results are integrated into the testbeds, (ii) how the implemented solution is integrated in its respective Year 2 ORCA showcase, (iii) a risk analysis whether all scenarios defined in WP2 are possible for real-time prototyping in the testbed facilities (iv) and the extension plans currently envisioned by the partner for Year 3, the final project year.

© ORCA Consortium 2017-2020

2 MMWAVE SYSTEMS

In order to support the development of mmWave systems for small cells, this Section focuses on the improvements of the mmWave systems that are offered in TUDs testbed. In particular, NI improved the 60GHz mmWave system by implementing a phase noise compensation enhancing the link-level performance. In addition, antenna characterization for 2D codebook was performed. Moreover, the functionalities of manual and automatic beam steering algorithms are integrated to the system, increasing the range of applications for the 60GHz mmWave system. TUD integrated the 60 GHz mmWave to the testbed and also developed a 26 GHz mmWave antenna to be used with USRPs, allowing deployment of mmWave with lower cost and also to benefit from the flexible GFDM PHY.

2.1 Implementation results

2.1.1 PHY improvements

2.1.1.1 60GHz mmWave System

Phase Noise Compensation for Improved Link-Level Performance

In a real transceiver system, the base band processing needs to take care of a variety of different impairments such as carrier frequency offset, quadrature impairments and more. Of specific interest in milimeter-wave (mmWave) transceivers is phase noise which stems from the implementation impairments of the local oscillators at the high frequencies of a mmWave system. Experiments have shown that the influence of this impairment can severely degrade the decoding link-level decoding performance of a transceiver system if no proper care is taken of compensating this behaviour. Figure 1 shows four realizations of phase noise over the same acquired signal time for different frontends as well as different modulation formats.

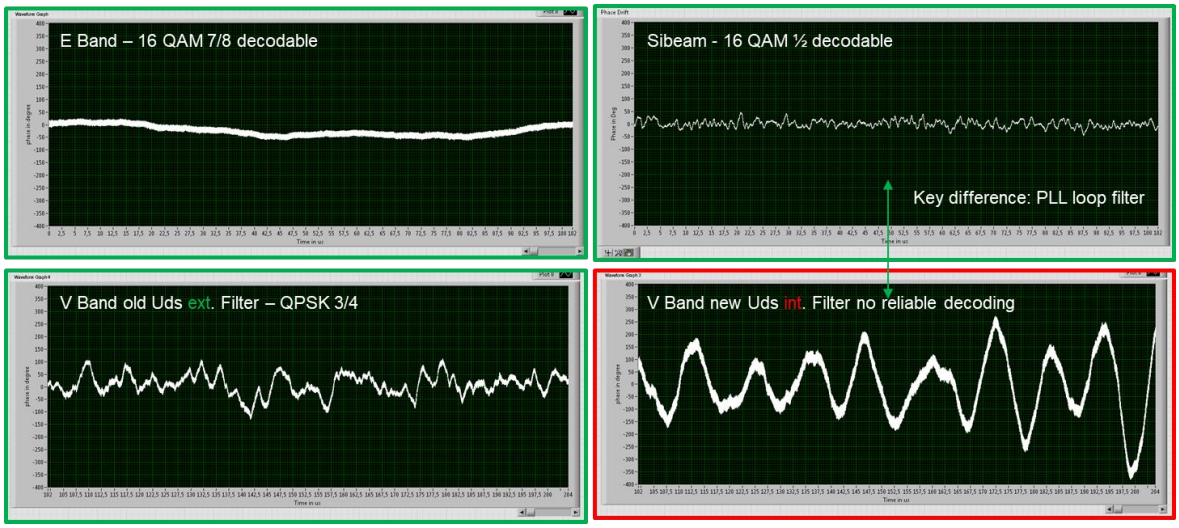


Figure 1: Phase noise measurements from different mmWave frontends.

The phase drift in these examples ranges from a few degrees in the E band to multiple 100 degrees for the V band. As these fluctuations are quite diverse, a compensation algorithm needs to adapt to all of them and also cover good performance for frontends with larger phase drifts to ensure proper link-level decoding performance.

Analysing the drift of a single symbol block reveals that it can be separated into a linear component and small-scale fluctuations around that linear component. A current limitation of blind compensation algorithms applied after equalization is that these can become unstable when the linear component is too large. Additionally, the phase drift has an influence on the channel estimation and equalization performance worsening the overall link-level performance so ideally, the linear component should be taken care of before any further estimation is done. The solution developed for the mmWave setup in ORCA is a two-step approach that subsequently takes care of compensating both components in the receiver chain. The implementation diagram in Figure 2 shows the position of the sub algorithms in the receiver chain.

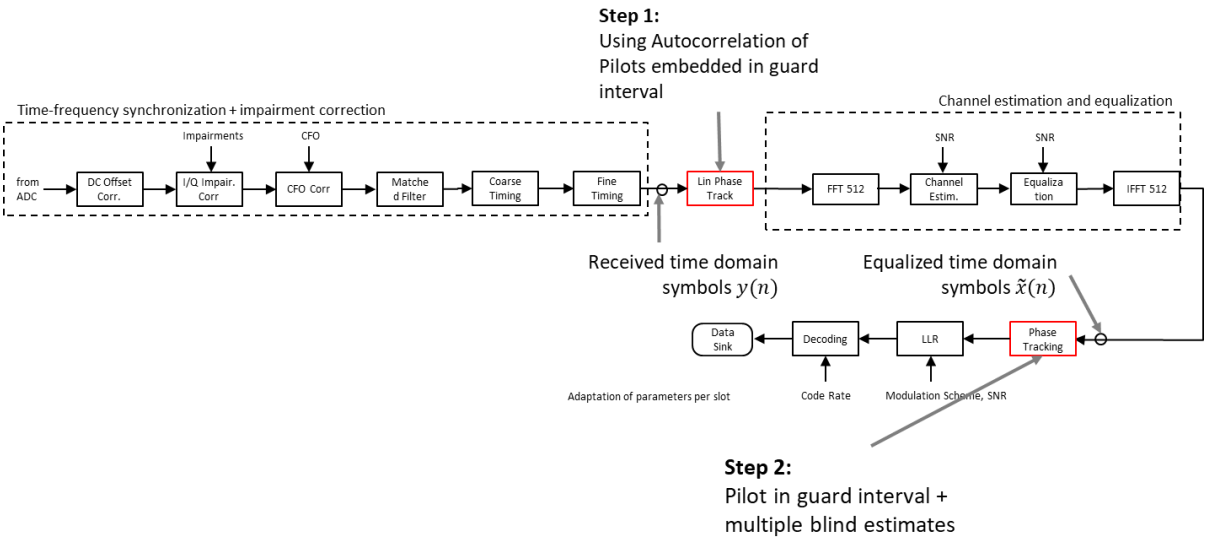


Figure 2: Signal Processing block diagram of mmWave Transceiver with highlighted phase compensation steps.

The linear phase drift component is directly compensated after initial time and frequency synchronization and impairment compensation. After this first step, normal FFT processing and equalization can be performed. The next two paragraphs shortly describe the principles of the aforementioned two steps.

The compensation of the phase drift in Step 1 makes use of the signal structure. Figure 3 shows the phase drift before applying compensation Step 1 over subsequent symbol blocks.

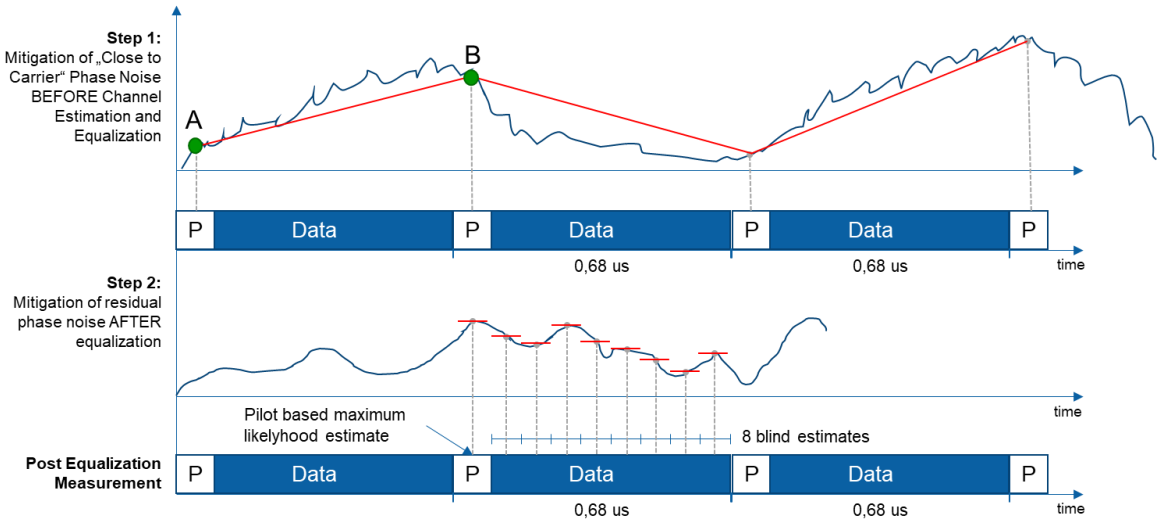


Figure 3: Schematic visualization of two-step compensation principle.

To estimate the absolute phase difference between point A and B in the signal, some a-priori know data/pilots need to be utilized. This can be

- Known pilot-preambles, e.g., in single-carrier implementations such as 802.11ad single-carrier PHY
- Cyclic Prefix/Postfix: used in 802.11ad OFDM PHY, LTE and 5G NR

This a-priori knowledge of the signal structure can be exploited via auto-correlation to yield the exact phase-difference between point A and B in the signal and subsequent symbol blocks accordingly for linear interpolation and corresponding compensation.

After compensation for the linear phase drift component, the second step incorporates the use of a blind feed-forward algorithm. The feedforward architecture is considered here as advantageous, as the mmWave system has very short symbol times and PHY processing is therefore very time-constraint. An iterative feedback-algorithm would be too time-consuming. Secondly, an algorithm for phase compensation can be applied that is normally prone to time-dispersive channels. As we employ the second step after channel equalization, the phase estimation algorithm sees a signal without any intersymbol interference and hence can be simplified. For the current implementation a non-data aided feed-forward algorithm for M-QAM from [2, Sec 5.7] is implemented. The symbol blocks are divided into sublocks and the blind algorithm does a phase estimation on each of those with a zero-order hold interpolation over the time-duration of the subblock. These phase fluctuations are then compensated.

The resulting gain in terms of SINR is shown in Figure 4.

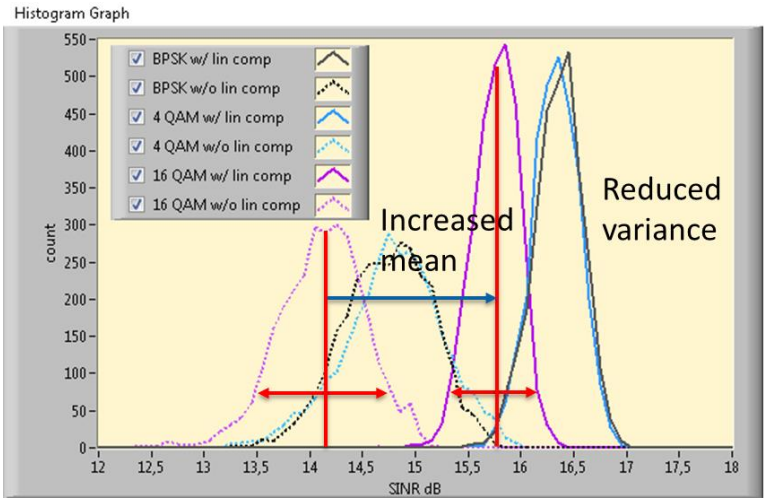


Figure 4: Measured SINR gain without and with phase compensation algorithm.

Transmitted signals under consideration are modulated with BPSK, 4QAM and 16 QAM. The SINR is increased in all cases. For BPSK and 4QAM, the mean of the SINR is increased by approx. 1.7dB. A similar gain is seen for 16QAM. A second advantage of the proposed phase drift compensation is the decrease in variance of the SINR. This results in a more stable link-level decoding performance as given a specific radio transmission scenario, the transceiver system gets a more stable SINR with less fluctuation and therefore reduced bit-errors after decoding.

Two-Dimensional Codebook Characterization

To benefit from beam steering algorithms at the MAC layer the codebooks for the available beams need to be designed carefully for the specific application. The mmWave setup uses an antenna array that in principle is capable of using arbitrary codebooks. For the usage in Year 1 of ORCA, only codebooks were used that incorporated beams differing by the azimuth angle. These were predefined for the specific beam steering frontend. To characterize these codebooks and to get an insight into the beam coverage, extensive characterizations were made to make the different beam configurations visible and gather data about the spatial coverage and antenna gain of beams in these codebooks. The automated measurement head is shown in Figure 5.

Figure 5: Measurement Head to characterize different beams and codebooks.

An exemplary characterization can be seen in Figure 6.

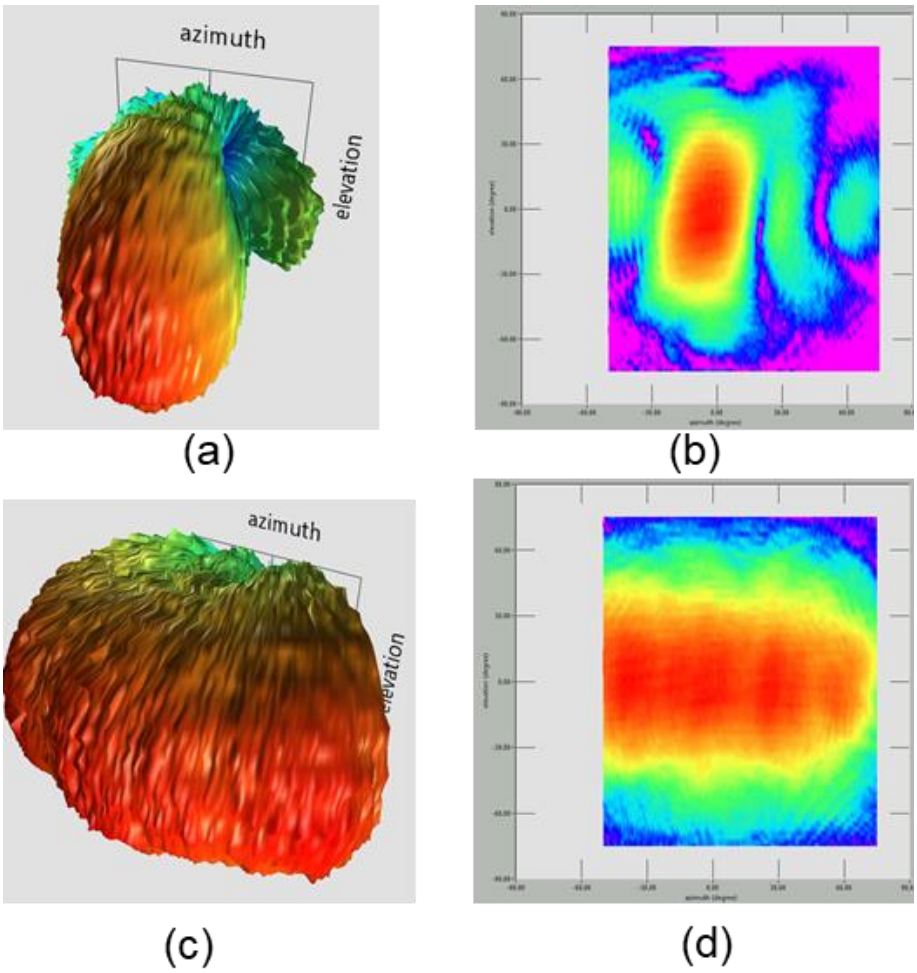
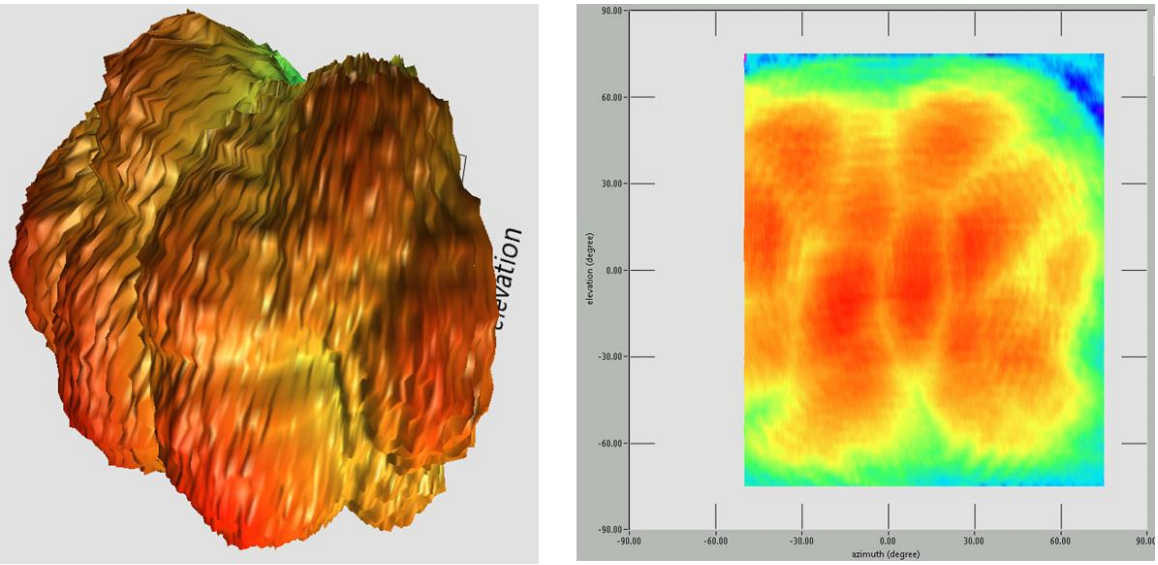


Figure 6: Characterization results for a single beam ((a) and (b)) from the azimuth codebook as well as the coverage of the combined codebook ((c) and (d)).

Different options to characterize the plots are possible to emphasize different aspects of the beams. Two very visual options are shown here exemplarily while the beam characterization options do have more options for characterization. One possibility is the 3D beam pattern that gives a 3D representation of the respective beam and therefore an overview of the directivity of the beam (See Figure 6 (a)). To better assess the coverage of a given beam or codebook, the heatmap can be used to show the distribution of the emitted energy in azimuth and elevation. When a complete codebook should be evaluated regarding coverage, the two above mentioned plots can also be used. The 3D beam plot and heatmap of the complete azimuth codebook are shown in Figure 6 (c) and (d) respectively. Especially in Figure 6 (d), the coverage of the emitted energy in the horizontal (azimuth) direction is clearly visible. This codebook can only steer its beams in azimuth direction. A goal for broadening the application of the beam steering solution in the mmWave setup towards more coverage is the incorporation of codebooks that allow steering of beams into not only the azimuth direction but also elevation direction. In the second year of ORCA, NI characterized such beam steering matrices to generate a codebook that would enhance the coverage. The resulting codebook is shown in Figure 7.

© ORCA Consortium 2017-2020



(a) (b)

Figure 7: 3D Beam Pattern and Heatmap of complete 2D beam codebook.

Comparing the heatmap of the azimuth codebook in Figure 6 (d) with the heatmap in Figure 7 (b) clearly shows an increase in coverage in the elevation direction for the emitted energy. While the azimuth codebook had a coverage of approximately +/- 40 degrees in elevation direction and +/- 60 degrees in azimuth direction, the 2D codebook increases the elevation coverage to +/- 60 degrees resulting in a homogeneous coverage of these two dimensions. This characterization of 2D codebooks will help to increase the options for beam steering algorithm development and gives insight into the way how different codebook designs translate to real-world measured beams. Although, the 2D codebooks have been characterized, it is not planned to integrate a support for these in the setup as the focus of the mmWave system shifted towards the 26 GHz system.

2.1.1.2 26 GHz mmWave System

A multi-beam antenna array connected with a 16x16 Butler matrix was developed to allow mmWave transmission in at 26 GHz. The PCB scan be seen in Figure 1

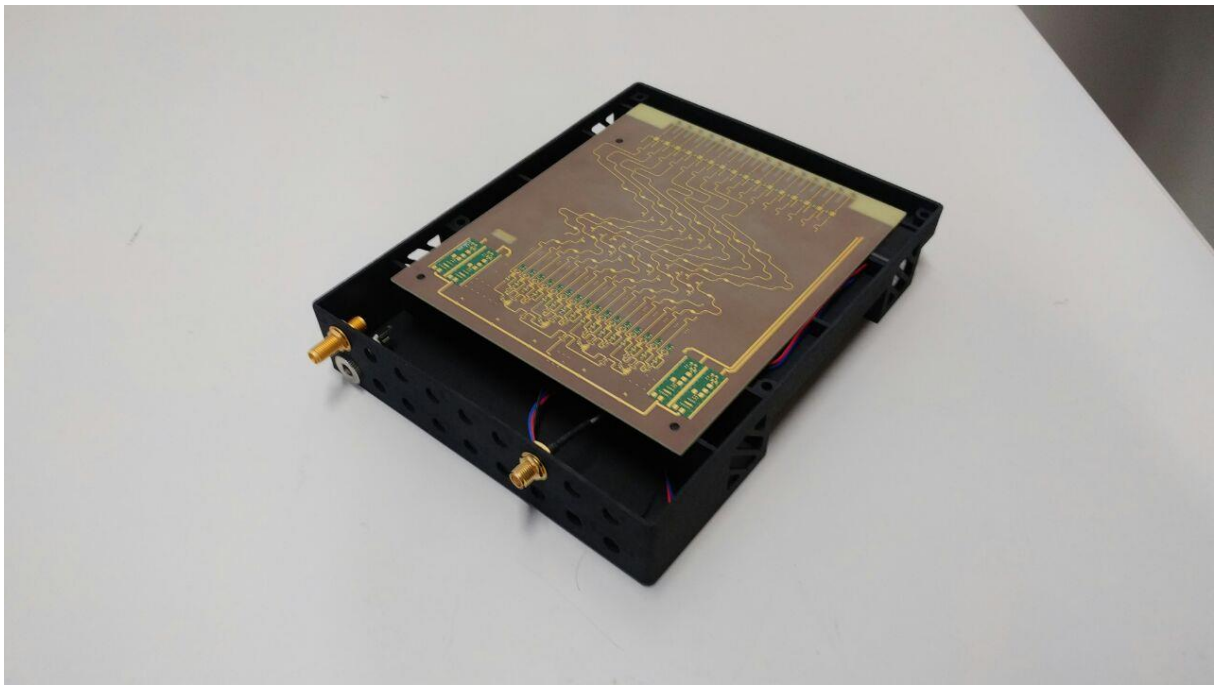


Figure 8: 16x16 Butler Matrix prototype.

The beam pattern of the 8x8 Butler matrix version of this structure was measured. The results showed that the theoretical model is very similar to the practical measurements [3], demonstrating that we can accurately model the irradiation patterns. The compact antenna design was implemented in cost effective PCB technology which is suitable for mmWave systems in 5G.

2.1.2 Higher layer protocol integration

In ORCA Year 2 the support of automatic and manual beam steering functionality at MAC layer (see offer in D2.2 [4]) was made available to the real-time mmWave baseband system supporting bi-directional TDD MAC protocol described in D3.1 [5].

Automatic beam steering functionality

As planned two different beam steering algorithms are now selectable and integrated into the advanced MAC layer (see MAC software architecture in D3.1 [5]) to achieve automatic beam steering functionality.

1. Exhaustive Search
2. <u>Parameterized Exhaustive Search</u>

The main task of the beam steering algorithm is to search the best beam pair in respect to channel quality.

In a scenario without mobility this could be achieved using the basic exhaustive search algorithm (1). In scenarios with mobility a beam tracking functionality is needed in order to ensure a stable link without losing the connection. This tracking functionality was implemented in the parameterized exhaustive search algorithm (2). Which will be described in more detail after a basic overview.

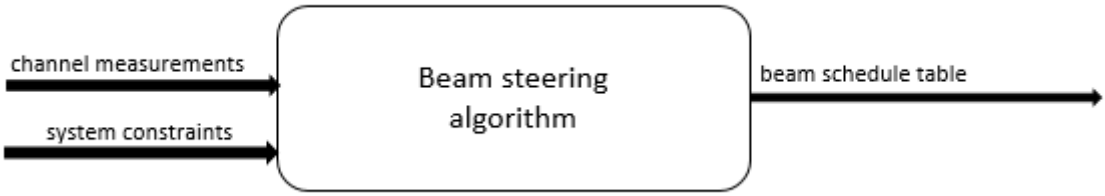


Figure 9: Beam steering algorithm inputs and outputs.

The beam steering algorithm module (Figure 9) system constrains like antenna configuration, radio frame structure as parameters and all channel quality measurements were taken in uplink and downlink from a beam steering database. Relying on the history of channel measurements from the database, the beam steering algorithm generates a beam schedule table targeting at a specific future radio frame. The output scheduling table contains information about the target beam combination for data transmission and probing slot assignments.

Parameterized Exhaustive Search Algorithm

Starting from the basic beam steering algorithm (1) following extensions where added. On the one hand, a measurement combination module was added in order to combine the measurements of multiple radio frames instead of only one in the basic algorithm. Hence, more information about channel quality is available with equal cost by combining multiple history measurements. As a result, the system was able to track channel variance with more updated information. Further transmission throughput was increased by reducing disconnection rate or transmitting with a better channel quality. On the other hand, except for a relatively constant subset of best beam combinations, additional beam combinations are probed. With this, the system is able to achieve updated information about channel variance and adjust transmission beam to reduce link disconnection rate. The amount of additional probing can be configured during setup time. The behaviour of the beam steering algorithm is described in Figure 10 and in the subsequent bullet list.

© ORCA Consortium 2017-2020

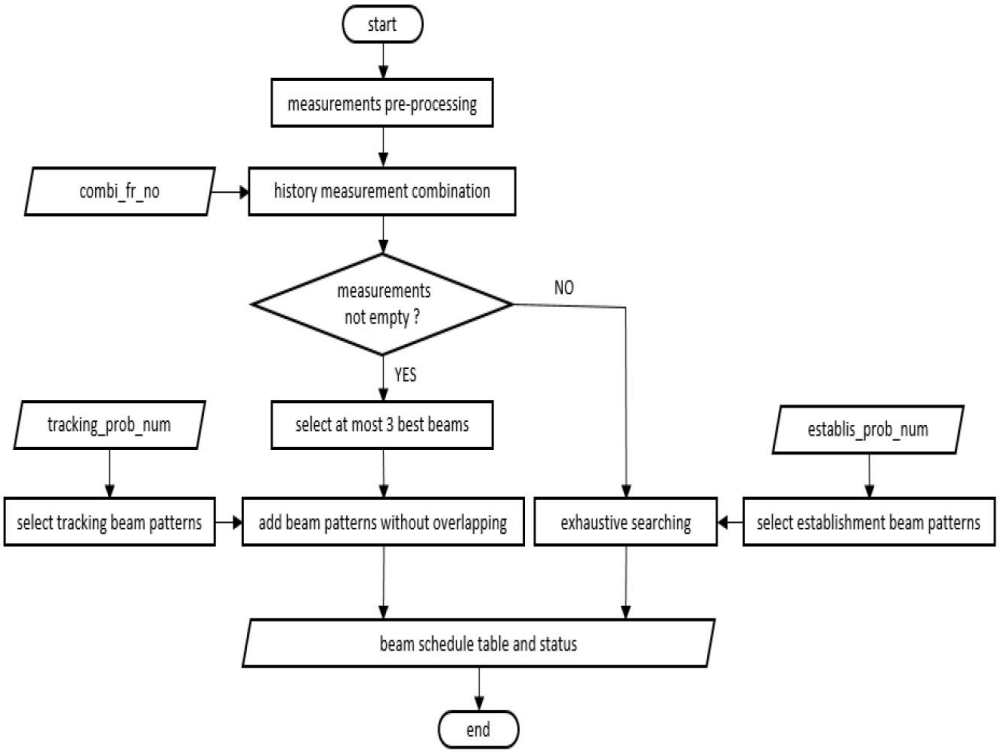


Figure 10: Flow chart of Parameterized Exhaustive Search Algorithm.

- Measurements pre-processing
  - Alignment of Access Point and User Device Measurements.
  - Measurement filtering for a specific user.
- Measurements combination
  - Measurements of multiple history radio frames are combined before selecting the best beam combinations for data transmission or further probing.
  - The number of combined measurements, defined as an algorithm input parameter, is ranging from 1 to 4 currently. The available measurements are defined as follows:

    $$M = \{\vec{M}_n, \vec{M}_{n-1} \dots \dots, \vec{M}_{n-1-H}\}$$

    where n is frame number of the newest available measurements and H denotes the total number of frames we would like to combine.
  - As indicated in Figure 11, the measurements from the newest frame are always reserved because of most accurate description of channel quality. Measurements from other frames are checked one by one from newest to oldest.

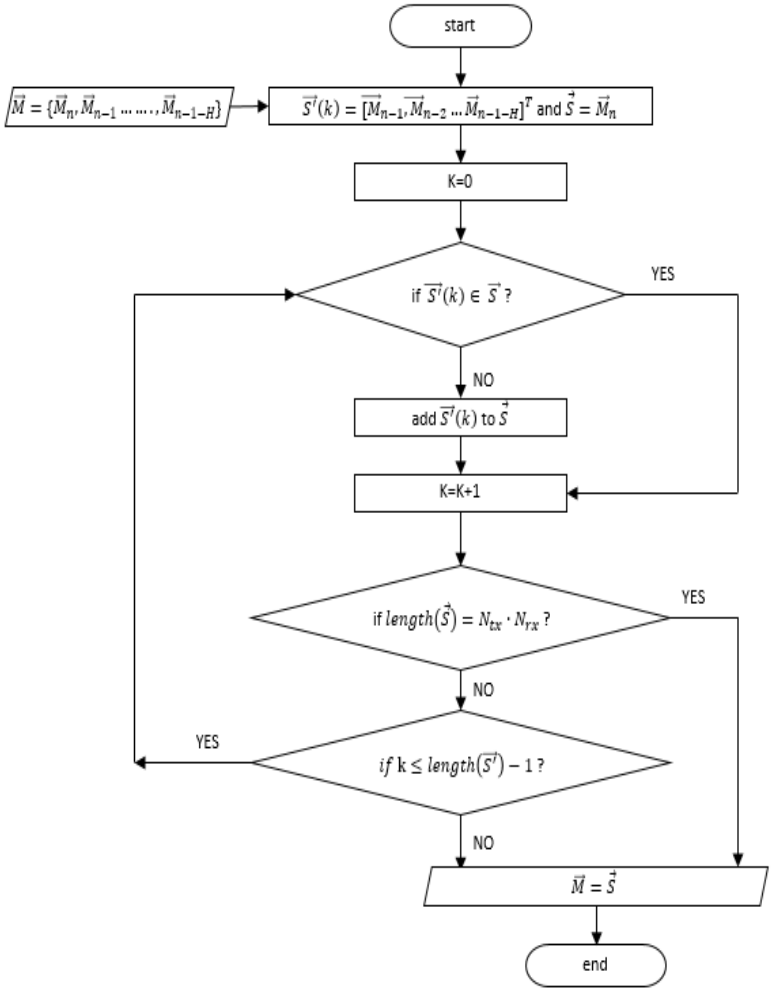


Figure 11: Flow chart of measurements combination procedure.

- Probing beam patterns addition
  - Additional beam combinations are probed to gain more information about the channel.
  - Different sets of beam combinations are probed in successive frames to be able to get a full map of channel information over multiple radio frames.

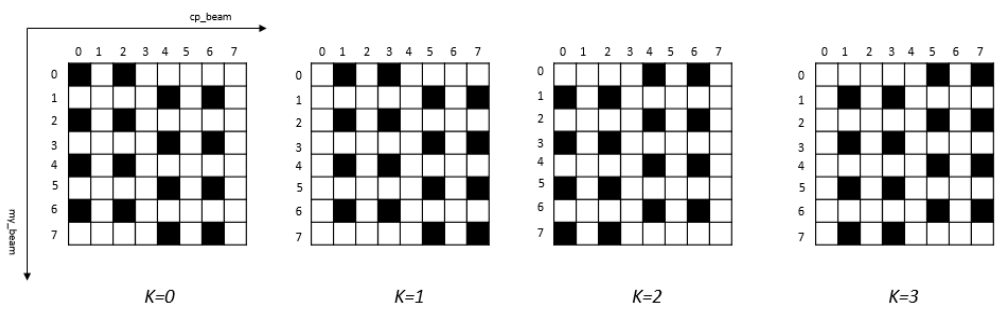


Figure 12: Example beam pattern set with Nprob=16.

  - The table below shows the available beam pattern set of the algorithm and information about whether it is used for establishment mode or tracking mode. N_{prob} indicates how

many beams are probed additionally in each beam pattern which would be applied in a single radio frame and N_{fr_num} indicates the number of different patterns in a pattern set, which is also the number of radio frames over which a full map channel information can be achieved.

N_{prob}	N_{fr_num}	Establishment mode	Tracking mode
2	32		√
4	16	√	√
8	8	√	√
16	4	√	√
32	2	√	

In summary, the key improvements compared to the basic exhaustive search algorithm are the following

- Handle channel variance.
- Increase system robustness with less link disconnections.
- Increase transmission data rate.
- Enable multi-user transmission.

Manual Beam steering functionality

Beside the automatic beam steering functionality using specific algorithms a manual beam steering mode was made available. With this, the experimenter can choose specific beam combination during the experiment. This allows experiments with a restricted directive transmission and allows e.g. side by side comparison of different fixed scenarios. Further, it helps debugging a new developed algorithms and codebooks.

The manual beam steering is controlled from the Access Point. While the beam settings for the Access Point are applied directly (Tx + Rx), the beam settings of the User Device (Tx + Rx) are sent via MAC protocol. Based on this control information the User Device applies the settings which are received from the Access Point. Figure 13 shows a configuration for a manual beam setting on Access Point to ID 3 for Tx + Rx and User Device to ID 5 for Tx + Rx.

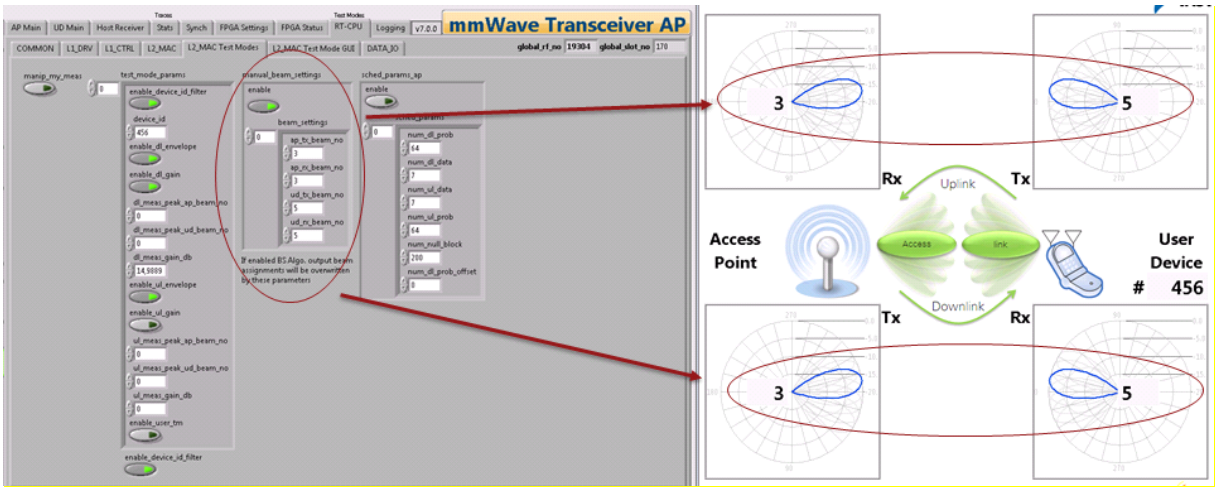


Figure 13: Manual beam steering configuration.

2.1.3 Testbed integration

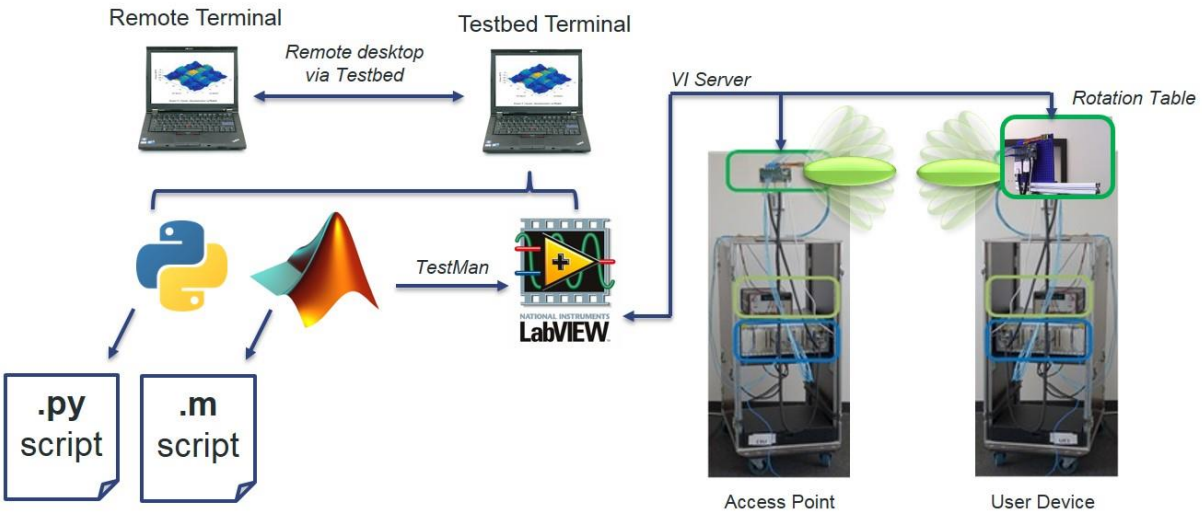


Figure 14: Integration of mmWave with testbed.

The real-time mmWave system is controlled by the user through a computer that executes LabVIEW. This computer is externally accessible via the testbed backbone, where an external user can access the mmWave functionalities via remote desktop connection. In addition, the setup is also integrated with other languages such as Matlab, which makes possible to design flexible and reproducible experiments. This setup is depicted in Figure 14.

2.2 Relation to showcase

The High Throughput mmWave system is related to Showcase 1, which is defined with more details in deliverable D2.3.

2.3 Risk analysis

We identify that the phase noise and RF performance of the 26 GHz antenna can be problematic to the multicarrier GFDM PHY. An alternative to this issue could be switching from multicarrier transmission to single-carrier waveform.

2.4 Implementation plan

2.4.1 PHY improvements

The goal is to use our current functional GFDM PHY with the 26 GHz antenna, where the output of USRP is up-converted with an oscillator. This signal is then fed to the Buttler matrix mmWave antenna. This setup is very advantageous. For instance, the development of algorithms for the new frequency band can be done in an easier manner by using the USRPs, including the development of beam steering algorithms. For the 60GHz mmWave system, no further improvements are planned as the focus has been shifted towards the 26GHz mmWave system.

2.4.2 Higher layer protocol integration

The GFDM-based mmWave system is expected to be integrated to higher layers using the TestMan, which is capable of integrating different applications.

For the 60GHz mmWave system, no further integration is planned as the focus has been shifted towards the 26GHz mmWave system.

2.4.3 Testbed integration

The 28GHz mmWave system will be made available as part of the TUD testbed. The signal processing of the system is executed in the USRP-RIO SDR with a control PC, which is already part of TUD's testbed.

In case, experimenters will conduct experiments including the 60GHz mmWave system, the goal for Year 3 is to support experimenters with the sophisticated HW setup as well as in adaption of parameters needed for the experiment.

© ORCA Consortium 2017-2020

3 SPECTRUM EFFICIENCY IN MULTI-CELL MASSIVE MIMO SYSTEMS

Single-cell massive MaMIMO experiments have been deployed and studied thoroughly, however, there is a lack of experiments which analyse the effects of inter-cell interference in multi-cell scenarios. This section presents an evaluation of spectral efficiency in a virtual two-cells MaMIMO and how different methods can help to boost this metric.

Method 1. Proposed combining vector

Considering a single network which has multiple cells operating in the same frequency. During its up-link transmission, a Base Station receives a signal which contains information of the desired users, but also inter-cell interference and noise. A combining vector coherently combines those signals and extract the required one.

Most of the combining vectors proposed in SoA analyse only intra-cell interference, treating inter-cell interference as noise. Multi-cell Minimum Mean Squared Error (M-MMSE) is the only combining vector which considers inter-cell interference, so far M-MMSE is considered as the optimal scheme achieving the highest SE per cell but with a high computational cost [6].

In this subsection we propose a new combining vector known as Partial M-MMSE (PM-MMSE), which is capable to achieve nearly the highest SE reducing the computational cost, selecting only those inter-cell users with cause strong inter-cell interference to the analysed cell.

Method 2. Cooperation agreement

Considering a multi-cell scenario where two networks deploy their cells adjacently, three different methods of cooperation between them are studied:

- **Two-cells scenario:** A common cellular planning of two cells is considered. Let us assume that each cell works at different carrier frequency and has 10 MHz bandwidth, thus for this case, there is no inter-cell interference
- **Spectrum sharing scenario:** Similar to the previous scenario, however, for this case, we assume that both cells operate at the same frequency with a combined bandwidth of 20 MHz under uncoordinated spectrum sharing. Therefore, inter-cell interference will impact the performance of each cell.
- **Spectrum and infrastructure sharing scenario:** In this case, we consider two cells that are perfectly synchronized, then inter-cell interference is removed. This scenario is also considered as a distributed massive MIMO system where two base stations cooperate with each other serving all the users simultaneously with a total bandwidth of 20 MHz

This subsection shows how this cooperation level impacts the system's spectral efficiency under different combining vectors

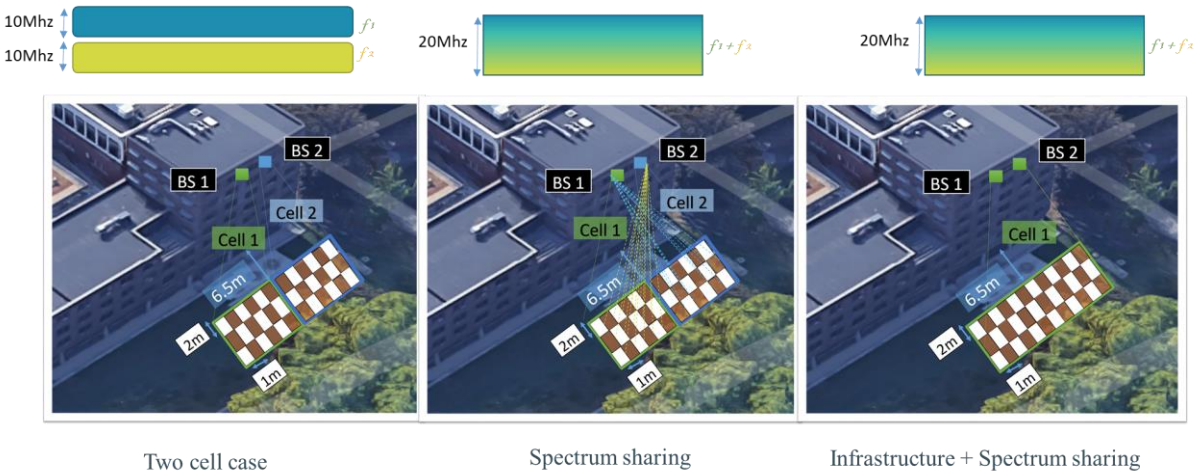


Figure 15: Scenario selection under different levels of cooperation.

3.1 Implementation results

To analyse the spectral efficiency performance in a multi-cell MaMIMO, we carried out a two-cells outdoor experiment deployed in the KULeuven massive MIMO testbed.

3.1.1 PHY improvements

Once the data channel was collected, it was processed offline according to the methodology studied and compared with different combined vectors.

Method 1. Proposed combining vector

PM-MMSE combining vector first selects all inter-cell user channels whose signal-to-interference-ratio (SIR) is above a proposed threshold Γ. So for every *k* inter-cell user its SIR is:

$$SIR_k = \beta_{ref} - \beta_k \leq \Gamma \quad (dB)$$

Where β_{ref} is the reference signal of the user with the best channel gain in the analysed cell. And β_k is the channel gain of the inter-cell user.

The impact of the Γ is presented in Figure 16. So, when $\Gamma \rightarrow -\infty$ no inter-cell channels are considered in PM-MMSE, then its spectral efficiency tends to be similar to a combining vector that does not consider inter-cell interference. On the other hand, when $\Gamma \rightarrow +\infty$ all inter-cell channels are considering in PM-MMSE, then PMMMSE→ M-MMSE.

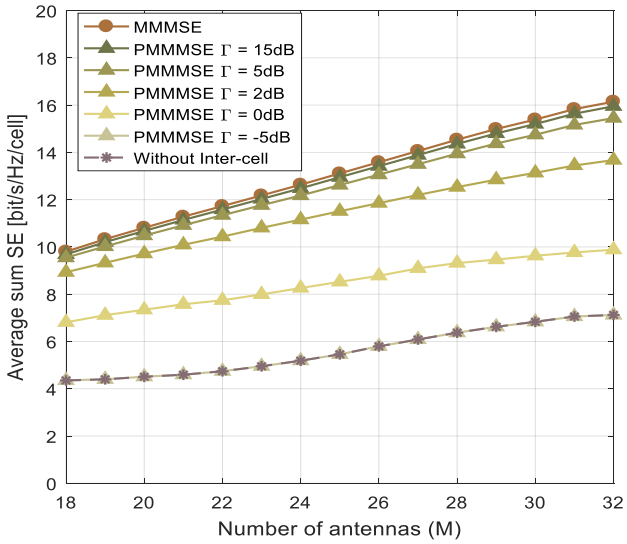


Figure 16: Average UL SE for different SIR thresholds and combining vectors.

In Figure 17 we can see that when our proposed scheme (P-MMSE) selects Γ = 5dB, it achieves a spectral efficiency with a small number of inter-cell users in comparison to the optimal case (M-MMSE), with almost half of the required computational cost.

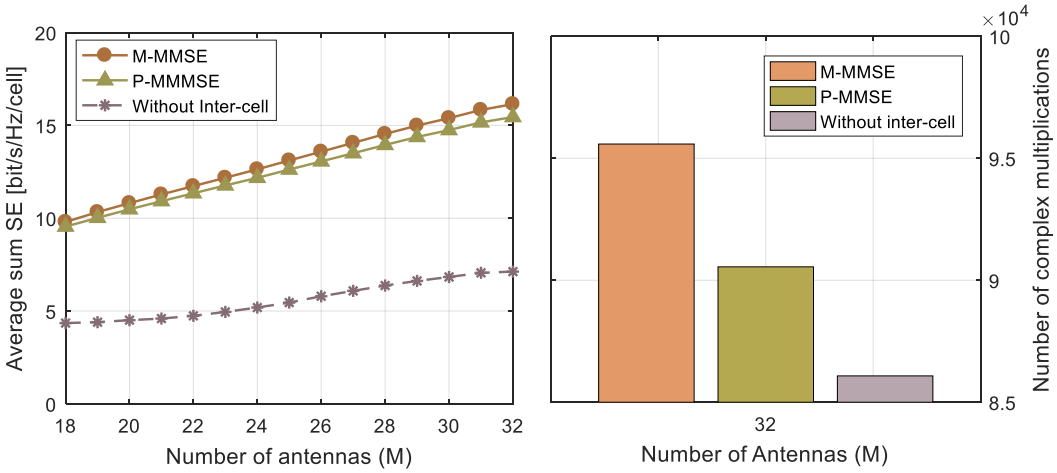


Figure 17: Average UL SE comparison for different combining vectors when PM-MMSE $\Gamma = 5dB$ and its computational cost.

Method 2. Cooperation agreement

Figure 18 shows in general, that the SE increases as the number of active antennas grow, regardless of the combining vector selected.

In the case of spectrum sharing, each cell is affected by inter-cell interference, here M-MMSE uses information of all the inter-cell users which helps to maximize spectral efficiency in contrast with other combining vectors i.e. RZF, MR and ZF. Although M-MMSE boosts the SE in this case, spectrum sharing shows the worst performance with respect to the other scenarios.

Undoubtedly, the case with the best SE per cell is when not only the spectrum is shared but also the infrastructure. Its main advantage is the absence of inter-cell interference as both distributed antenna arrays are synchronized.

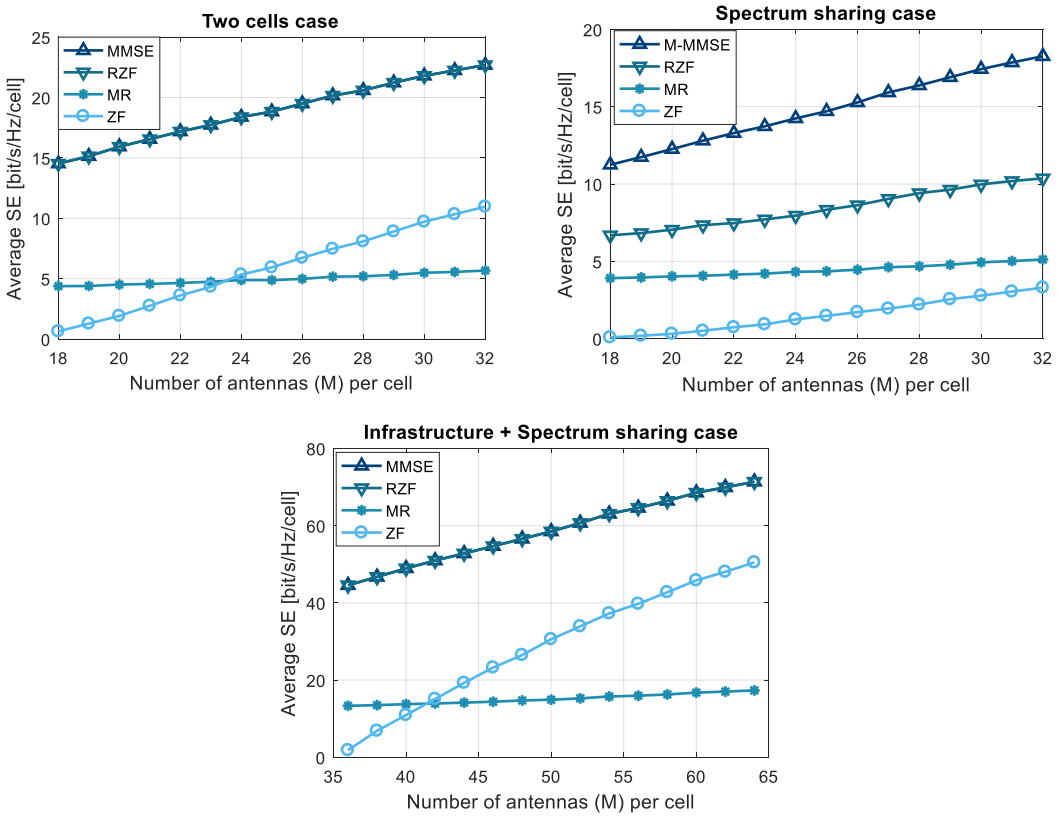


Figure 18: Average spectral efficiency per cell versus the number of active antennas, for different cooperation agreement and under different types of combining vectors in a system.

Due to different cell sizes across the scenarios, the area SE provides a more fair comparison between different scenarios as it is shown in Table 1. For all the metrics, spectrum and infrastructure sharing case provides the best performance as a system. Improving area SE by approximately 50% in comparison with the two-cells case, and almost 100% more for spectrum sharing case when M-MMSE is applied. Following the same trend, an infrastructure and shared scenario increases the capacity six times more than a two-cells case and about four times to the highest achievable value (M-MMSE) for spectrum share case.

Table 1: Network performance metrics for all the study scenarios, with 64 active antennas.

Scenario	Area per cell [m^2]	Bandwidth per cell [MHz]	Spectral Efficiency [bits/Hz/s/Cell]		Area Spectral Efficiency [bits/Hz/s/m^2]		Capacity per cell [Mbits/s]	
			RZF	M-MMSE	RZF	M-MMSE	RZF	M-MMSE
Two Cell	36	10	22.69		0.63		226.9	
Spectrum Sharing	36	20	10.37	18.23	0.288	0.506	207.4	364.4
Infrastructure + Spectrum Sharing	72	20	**71.36**		**0.991**		**1427.2**	

3.1.2 Higher layer protocol integration

Due to the complex framework implemented in our testbed, at the moment user selection for the proposed combining vector PM-MMSE is carried out offline as a high-level PHY. Similarly, to the scenario selection in methodology 2.

3.1.3 Testbed integration

As one of the unique characteristics of this testbed is the distribution of the antennas into two separate patch antenna arrays of equal size, we created a virtual two-cells scenario.

For this experiment, both arrays were placed at the rear windows of KULeuven-ESAT building on the fifth floor at 21 meters high, separated with 6.5 meters from each other and titled 24°. Each array collected real channel measurements and noise from 36 different locations of the user equipment. See Figure 19.

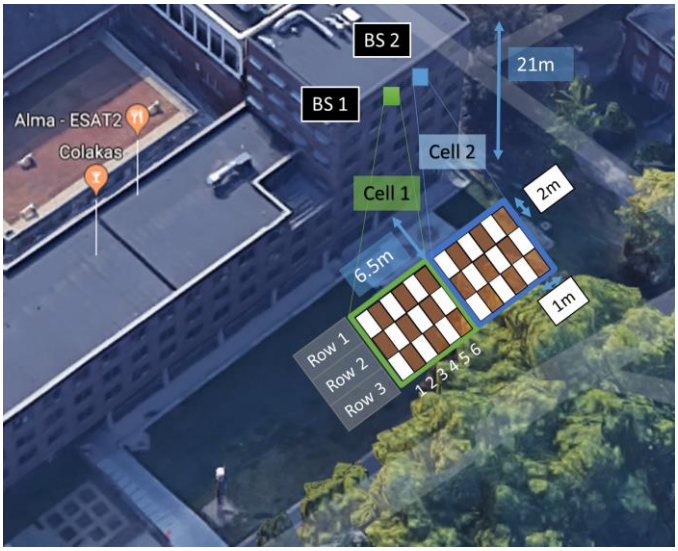


Figure 19: Overlook of the symmetric two-cell Massive MIMO outdoor scenario.

3.2 Relation to showcase

The previous experiment is oriented to SC1: In where each UE could be considering as a static robot and depending on the selected methodology the system can achieve a high spectral efficiency, capacity and throughput.

3.3 Risk analysis

The previous results are scenario-dependent; therefore, more experiments must be carried out with different array locations, as a multi-cell system and with a heterogeneous distribution of the user.

3.4 Implementation plan

The plans to extend the functionalities of KULeuven MaMIMO testbed in the third year are discussed below.

3.4.1 PHY improvements

In the upcoming year we would like to include an extensive analysis of the channel for mobile users in an indoor multi-cell experiment and to validate proposed schemes to reduce inter-cell interference in MaMIMO systems.

3.4.2 Higher layer protocol integration

The Massive MIMO testbed is as such fully bidirectional and closed-loop, i.e., it is possible to transmit and acknowledge in real-time video streams over the setup. Further integration with higher layer protocols is not in the scope of the ORCA project.

3.4.3 Testbed integration

All the data collected in this experiment is available in ORCA website. Remote access to KULeuven is described in D6.3.

© ORCA Consortium 2017-2020

4 FLEXIBLE PHYSICAL LAYER BASED ON GFDM

4.1 Implementation Results

Before carrying out further works on the PHY layer implementation a series of measurements were conducted to identify the performance. The goal is to operate a stable, low latency link for a robot demonstrator as depicted in Figure 20. The first measurement campaign revealed the performance under different indoor channel conditions. The second measurement evaluated the latency of the transceiver and is described in the higher layer integration section.

Figure 20: Robot demonstrator – connecting two robots over a low latency link.

The measurement set-up is depicted in Figure 21 and consists of a USRP-RIO 2953R GFDM transmitter connected to a Channel Emulator based on the National Instruments Vector Signal Transceiver (VST) 5644 running a slight modified channel emulation software from NI. The VST converts the RF signals to the baseband, applies the channel model and up-converts back to RF frequency. A USRP-RIO 2053R receives the signals and applies the GFDM signal processing. The payload was created by an external application using UDP packets to connect to the GFDM transmitter. The packet error rate (PER) depicted in Figure 22 was measured at the application and not at the PHY implementation.

The channel realizations are calculated using MATLAB based on the parameters given in Table 2 and Table 3. The VST control software is receiving the current channel realizations via TestMan from MATLAB and performance the filtering of the signals on the FPGA in real-time.

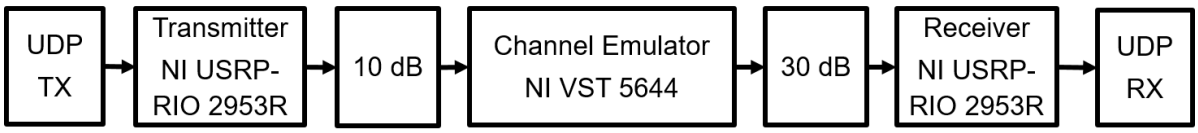


Figure 21: Evaluation set-up using a real-time channel emulator.

Table 2: Channel Model A Indoor Office Rec. ITU-R M.1225.

Tap	Relative delay (ns)	Average Power (dB)
1	0	0
2	50	-3
3	110	-10
4	170	-18
5	290	-26
6	310	-32

The channel model is based on the ITU recommendation for indoor offices since the PHY layer implementation is mainly used in indoor environment. For the demonstrator a WiFi-like configuration was used. Thus, the sampling frequency is set to 20 MHz and the packets contain 36 Bytes. The modulation coding scheme is kept to QPSK and convolutional code with code rate ½. Every second millisecond a packet is sent at 3.75 GHz, a frequency that is allocated for local networks in Germany.

Table 3: Channel emulation parameter.

Parameter	Rayleigh fading	Rician fading
Fading type	Frequency-selective	Frequency-selective
Line-of-sight	NLOS	LOS component on the first path only
Sampling frequency	20 MHz	20 MHz
Maximum Doppler shift	0.001 Hz	0.001 Hz
Fading length	4000 packets	4000 packets
Realizations	100	100

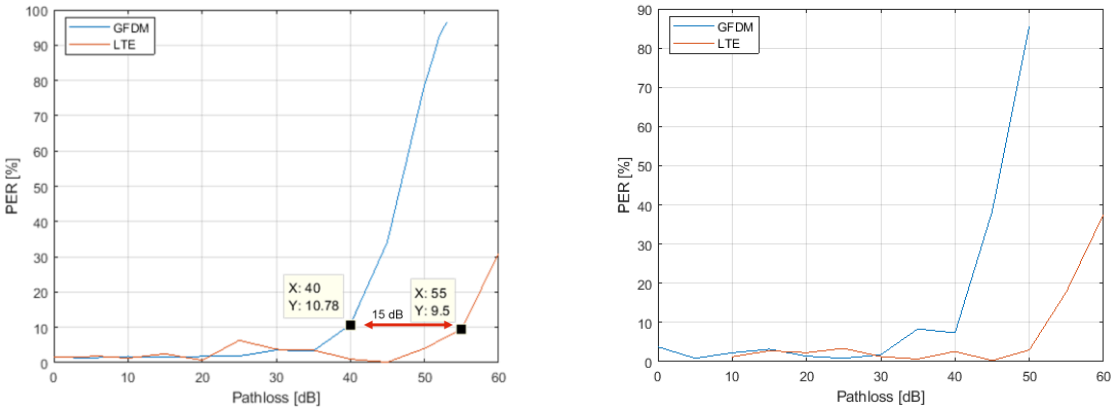
Since, each channel realization is calculated using MATLAB a different realization cannot be applied to every packet, instead for a series of 4000 packets a new channel is tested. After the test with 100 different channel realization with 4000 packets each is performed, the path loss is decreased by 5 dB.

© ORCA Consortium 2017-2020

Table 4: Transmission parameter.

Parameter	Value
Center frequency	3.75 GHz
Transmit power	18 dBm
Packet length	36 Bytes
Repetition	Packet every 2 ms
Clocks	Shared between transmitter and receiver
MCS	QPSK, Convolution Code with Rate ½

The results of the comparison between the GFDM transceiver and the LTE Application Framework from NI are depicted in Figure 22. In both figures there is a 15 dB gap between both implementations, which can be explained that because of the flexible implementation it is difficult to maintain the numerical precision over the wide parameter range. Especially, the used preamble is not using the full 16-bit DAC precision during this measurements, thus the synchronization could not lock in higher path loss regions. In comparison, a fixed PHY implementation, e.g. setting the number of subcarrier, preamble, etc. to fixed values, can achieve higher numerical precision because the headroom can be calculated precisely, whereas in the GFDM implementation a compromise has to be found. A free space path loss of 53 dB is equivalent to 30 meters for 20 dB transmitter gain at 3.75 GHz.



a) Rayleigh fading channel b) Rician fading

Figure 22: Packet error [%] rate as a function of the path loss [dB].

4.1.1 PHY improvements

Since, the performance is not optimal various parts of the PHY design have been changed to improve the transceiver.

4.1.1.1 Automatic Gain Control

Automatic gain control (AGC) is used to calculate the G_{RX} RF gain. This controls the signal amplitude to adjust the signal level at the input of the analog-to-digital converter (ADC) to a suitable value despite fluctuations in the amplitude of the received signal. The average or peak output signal level is used to dynamically adjust G_{RX}, allowing the ADC to operate satisfactorily with a wider range of received signal levels. Our implementation of AGC supports both burst and continuous transmission. A standalone

machine is used to select the signal source used to calculate the G_{RX} gain, where all calculations are performed digitally. Two signal sources are used as shown in Figure 23. The first signal comes directly from the output of the ADC. This source is activated at the beginning of the transmission until the first frame is found. Then the power calculation is switched to the signal at the sync output. While the synchronization can recognize further frames, the AGC continues to use this second source. If no frame is detected after a timeout, the AGC signal source is switched back to the output of the ADC. Consequently, the output of the ADC will most likely be used in burst mode. In continuous mode, the signal source is used after synchronization. The G_{RX} gain is set according to the difference between the measured power and a predefined threshold related to the required dynamic range of the received signal.

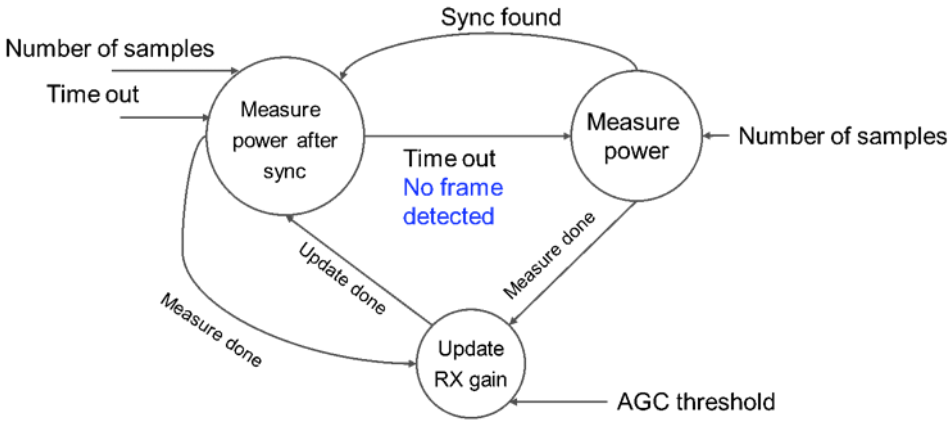


Figure 23: Automatic Gain Control state machine.

4.1.1.2 RF Control Module

The RF control module has been moved from the host computer to the FPGA. In addition, we provide separate control for each of the two RF transceivers on the USRP device. With this modification, we can use any RF transceiver with different sampling rate and frequency. For example, the channel sampling must operate at a sampling rate of 80 MHz, while the modem receiver can operate at a lower sampling rate. Figure 24 shows the RF parameters that can be directly controlled by the FPGA.

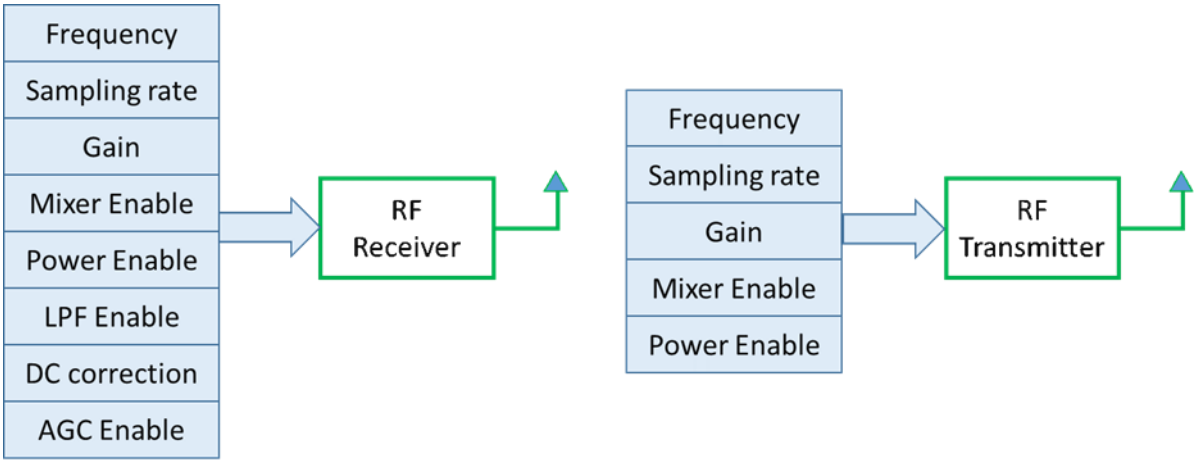


Figure 24: RF Control Module.

4.1.1.3 Scrambling

A common way to reduce peaks and therefore limit the requirements for the numerical precision is to introduce scrambling in the transceiver chain. The implemented scrambler is adding the byte index to each data byte to ensure that the sequence is not uniform. The descrambler subtracts the byte index from each data byte.

4.1.1.4 Interleaving

The interleaver is introduced to spread the data over the whole packet. Since, input data, especially in the GFDM mode, can be seen as a matrix, the interleaving operation can be carried out by transposing this matrix before mapping the data to the QAM symbols.

4.1.1.5 Higher Order QAM Mapper / De-mapper

The modulation-coding scheme was enhanced to 2, 4, 16, 64 and 256 as well as now several code rates can be applied: $\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}$. The mapping/de-mapping is implemented on the FPGA as shown in Figure 25.

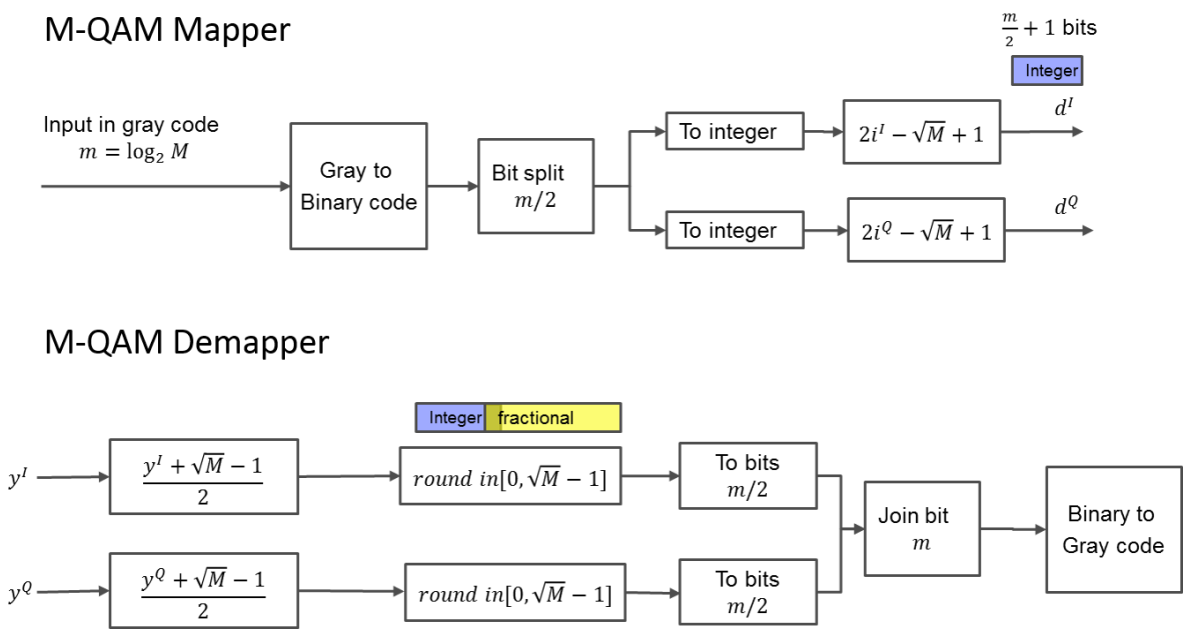


Figure 25: Higher Order QAM Mapper / De-mapper.

4.1.1.6 CRC

A CRC calculation block as depicted in Figure 26 was added to the FPGA to allow a quick detection if a received packet is correct or not and to forward only correct packets to MAC.

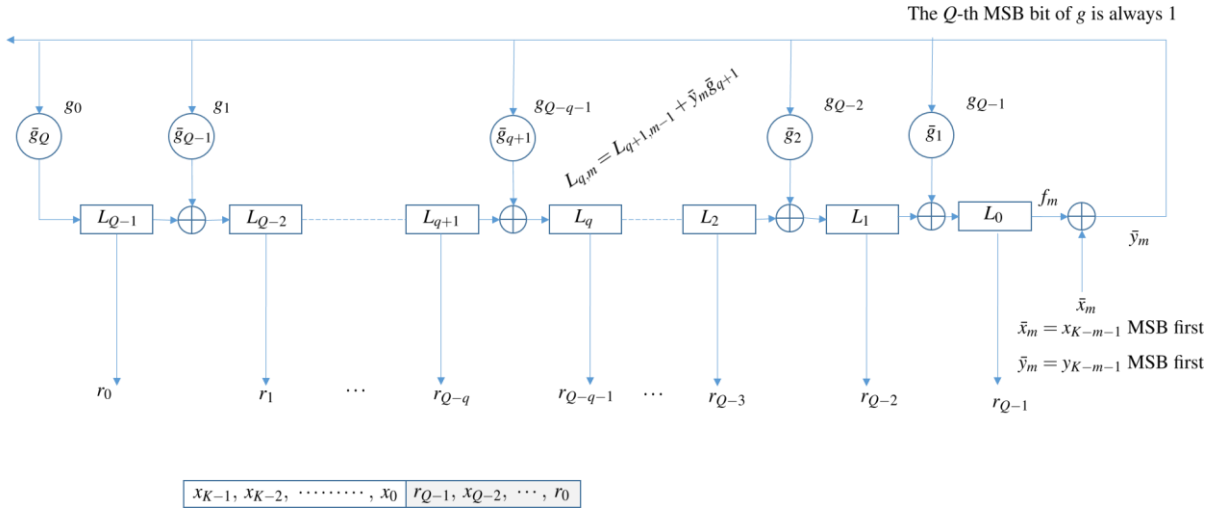


Figure 26: Cyclic Redundancy Check.

4.1.1 Higher layer protocol integration

The goal is to connect the PHY to a robot demonstrator. Prior, the latency was evaluated using a similar set-up as described in the beginning of this section. The host computer is a Hewlett Packard Enterprise (HPE) edgeline m710 blade with a 8 core Intel Xeon E3-1284L @ 3.2 GHz and 64 GB RAM running Windows Server 2012. The UDP test application uses C# and the .NET Framework. Further, all packets are routed to the same UDP test application to ensure a correct measurement of the latency.

In Figure 27 the latency is shown while the UDP packets are transmit and looped back to the same application and the packet rate and length of the packets is varying. Therefore, this figure reveals the latency introduced by the test application itself. Worth mentioning is, that depending on the packet rate the scheduler of the operating system gives more or less priority to the data flow.

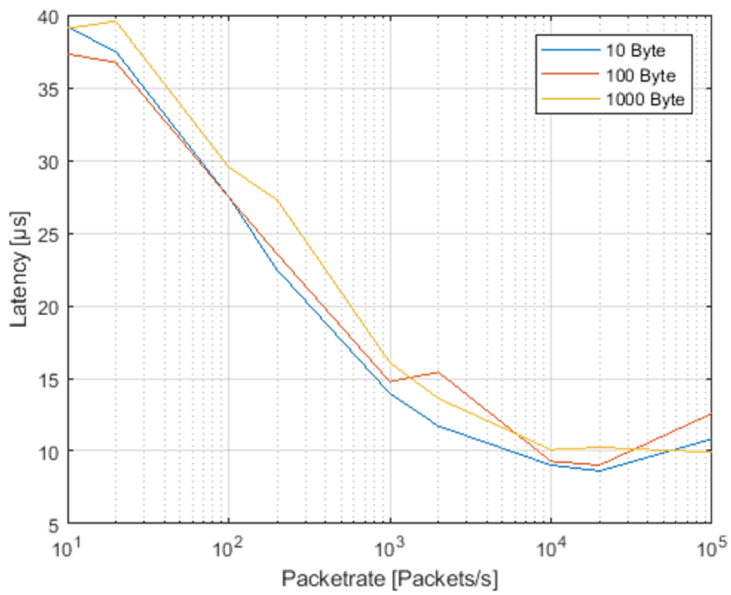


Figure 27: Latency as a function of the packet rate between a UDP sending application and an application receiving UDP packets on the same Windows 2012 server host machine, without network or PHY.

Afterwards the LABVIEW host environment, the PCIe connection and the FPGA is added to the evaluation set-up and the results are shown in Figure 28. The goal is to understand the latency introduced by the software and hardware without the PHY signal processing. Thus, the data is forwarded to the FPGA and immediately send back to the UDP test application. Both results show that minimal latency can be reached by having a sufficient packet rate of around 1 ms or less.

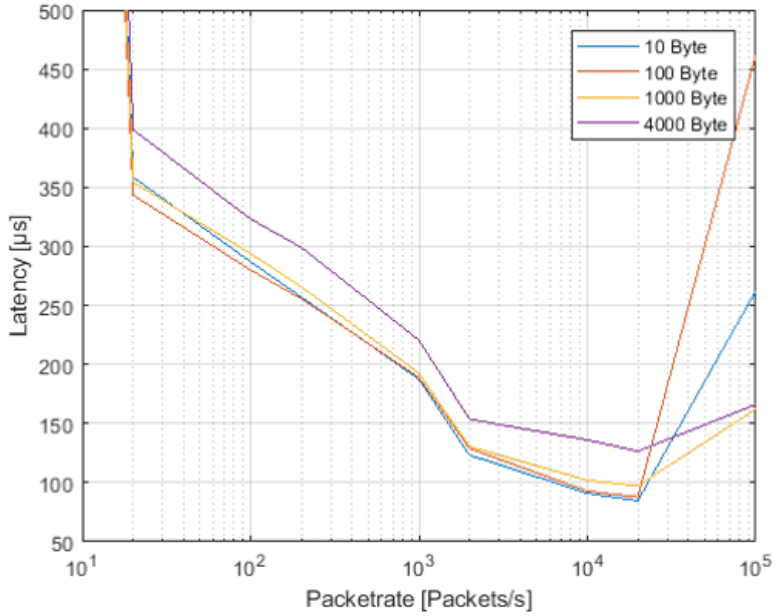


Figure 28: Latency as a function of the packet rate between a UDP sending application and an application receiving UDP packets on the same host machine, including the FPGA bypassing the PHY signal processing on the FPGA. This figure depicts the minimal achievable latency with the hardware set-up, without any signal processing.

Finally, the latency of different PHY implementations running on the same set-up are depicted in Figure 29 and showing that the GFDM transceiver is the fastest among the PHY implementations.

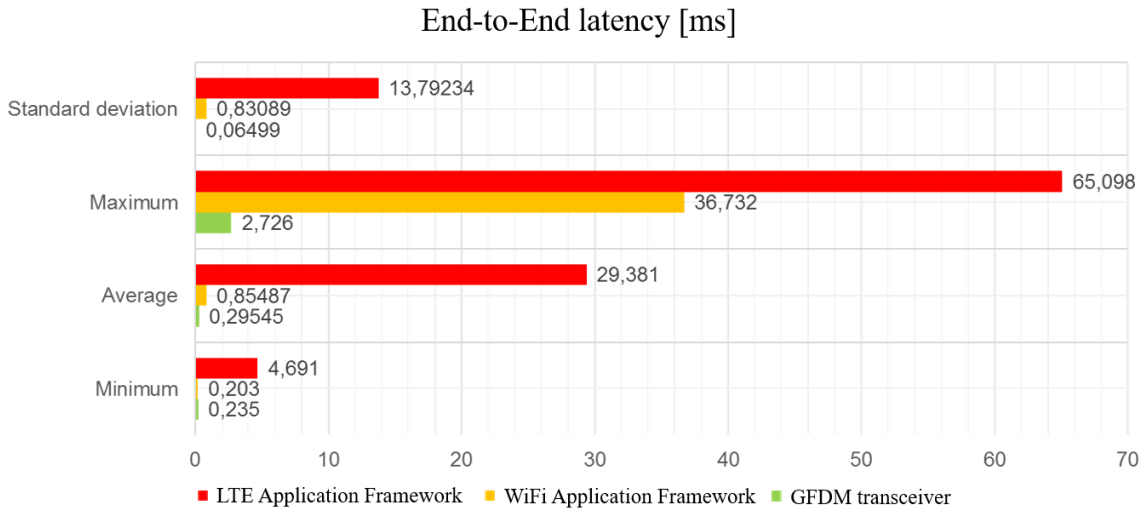


Figure 29: Latency of different PHY implementations.

4.1.2 Testbed integration

The described systems will be made available as part of the TUD testbed. At least one base station will be equipped with a HPE edgeline server and an attached USRP-RIO SDR platform. The user terminals are based on a USRP-RIO SDR with a control PC each and an additional access point will have a NI PXI based chassis. All platforms support the PHY and higher layer capabilities with a varying processing speed.

4.2 Relation to showcase

The presented PHY implementation is relevant for showcase 2 and showcase 4, since it provides low-latency PHY implementation. Further, the system will be evaluated in the year 3 of ORCA using a set-up build out of several FRANKA EMIKA robot arms to understand the influence of a wireless network into the control loops of an industrial robot application.

4.3 Risk analysis

There is no specific risk observed at this phase of the work package.

4.4 Implementation plan

In year 3 of ORCA, GFDM will be evaluated in scenarios with multiple users. Besides conventional TDMA with the current implementation, also novel forms of FDMA are going to be analysed, especially, in context with 5G mixed numerology and network slicing. These ideas have also influence on the PHY layer besides applying flexible parameter sets. In the next section, a partially realized PHY implementation is presented that allows very different waveforms or numerology being multiplexed in frequency domain. Main difference is that a single implementation can handle the numerologies as part of the signal processing in the modem without using dedicated multiple modems followed by a filter or FFT to combine the different signals.

4.4.1 PHY improvements

In this section, we represent GFDM by means of the discrete Zak transform [7]. This reflects the involvement of TD and FD-shifted modulation pulses. It also clarifies the structure and facilitates the implementation. The basic TD-GFDM equation can be reformulated in a matrix form by using two indexes $q = 0,\cdots,K-1$ and $p = 0\cdots,M-1$, such that $n = q + pK$. Thereby,

$$[\boldsymbol{x}]_{(q+pK)} = \sum_{m=0}^{M-1}\sum_{k=0}^{K-1} d_{k,m} g[< q + pK - mK >_N] e^{j2\pi \frac{k}{K} q}$$

For a vector $a \in \mathbb{C}^{QP\times 1}$, we define the matrix operator,

$$\boldsymbol{V}_{Q,P}^{(\boldsymbol{a})} = \mathrm{unvec}_{P\times Q}\{\boldsymbol{a}\}^T \Leftrightarrow \left[\boldsymbol{V}_{Q,P}^{(\boldsymbol{a})}\right]_{(q,p)} = [\boldsymbol{a}]_{(p+qP)}\,.$$

Using this notation, the TD-GFDM block can be expressed as

$$\left[\boldsymbol{V}_{M,K}^{(\boldsymbol{x})}\right]_{(p,q)} = \sum_{m=0}^{M-1}\left[\boldsymbol{V}_{M,K}^{(\boldsymbol{g})}\right]_{(<p-m>_M,q)}\left[\boldsymbol{D}^T\boldsymbol{F}_K^H\right]_{(m,q)}\,.$$

This defines circular convolution between the q-th column of $V_{M,K}$ and the q-th column of D, and it can be expressed with M-DFT as

$$\left[\boldsymbol{F}_M \boldsymbol{V}_{M,K}^{(\boldsymbol{x})}\right]_{(p,q)} = \left[\boldsymbol{F}_M \boldsymbol{V}_{M,K}^{(\boldsymbol{g})}\right]_{(q,q)} \cdot \left[\boldsymbol{F}_M \boldsymbol{D}^T \boldsymbol{F}_K^H\right]_{(p,q)},$$

Using Zak transform notation, $Z_{M,K} = \boldsymbol{F}_Q V_{Q,P}^{(\boldsymbol{a})}$, as shown in Fig. 30, we get

$$\boldsymbol{V}_{M,K}^{(\boldsymbol{x})} = \frac{1}{MK} \boldsymbol{F}_M^H \left(K \boldsymbol{Z}_{M,K}^{(\boldsymbol{g})} \odot \left[\boldsymbol{F}_M \boldsymbol{D}^T \boldsymbol{F}_K^H\right]\right).$$

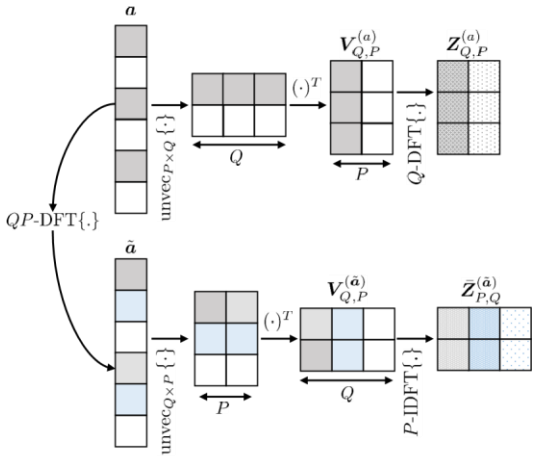


Figure 30 Zak transform.

The demodulator performs the inverse steps. Let y_{eq} be the TD equalized signal, then

$$\hat{\boldsymbol{D}} = \frac{1}{M} \boldsymbol{F}_K \left(\boldsymbol{W}_{\mathrm{rx}} \odot \left[\boldsymbol{F}_M \boldsymbol{V}_{M,K}^{(\boldsymbol{y}_{\mathrm{eq}})}\right]^T\right) \boldsymbol{F}_M^H.$$

Here $W_{rx} \in C^{K\times M}$ is the receive window corresponding to the demodulator prototype pulse. Accordingly, the demodulation convolution in the TD is given by

$$\left[\frac{1}{K} \hat{\boldsymbol{D}}^T \boldsymbol{F}_K^H\right]_{(p,q)} = \sum_{m=0}^{M-1} \left[\boldsymbol{V}_{M,K}^{(\tilde{\boldsymbol{\gamma}})}\right]_{(<p-m>_M,q)} \left[\boldsymbol{V}_{M,K}^{(\boldsymbol{y}_{\mathrm{eq}})}\right]_{(m,q)}$$

where $\boldsymbol{V}_{M,K}^{(\tilde{\boldsymbol{\gamma}})} = \frac{1}{M} \boldsymbol{F}_M^H \boldsymbol{W}_{\mathrm{rx}}^T$. A dual FD representation can be derived in a similar way. Table 5 summarizes the GFDM modem equations for both TD and FD representations.

Our proposed architecture is developed based on the Zak transform representation of the GFDM samples. The TD representation can be reformulated as

$$\boldsymbol{V}_{M,K}^{(\boldsymbol{x})} = \frac{1}{M} \boldsymbol{F}_M^H \left(\boldsymbol{W}_{\mathrm{tx}}^T \odot \left(\boldsymbol{F}_M \left[\frac{1}{K} \boldsymbol{F}_K^H \boldsymbol{D}\right]^T\right)\right),$$

and the FD representation can be reformulated as

$$\boldsymbol{V}_{K,M}^{(\tilde{\boldsymbol{x}})} = \boldsymbol{F}_K \left(\boldsymbol{W}_{\mathrm{tx}} \odot \left(\frac{1}{K} \boldsymbol{F}_K^H \left[\boldsymbol{F}_M \boldsymbol{D}^T\right]^T\right)\right).$$

Both equations have similar structure with simple differences. In the TD, the data symbols are fed to the modulator by the columns of D. Then, a K-IDFT is computed for each column, and the result of N samples need to be stored in a matrix $\boldsymbol{D}$ of size $K\times M$. Afterwards, the samples of $\boldsymbol{D}$ are forwarded rowby-row to an M-DFT block. The output of the M-DFT can be directly element-wise multiplied with the rows of the stored modulation window W_{tx}. The result is then fed to an M-IDFT block and the output

is stored in the columns of the matrix $\boldsymbol{V}_{M,K}^{(x)}$. Finally, the core block is generated by reading $\boldsymbol{V}_{M,K}^{(x)}$ in rows. The FD modulator, works similarly by feeding the data of D in rows, and replacing the K by M and the DFT by inverse discrete Fourier transform (IDFT). The resulting FD samples are stored in a matrix $\boldsymbol{V}_{M,K}^{(\tilde{x})}$ and read by rows. Additionally, the FD modulator requires a final N-DFT to transform the block to the TD.

Table 5: Summary of GFDM equations.

	Time domain	Frequency domain
	Modulation	
Conv.	$\left[\boldsymbol{V}_{M,K}^{(\boldsymbol{x})}\right]_{(p,q)} = \sum_{m=0}^{M-1}\left[\boldsymbol{V}_{M,K}^{(\boldsymbol{g})}\right]_{(<p-m>_M,q)}\left[\boldsymbol{D}^T\boldsymbol{F}_K^H\right]_{(m,q)}$	$\left[\boldsymbol{V}_{K,M}^{(\tilde{\boldsymbol{x}})}\right]_{(q,p)} = \sum_{k=0}^{K-1}\left[\boldsymbol{V}_{K,M}^{(\tilde{\boldsymbol{g}})}\right]_{(<q-k>_K,p)}\left[\boldsymbol{D}\boldsymbol{F}_M\right]_{(k,p)}$
Zak trans.	$\boldsymbol{V}_{M,K}^{(\boldsymbol{x})} = \frac{1}{MK}\boldsymbol{F}_M^H\left(\boldsymbol{W}_{\text{tx}}\odot\left[\boldsymbol{F}_K^H\boldsymbol{D}\boldsymbol{F}_M\right]\right)^T$	$\boldsymbol{V}_{K,M}^{(\tilde{\boldsymbol{x}})} = \frac{1}{K}\boldsymbol{F}_K\left(\boldsymbol{W}_{\text{tx}}\odot\left[\boldsymbol{F}_K^H\boldsymbol{D}\boldsymbol{F}_M\right]\right)$
Tx win.	$\boldsymbol{W}_{\text{tx}} = K\boldsymbol{Z}_{M,K}^{(\boldsymbol{g})^T}$	$\boldsymbol{W}_{\text{tx}} = K\bar{\boldsymbol{Z}}_{K,M}^{(\tilde{\boldsymbol{g}})}$
	Demodulation	
Zak trans.	$\hat{\boldsymbol{D}} = \frac{1}{M}\boldsymbol{F}_K\left(\boldsymbol{W}_{\text{rx}}\odot\left[\boldsymbol{F}_M\boldsymbol{V}_{M,K}^{(\boldsymbol{y}_{\text{eq}})}\right]^T\right)\boldsymbol{F}_M^H$	$\hat{\boldsymbol{D}} = \frac{1}{M}\boldsymbol{F}_K\left(\boldsymbol{W}_{\text{rx}}\odot\left[\frac{1}{K}\boldsymbol{F}_K^H\boldsymbol{V}_{K,M}^{(\tilde{\boldsymbol{y}}_{\text{eq}})}\right]\right)\boldsymbol{F}_M^H$
Conv.	$\left[\frac{1}{K}\hat{\boldsymbol{D}}^T\boldsymbol{F}_K^H\right]_{(p,q)} = \sum_{m=0}^{M-1}\left[\boldsymbol{V}_{M,K}^{(\hat{\boldsymbol{\gamma}})}\right]_{(<p-m>_M,q)}\left[\boldsymbol{V}_{M,K}^{(\boldsymbol{y}_{\text{eq}})}\right]_{(m,q)}$	$\left[\hat{\boldsymbol{D}}\boldsymbol{F}_M\right]_{(q,p)} = \frac{1}{K}\sum_{k=0}^{K-1}\left[\boldsymbol{V}_{K,M}^{(\tilde{\hat{\boldsymbol{\gamma}}})}\right]_{(<q-k>_K,p)}\left[\boldsymbol{V}_{K,M}^{(\tilde{\boldsymbol{y}}_{\text{eq}})}\right]_{(k,p)}$
Rx win.	$\boldsymbol{V}_{M,K}^{(\hat{\boldsymbol{\gamma}})} = \frac{1}{M}\boldsymbol{F}_M^H\boldsymbol{W}_{\text{rx}}^T$	$\boldsymbol{V}_{K,M}^{(\tilde{\hat{\boldsymbol{\gamma}}})} = \boldsymbol{F}_K\boldsymbol{W}_{\text{rx}}$

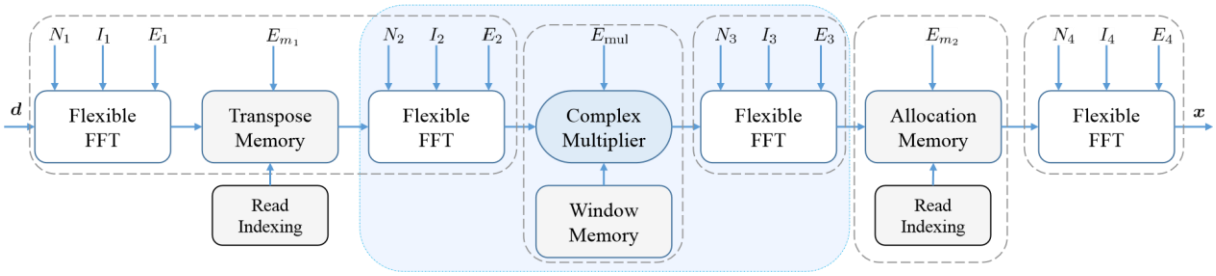


Figure 31: Unified architecture for TD and FD GFDM processing.

The radix-2 values of K and M enables the implementation of DFT with flexible FFT intellectual property (IP) cores, e.g. Xilinx FFT. This core allows the run-time reconfiguration of the size of DFT and setting the block either in DFT or IDFT mode. The proposed architecture is illustrated in Fig. 5. In this design, 4 flexible FFT cores with the configuration parameter N_x for the DFT size and I_x to set the core in the direct [D] or inverse [I] mode. Additionally, each FFT block can be enabled [E] or disabled [D] with the parameter E_x, such that the disabled block forwards the samples to the next stage. The 4-th FFT core is optional depending on the desired domain of implementation. Moreover, one memory block is used to store the result of the first transform and performs the transpose by the configuration of the indexing unit. The modulator window is stored in a memory as part of the modulator configuration. This memory is always read incrementally and written with respect to the implementation domain. Furthermore, one high throughput complex multiplier is used to perform the element-wise multiplication. A third memory is required to store the samples prior to generating the final block by performing the transpose. All the building blocks can be disabled as well. Actually the highlighted box corresponds to the computation of the convolution. Table 6 summarize the configuration parameters of the modulator. The demodulator has similar architecture in reversed order.

© ORCA Consortium 2017-2020

Table 6: Configuration parameters for modulator.

Parameter	TD	FD
$[N_1,N_2,N_3,N_4]$	$[K,M,M,-]$	$[M,K,K,N]$
$[I_1,I_2,I_3,I_4]$	$[I,D,I,-]$	$[D,I,D,I]$
$[E_1,E_2,E_3,E_4]$	$[E,E,E,D]$	$[E,E,E,E]$
Window	$W_{\mathrm{tx}T}$	W_{tx}
allocation	$\{\cdot\}^T$	$\{\cdot\}^T$

The N_4-FFT block is enabled or disabled depending on the domain of the equalized signal and the configuration of the demodulator. For instance, if the input is an FD equalized signal and the modulator is configured in TD, this block needs to be configured as N -IFFT. The allocation memory is used to store the received signal in columns and forward it in rows to the next block. The samples at the output of the demodulator correspond to the columns of $\hat{\boldsymbol{D}}$ in the TD and to the rows in the FD configuration. The demodulator settings with respect to FD equalization are listed in Table 7.

Table 7: Demodulator configuration, FD equalization.

Parameter	TD	FD
deallocation	$\mathrm{unvec}_{K\times M}\{\cdot\}^T$	$\mathrm{unvec}_{M\times K}\{\cdot\}^T$
Window	$W_{\mathrm{rx}T}$	W_{rx}
$[N_4,N_3,N_2,N_1]$	$[N,M,M,K]$	$[-,K,K,M]$
$[I_4,I_3,I_2,I_1]$	$[I,D,I,D]$	$[-,I,D,I]$
$[E_4,E_3,E_2,E_1]$	$[E,E,E,E]$	$[D,E,E,E]$

Usually, FD equalization is used. Thus, the modem can be configured as TD modulator and FD demodulator. In this way, the overall modulation and equalization processing requires only one N -FFT for the equalizer. On the other hand, including N_4-FFT preserves the symmetry of the architecture. Therefore, with fast reconfiguration, the modulator can be reconfigured as demodulator. This solution is typical for low- cost time division duplex transmission. The bypass function of each block allows faster processing of certain waveforms. For instance, OFDM requires only one IFFT to generate TD samples. DFT-spread-OFDM requires only a single FFT in the FD. Additionally, one or more FFT cores can be disabled for certain precoding methods like in [8]. Furthermore, N_4- FFT core can be configured with $N_4 > N$ to produce over- sampled signal. The allocation indexing can be altered to implement sort of frequency division multiple access (FDMA) and multiplex pilot samples as shown in deliverable D4.3.

4.4.2 Higher layer protocol integration

The integration of the PHY implementations is described in deliverable 4.3.

4.4.3 Testbed integration

The described implementation will be made available as part of the TUD testbed. The implementation will run on the XILINX FPGA inside USRP-RIO SDR. All platforms support the PHY and higher layer capabilities with a varying processing speed.

5 HYBRID FPGA PLATFORM INCL. FLEXIBLE MAC

5.1 Implementation results

5.1.1 PHY improvements

5.1.1.1 Zero Tx/Rx turnaround time on SDR for better latency performance

When implementing a wireless standard that uses Time Division Duplex (TDD), the SDR needs to be switched regularly between Receiving (Rx) and Transmitting (Tx) modes. The switching time or Turnaround Time (TT) is defined as the time required by the PHYsical (PHY) layer to change from Rx mode to Tx mode or vice versa. During TT, the components in the analog RF front-end are powered up and stabilized, which consumes a considerable amount of time. This time consumption becomes more critical for low latency feedback applications, such as process control loops in industrial systems where remote control of robotic arms or other machinery is involved. The influence of TT on latency of a wireless system in a simple Device-to-Device (D2D) topology is illustrated in Figure 32.

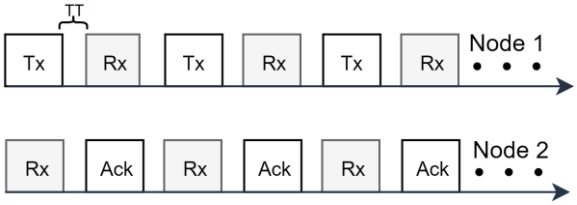


Figure 32: The impact of turnaround time on latency during TDD operation in a D2D scenario.

It is evident that during TT there is no data transfer, and the wireless channel capacity is wasted. Normally, the negative impact of TT becomes more severe when smaller packets and faster switching scenarios are considered. For example, a special technique proposed in [9] actually leverages the fast and frequent Tx/Rx switching to achieve "virtual Full Duplex". In this solution, a Tx or Rx slot is in the order of microseconds, and only a few physical layer symbols instead of a complete packet is transferred in a slot. Since the Tx/Rx switching occurs very frequently, it appears as if both nodes are transmitting and receiving concurrently, creating the perception of real Full Duplex (FD) realized with a Half-Duplex (HD) implementation. This solution is however based on the assumption that TT is in the order of nanoseconds, and is only validated in a simulation environment. Unfortunately, TT of an SDR front-end is generally much higher (i.e., ≈18 µs in case of AD9361 [10]). AD9361 introduced by Analog Devices is an RF transceiver chip widely used on SDR RF front-ends, including several devices (e.g., B200 and E310) in the popular USRP family [11], and a range of the FMCOMMS boards [10], [12], [13]. As such, a short TT cannot be taken for granted on the current mainstream SDR platforms. This is one of the main differences between SDR and commercial transceiver chipsets today. In this chapter, a novel approach called BBHD-RFFD – BaseBand processing unit operating in Half Duplex mode and analog Radio Frequency front-end operating in Full Duplex mode – is introduced to reduce the TT on an SDR RF-FE.

The conventional way to configure the RF front end for TDD operation (referred to as conventional TDD approach hereafter), is presented in Figure 33-a for the ease of comparison and discussion, whereas Figure 33-b illustrates our proposed approach. Both figures have the basic blocks in the radio communication stack, namely the MAC, PHY and RF front-end. For experimental validation we have implemented MAC and PHY on a "System on Chip" hardware architecture. More information on this implementation is provided in the next paragraph. The exact way how MAC and PHY are realized, is not relevant for the BBHD-RFFD approach, the only thing that matters is the control path (blue lines) towards the "RF Front-End". In a conventional TDD approach, the respective RF front-end components

are switched on or off upon transition between Tx and Rx modes, as indicated by ① in Figure 33-a. Since most of the RF components are analog, they consume more time to be turned on and stabilized than digital components. The turn-on time in combination with their software commands considerably contribute to the overall TT of a wireless system.

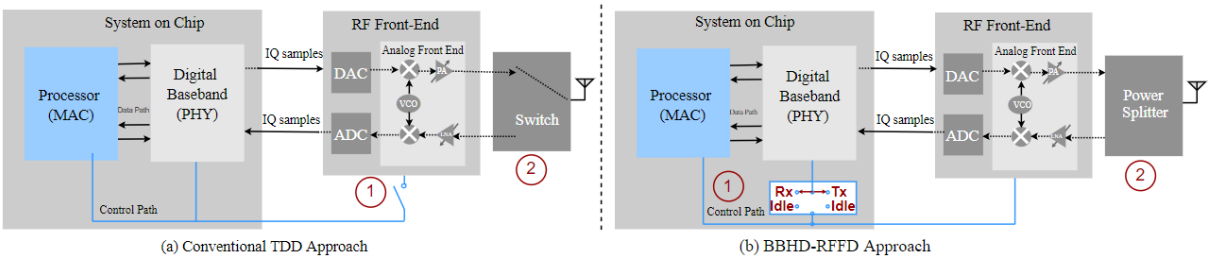


Figure 33: Block diagram of (a) the conventional TDD and (b) the proposed BBHD-RFFD approaches.

In the BBHD-RFFD approach, the RF components of transmitter and receiver remain on all the time, thus no time is required to turn on the analog components. Apparently, this configuration comes with more power consumption, therefore we also introduce Power Saving Scheme (PSS) in the BBHD-RFFD approach. More information on the PSS can be found in Section V-D. In this case, switching only occurs in digital baseband domain, as indicated by ① in Figure 33-b. In general, physical layer functionality is comprised of multiple Baseband Processing Units (BPUs). Essentially, the Tx/Rx switching comes down to activating the respective BPUs either in the Tx or the Rx path. The software controlling the switching ensures the HD operation by enabling only one set of BPUs at a time. Since Tx and Rx paths have separate BPUs, the respective path can be attached to the corresponding In-phase and Quadrature phase (IQ) samples of the radio board. For instance, in Tx mode, BPUs related to Tx path are attached to IQ samples of DAC, meanwhile, BPUs related to Rx path are detached from IQ samples of ADC. Thanks to microelectronics, the time consumed for digital switching action takes a few nanoseconds. Our digital system is operating at 100MHz, therefore it takes only 10 ns to switch between Tx and Rx modes. The proposed scheme is however not applicable for most single-standard commercial off-the-shelf chips. For instance, the IEEE 802.15.4 compliant chip CC2538 cannot turn on Tx and Rx related RF components at the same time [14], hence it can only support the conventional TDD approach.

The BBHD-RFFD approach is realized on an off-the-shelf SDR hardware platform as follows. The FMCOMMS2 [10] board with AD9361 chipset is exploited as SDR RF-FE, while IEEE 802.15.4 [15] standard compliant PHY and ALOHA MAC layer are implemented in the programmable logic (PL) and processor system (PS) parts of a Zynq-7000 All Programmable System on Chip (AP SoC) (i.e., ZedBoard in our setting) [16]. The FMCOMMS board is usually accompanied by a Xilinx FPGA board to form a complete SDR platform, which is also the case in our setup. The FMCOMMS2 board has separate Tx and Rx antenna ports, a Mini-Circuits' ZN2PD2-63+ power splitter [17] is used to attach both Tx and Rx ports to the antenna, as shown at ② in Figure 33-b, without using an antenna switch as shown at ② in Figure 33-a. The function of the power splitter is twofold, (i) it allows us to transmit or receive the RF signal via a single antenna; and (ii) it gives an isolation of more than 20 dB between the Tx and Rx ports. This isolation is important to our solution because when leaving the RF components constantly activated, the Rx path is impacted by self-interference even when the Tx BPUs are deactivated. In this paper, the DC leakage and internal noise of various components in both transmit and receive path are collectively defined as self-interference. The power splitter is a low cost and effective approach to mitigate the impact of self-interference, as is further detailed in the following paragraphs.

We investigate how the different operation modes of the AD9361 chipset can be used for the conventional TDD and the BBHD-RFFD approaches. The AD9361 can be operated in the following three duplex modes:

1) TDD WithOut Calibration (TDD-WOC): In this mode, RF components are calibrated once during the initialization phase, afterwards the components specific to Tx or Rx chains are simply turned on without calibration. Calibration refers to activities such as tuning the Voltage Controlled Oscillator (VCO) and waiting for the Phase Locked Loop (PLL) to stabilize around the desired output frequency.

2) TDD With Calibration (TDD-WC): In this mode, all the RF components are turned on and then calibrated every time when switching between Tx and Rx occurs.

3) Frequency Division Duplex (FDD): In this mode, all the RF components are turned on and calibrated during initialization in FDD mode, and all the RF components pertaining to Rx and Tx chains remain operational afterwards. Since both Tx and Rx paths remain on all the time, there is no need of switching (Tx to Rx or vice versa) in FDD mode.

Both TDD-WC or TDD-WOC can be exploited to implement the conventional TDD approach. It is recommended to repeat calibration in AD9361 each time the operating channel frequency is changed. Thus, TDD-WC should be employed in a situation where regular change of Tx/Rx frequency occurs (e.g., Bluetooth). The FDD mode of AD9361 is exploited to implement the BBHD-RFFD approach.

In addition to the three duplex modes, the 'DC offset calibration and tracking feature' of AD9361 helps to minimize the DC offset in the received IQ samples, while the 'quadrature calibration and tracking' feature aims to maintain orthogonal relation between the received I and Q samples by performing IQ correction [18]. The aforementioned calibration and tracking features of AD9361 are hereinafter referred to as DC/IQ tracking, which is further explored in experiments to suppress the DC component in the self-interference.

In this section, we first measure the amount of self-interference under different settings, and derive the optimal setting to implement the BBHD-RFFD approach, and then experiments are conducted to benchmark our proposed approach against the conventional TDD approach, in terms of performance in TT and receiver sensitivity.

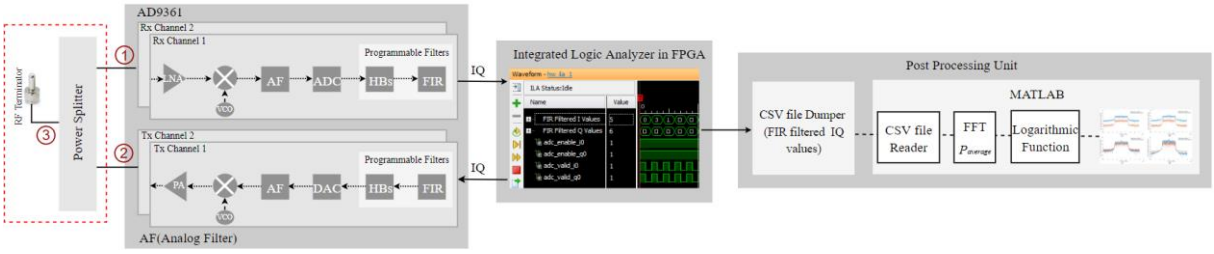


Figure 34: Experimental setup to capture IQ samples from AD9361.

The above figure shows the setup utilized to measure the self-interference under different conditions and configuration of AD9361, an RF front-end in the FMCOMMS2 board used in our experimental setting. All measurements in this section use 2.41 GHz as a center frequency. AD9361 has two identical and independently controlled transmit and receive paths, which are both shown in the figure above, though only one set of Tx/Rx channel is used in the measurement. The measurement is performed in the following 3 steps:

1) Unintended RF signals (i.e., self-interference) pass through Rx chain of AD9361. In this chain, the I and Q samples obtained at the output of ADC, operating at 8 Msps, are filtered by a low pass Finite Impulse Response (FIR) filter with cutoff frequency of 2.6 MHz.

2) Integrated Logic Analyzer (ILA), a logic analyzer core that can be used to monitor the internal signals of a design, in FPGA (the PL part of Zynq 7000 SoC) is used to capture the filtered IQ samples.

3) Post processing of the collected IQ samples is done in MATLAB (a high-performance computer software for technical computing), to obtain the Power Spectrum Density (PSD) and the average power of the IQ samples.

The self-interference when no transmission takes place has been analysed with PSD measurements for the TDD-WOC and FDD modes of AD9361 for the following 4 cases:

1) Case 1 (see Figure 35-a): DC/IQ tracking is disabled, Tx and Rx ports are directly connected without any power splitter.

2) Case 2 (see Figure 35-b): DC/IQ tracking is enabled, Tx and Rx ports are directly connected without any power splitter.

3) Case 3 (see Figure 35-c): DC tracking is enabled, Tx and Rx ports are connected via power splitter, the 3rd port of power splitter is terminated, to prevent external interference, as shown in the dashed red rectangular of Figure 34.

4) Case 4 (see Figure 35-d): DC tracking is enabled, Tx and Rx ports are attached to two antennas, positioned orthogonal to each other. Extra measures have been taken to ensure that the operational band is clean at the time of the experiment.

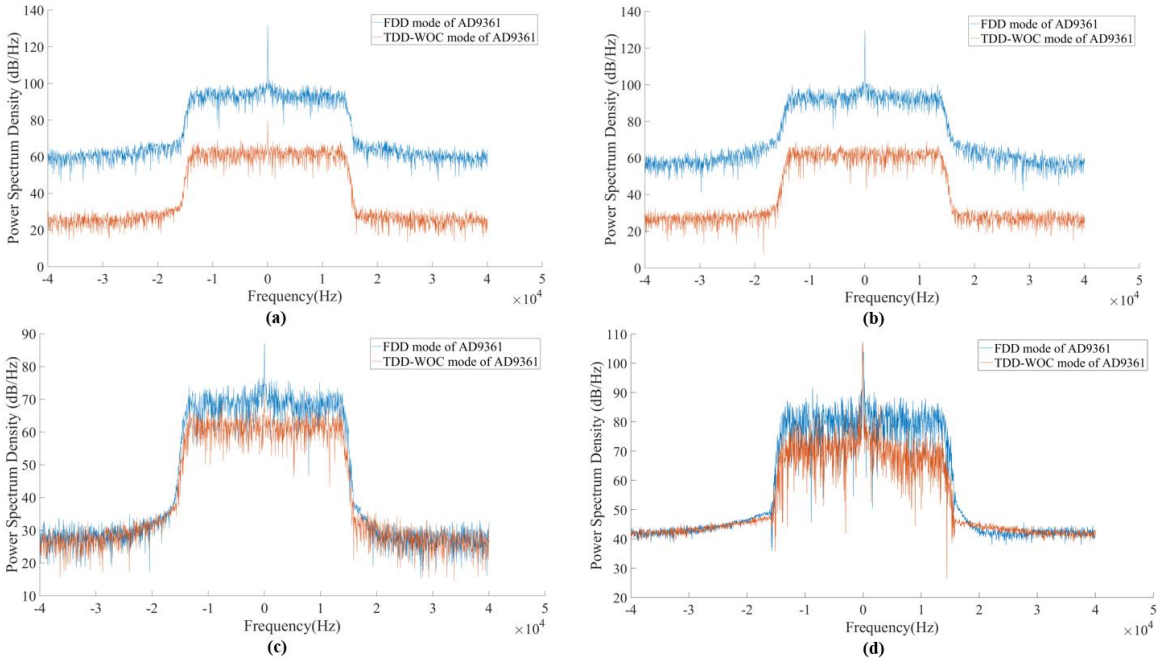


Figure 35: The comparison of self-interference (visualized by PSD), between FDD and TDD-WOC modes of AD9361, under the following settings: (a) Tx and Rx ports are directly connected and DC tracking is disabled; (b) Tx and Rx ports are directly connected and DC tracking is enabled; (c) Tx and Rx ports are connected through a power splitter with DC tracking enabled; (d) Tx and Rx ports are connected to two antennas, placed orthogonal to each other.

As there is no self-interference in the TDD-WOC mode, the average power difference between FDD and TDD-WOC represents the intensity of the self-interference. This is calculated as follows:

$$P_{diff} = P_{fdd} - P_{tdd\text{-}woc}$$

where P_{fdd} and $P_{tdd-woc}$, denote average power P_{avg} obtained during FDD and TDD-WOC modes, respectively. The for all the aforementioned four cases are listed in the table below.

Table 8: Self-Interference measured by power difference in the 4 cases.

Case	P_{fdd} (dB)	$P_{tdd\text{-}woc}$ (dB)	P_{diff} (dB)
Case 1	65.75	26.84	38.9
Case 2	64.43	26.85	37.6
Case 3	34.65	26.45	8.2
Case 4	47.23	44.03	3.2

Among the 4 cases, case 1 is used as a benchmark, as it does not involve any effort to remove self-interference. When comparing case 2 with case 1, the self-interference of FDD mode is reduced by 1.3dB, thanks to the DC tracking feature of AD9361. When comparing case 3 with case 1, the self-interference of FDD mode is suppressed by 30.7dB. The high-quality isolation (i.e., observed 29.4dB at 2.41GHz) between two ports of the power splitter, together with the DC tracking feature in case 3 helps to mitigate the effect of self-interference of FDD mode. Although the power splitter has strongly reduced the self-interference, it also causes the Tx power degradation by a factor of 3.6dB, due to the insertion loss of the splitter. The observed self-interference reduction in case 4 is further improved by 5 dB compared to case 3, indicating that this setup offers better results in terms of performance. Case 4 has the advantage of less Tx power loss, however, it requires accurate antenna placement (orthogonal) to achieve optimal performance, whereas case 3 is easier to implement and likely to offer more repeatable measurements. In conclusion, the setup in case 3 – TDD-WOC and FDD modes of AD9361 along with power splitter and DC tracking enabled feature – has been selected to implement the conventional TDD approach and BBHD-RFFD approach, respectively.

The experiment setup used for TT and receiver sensitivity measurements is illustrated in Figure 36, consisting of two identical SDR nodes. In terms of hardware, each SDR node is composed of a Zynq-7000 SoC (i.e, ZedBoard in our setting), an FMCOMMS2 board which incorporates AD9361 front-end, and a power splitter. A SubMiniature version A (SMA) cable is used to connect the two SDR nodes, ensuring interference free condition for the sensitivity measurement. Each SDR node is configured to act as an IEEE 802.15.4 compliant transceiver.

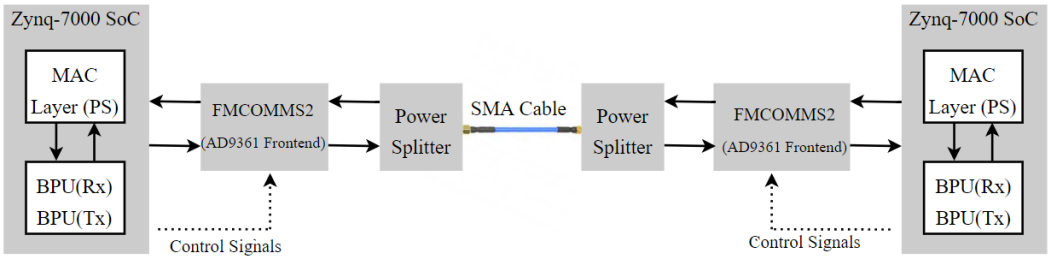


Figure 36: Experimental setting for turnaround time and receiver sensitivity measurements.

The TT measurement is relatively simple; it requires only one SDR node, TT is determined by accessing relevant registers of AD9361. More specifically, the following steps are used to measure the TT:

1) A MAC application running on PS (as shown in Fig. 6) issues commands to AD9361 to trigger Tx/Rx switching.

2) In parallel a global timer on PS is enabled and started.

3) The MAC application continuously read the status register of AD9361 (i.e., polling).

4) Upon obtaining the desired status (i.e., polling ended, and related registers of Rx or Tx have changed their status), the global timer is stopped and TT is calculated by reading the current value of the global timer.

The table below shows the TT consumed by all three modes of AD9361 whenever switching takes place. It can be observed that Rx to Tx switching always takes more time because the turn-on time of DAC is higher than that of ADC in AD9361. The worst TT values are observed in TDD-WC mode whereas nearly zero (i.e., ≈10 ns) TT is observed in FDD mode (i.e., BBHD-RFFD approach). The maximum TT specified by the IEEE 802.15.4 standard is 192 µs [15], therefore this standard can be implemented with any of the three modes of AD9361. The short inter-frame space (SIFS) values for IEEE 802.11 standard are 10 µs (2.4GHz) [19] and 3 µs (60GHz) [20]. Both TDD-WC and TDD WOC of AD9361 would fail to meet SIFS (TT is one of the many parts of SIFS) requirements of Wi-Fi, but our proposed approach makes this feasible. In fact, any standard with stringent TT can be realized with our approach.

Table 9: The average TT measured under three duplex operation modes of AD9361with a standard deviation of around 1µS.

TT (µs)	TDD-WC	TDD-DOC (conventional TDD)	FDD (BBHD-RFFD)
Tx to Rx	40.2	4.2	≈ 0
Rx to Tx	53.0	17.2	≈ 0

As mentioned before, since RF parts of both transmitter and receiver are constantly activated in our approach, the receiving path is suffering from self-interference. In section IV the self-interference has been analyzed to be 8.2 dB and this causes the noise to increase by the same amount. It is therefore expected that the receiver sensitivity of the system will be equally affected. We are using IEEE 802.15.4 compliant BPUs, therefore the sensitivity is measured according to this standard requirement [15]. The sensitivity of a receiver is defined as the input power level when the Packet Error Rate (PER) drops to 1%. PER is measured by counting the proportion of packets lost during transmission against 10,000 transmitted packets, each containing 20 octets with 14 bytes PHY payload. In order to determine the receiver sensitivity, the PER measurement is repeated when packets are sent with different Tx powers, which is controlled by tuning the variable attenuator in the AD9361 Tx chain.

For determining receiver sensitivity, however, PER needs to be mapped to the input power of the receiver rather than Tx attenuation. The translation from the Tx attenuation (dB) to absolute Rx power (dBm) is achieved using the following steps: (1) the SMA cable along with power splitter is detached from the receiver AD9361 RF front-end and attached to an Anritsu MS2690 A spectrum analyzer [21], in another word, the spectrum analyzer replaces the receiver SDR in the setup depicted in Figure 36 (2) the Rx power is measured via spectrum analyzer when different settings of the attenuator are applied to AD9361. The figure below shows that the relation between the Tx attenuation and the measured Rx power by the spectrum analyzer is approximately linear. By making use of this linear relation, Tx attenuation can be easily translated into absolute power level, which is then used to acquire the receiver sensitivity.

© ORCA Consortium 2017-2020

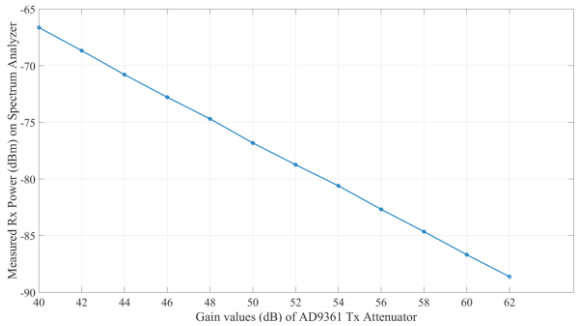


Figure 37: Measured Rx power (in dBm) vs AD9361 Tx attenuator.

The PER obtained under a range of Rx input power are illustrated in the figure below. It can be observed that the receiver sensitivity of BBHD-RFFD and the conventional TDD are -91dBm and -98.5dBm, respectively. The measured receiver sensitivity of the conventional TDD approach is comparable with that offered by commercial chipsets. For instance, cc2538 chipset has the receiver sensitivity of -97dBm [14]). It can be concluded from the measurements that the receiver sensitivity of our approach is degraded by 7.5dB, compared to the conventional TDD. This degradation is obviously caused by the 8.2dB self-interference, however it is not exactly aligned (i.e., 0.7dB difference). This is because DC leakage is a part of the self-interference, which does not affect the receiver sensitivity. In conclusion, despite the degradation, the receiver sensitivity in BBHD-RFFD approach is still well above the minimum requirement by the standard (i.e., 85dBm at 2.4GHz). We thus expect no severe impact on other systems.

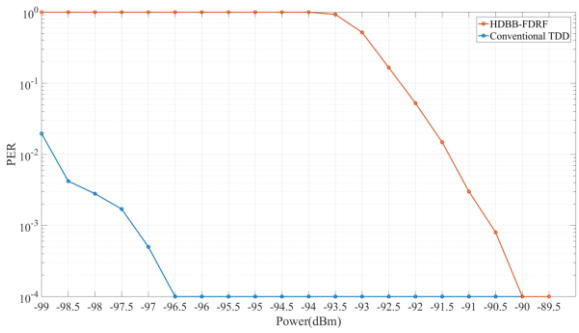


Figure 38: Receiver sensitivity measurements.

5.1.2 Higher layer protocol integration

The zero turnaround time is basically higher layer transparent, the transmit and receive radio interface remains identical. The only difference is that the upper layer protocol designer has less restrictions, before this feature is added, they would need to manually consider the turnaround time to design their protocol, while after this feature is included they won't need to consider the minimum turnaround time.

5.1.3 Testbed integration

The AD9361 based SDR platform with Zynq FPGA boards are deployed in w-iLab.t. Anyone who uses this platform to implement radio stack that requires tight latency performance can benefit. TAISC is an example of such kind of upper layer radio stack framework, which is verified to be running in the w-iLab.t testbed. More details are available in D6.3.

5.2 Relation to showcase

This feature is used in showcase2, though rather than directly applying FDD mode of AD9361, the switching of configuration approach of AD9361 chipset from SPI to pin configuration is sufficient. More details of AD9361 configuration are discussed in D4.3. As these configurations were sufficient to demonstrate a low latency link of showcase2, FDD mode is not applied since it will only bring unnecessary self-interference.

5.3 Risk analysis

There is no specific risk observed at this phase of the work package.

5.4 Implementation plan

5.4.1 PHY improvements

The main physical layer improvement of DSSS accelerator will be about virtualization. In fact, in Y2 we have already started in this area, virtualization of transmitter up to 4. In order to realize virtualization, FPGA footprint needs to be reduced, which comes down to the following tasks:

- Optimize memory usage, there are currently two stages memory usage, one can be removed in the current implementation, which will lead to big benefit for PHY virtualization
- Currently in the virtualized transmitter implementation, we are us using FIR filter as interpolation filter in the up sampling process, this could be improved by using a poly phase filter, which allows using less filter tabs and consequently saving multipliers on the FPGA. The filter memory can be further decreased, since it is symmetrical
- As an alternative of writing own filter or using poly phase filter for interpolation of upsampling, an Xilinx IP core already exists for this purpose, this module will be explored in case it is better for performance or FPGA footprint.

5.4.2 Higher layer protocol integration

The virtualized physical instances will be exposed to the upper layer. The TAISC MAC engine will be extended in order to cope with multiple physical layer instances. In TAISC it is possible to install multiple MAC protocols, for which each of the MAC protocols will be coupled to a virtualised physical layer instance. Until now, TAISC only supported a single MAC protocol at a time, with no concurrent operation possible. Supporting parallel execution of multiple protocols could give issues on the core time-annotated principles, since TAISC promises predictable and in-time execution of instructions. With concurrent operation comes the risk of interrupting the execution of a MAC protocol due to parallel execution of other MAC protocol, having an unpredictable impact on instruction durations and by consequence the predictable performance. To make sure that TAISC instructions are not interrupted by other protocols or processes, the time domain will be divided into consecutive time slots. These slots will be allocated to individual MAC protocols, guaranteeing uninterrupted execution within the time slot. The TAISC compiler should make sure that the duration of a time slot is long enough to guarantee that the execution of the longest instruction can fit within the slot.

Following developments should be made:

- The calculation of the time slot duration;

- The TAISC core will divide the time domain into slots, and allocate the slots to individual MAC protocol instances;
- Functions that pass control to the operating system (e.g. transmission success, packet received, report, etc.) should be rethought, since these also need to finish within the duration of the time slot.

5.4.3 Testbed integration

The virtualization of physical layer of DSSS and extended TAISC will be integrated and tested in the w-iLab.t testbed. The new features will be made available in the Github repositories for testbed users to access. The ORCA website for software components and the relevant tutorials in the w-iLab.t website will be updated to incorporate new features.

© ORCA Consortium 2017-2020

6 RADIO VIRTUALISATION ON DEMAND

This section describes TCD´s eXtensible Virtualisation Layer (XVL), a framework for radio virtualisation on demand implemented in Year 2, and the main results obtained. In this deliverable, we describe the general PHY functionality, its integration with upper layers, and future steps. For further details on the control plane and slicing aspects of TCD's contribution to Year 2, please refer to Section 6 of D4.3.

6.1 Implementation results

Throughout this section, we describe XVL's capabilities with regard to its use for creating radio slices on demand. This is in line with its intended use in the Year 2 Showcase 3, where TCD is involved. Additionally, we expect that a considerable percentage of the functionality provided by this framework can also be repurposed for investigating open research challenges in radio virtualisation, such as isolation, signalling and mobility management.

XVL is a cross-platform software layer that sits on top of existing radio hypervisors. It works as a wrapper, leveraging the underlying radio hypervisor's capabilities and providing it with the missing functionality for supporting RAN as a Server (RANaaS). This modular design allows the radio hypervisor to focus on the radio virtualisation, offloading to XVL the majority of other tasks, e.g., the communication interface, resource management, and monitoring. Namely, XVL provides a communication interface for resource negotiation and allocation, supports the creation and destruction of virtual radios on demand, ensures their computational isolation, and allows the monitoring of their performance.

6.1.1 PHY improvements

Every virtual radio has a virtual RF front-end, which the virtual radio employs for transmission and/or reception. Each virtual RF front-end uses a fraction of the total radio resources of the real RF front-end. The radio resources can be defined in terms of bandwidth, timeslot, or antennas, depending on the radio hypervisor's approach to virtualising the real RF front-end. Regardless of the type of radio resource, each virtual radio must send an appropriate number of IQ samples to the radio hypervisor at precise timing. The radio hypervisor consumes the IQ samples of all virtual radios at once and multiplexes the virtual RF front-ends into the real RF front-end. The number of necessary IQ samples per virtual radio, however, varies depending on the amount of radio resources per virtual RF front-end. In the case of an FFT-based hypervisor, e.g., HyDRA or MySVL, the bandwidth of each virtual RF front-end is mapped onto a number of FFT bins. The virtual radios must generate a number of samples equal to the number of FFT bins to create an FFT window. Then, the radio hypervisor multiplexes the windows of the virtual RF front-ends together using an IFFT.

© ORCA Consortium 2017-2020

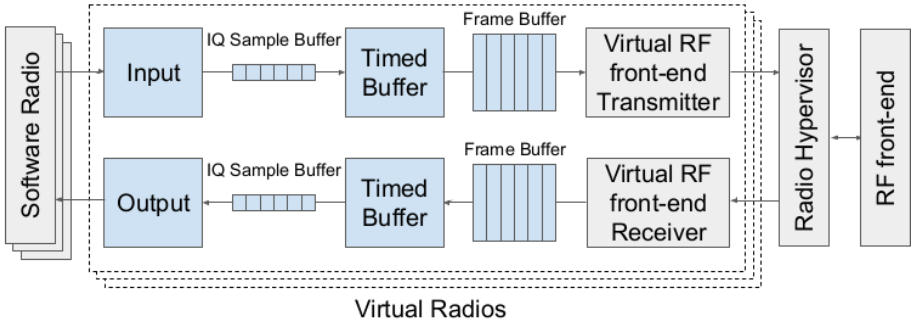


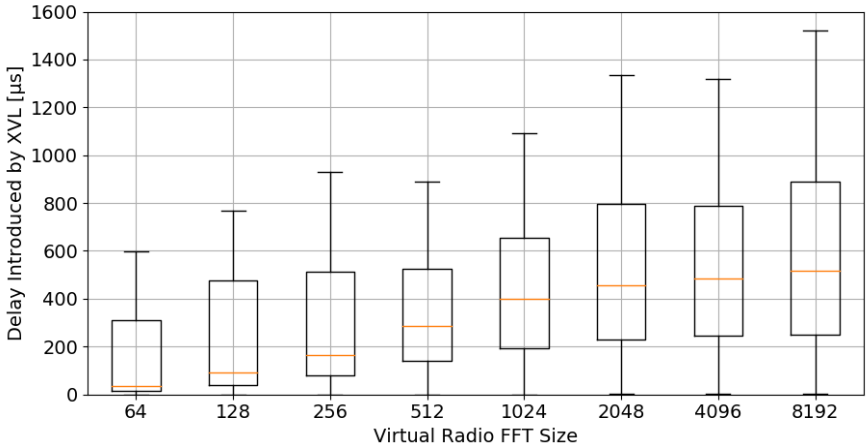
Figure 39: The timed buffer consumes the received IQ samples and generates the windows that the radio hypervisor requires. The same process happens in the opposite direction.

XVL employs timed-buffers to ensure that the radio hypervisor receives the appropriate number of IQ samples at the right moment. Based on the overall sampling rate of the virtual radio and the hypervisor, the timed-buffers output IQ samples at the precise moment for the radio hypervisor's consumption. Figure 39 illustrates the operation of timed-buffers inside a virtual radio, where UDP sockets are used for receiving/sending IQ samples from/to remote software radios. In the case of overflows, XVL's internal timed-buffer will start filling, until reaching a cap. After that, incoming samples are dropped to prevent the XVL process from overflowing memory. In the case of underflows, XVL can either pad the frames with missing IQ samples with zeroes or transmit empty frames while waiting for the missing samples. This way, the radio hypervisor and the rest of the system will not halt waiting for the delayed or missing IQ samples of a given virtual radio.

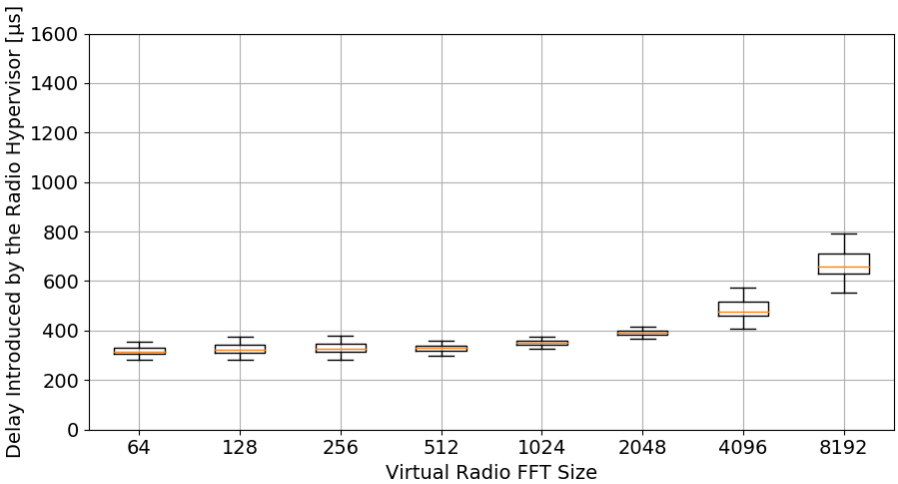
The virtual radios can either operate as transmitters, receivers, or transceivers. However, the virtual radios may not be symmetrical in both directions, i.e., they may employ different RATs and require different amounts of radio resources. For this reason, every virtual radio in XVL possesses completely independent transmitter and receiver chains. Figure 40 shows an example of a virtual radio with both types of chains: the transmit chain (top) receiving samples from a given software radio, and the receive chain (bottom), sending samples towards a given software radio. The radio hypervisor may realise each type of chain in different manners, i.e., different virtualisation approaches, or may use different radio hardware, i.e., different physical devices for transmission and reception.

6.1.1.1 Service Delay

In this analysis, we are interested in the delay that XVL and HyDRA introduce to serve virtual radios and forward their IQ samples. The service delay is crucial to analyse the impact caused by using virtualised infrastructure instead of a real one, and to evaluate whether it can impair the operation of the virtual radios. First, we measured the time spent by XVL for delivering the IQ samples to the radio hypervisor, i.e., from successful UDP receptions, through the timed buffers, and generation of a frame. Figure 40a shows the results of our measurements, with a median of 343µs. Then, we measured the time spent by HyDRA to virtualise the RF front-end, i.e., consume the frames, multiplex the virtual RF front-ends using an IFFT, and forward the IQ samples to the real RF front- end. Figure 40b shows the results of our measurements, with a median of 334µs. These results show that XVL introduces a service delay within the same order of magnitude of the radio hypervisor. However, the overall median introduced delay by using XVL and HyDRA for supporting RANaaS falls under 1 ms. A 1 ms delay can be impactful in the delay budget for Centralised-RAN (C-RAN) scenarios, and must be taken into account in the network planning.



(a) The service delay introduced by XVL. The service delay increases with the number of resources, due to the longer time required to allocate larger frames in memory.



(b) The service delay introduced by HyDRA. The service delay increases with the number of resources, due to the increasing complexity of the FFT/IFFT operations.

Figure 40: Delay measurements for different virtual radio configurations. All the measurements were made using a real bandwidth of 2 MHz and an overall FFT size of 8192 bins, which corresponds to the worst case scenario in Section 6.1.2.1.

6.1.2 Higher layer protocol integration

The integration with higher layers is accomplished through a negotiation of radio resources. The XVL operates based on a client-server paradigm, with a single server, i.e., one instance of XVL on top of a radio hypervisor, serving multiple clients, i.e., verticals and MVNOs. The clients can send messages to the server for querying information about the use of resources and requesting resources at any time. Figure 41 depicts the process of negotiating and using resources through XVL. Upon successful negotiation and reservation of resources, XVL interfaces with the radio hypervisor for creating/destroying the RAN slice. XVL has callbacks for informing the radio hypervisor about changes in the resource mapping and/or allocation of virtual radios, so the radio hypervisor can instantiate a new virtual RF front-end. Once the radio hypervisor completes the creation of the virtual RF front-end, XVL

can perform additional configurations in the virtual radio. Finally, XVL informs the client of the proper ways to reach its virtual radio, e.g., through CPRI, OBSAI, or a UDP socket, from which the clients can start transmitting or receiving IQ samples.

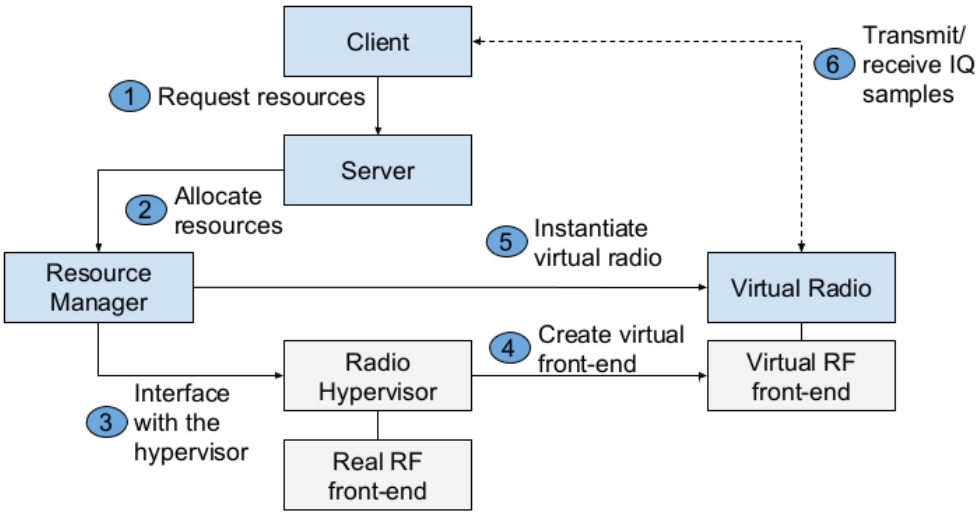


Figure 41: An example of the steps involved in negotiating the use of resources and allocating a new virtual radio. XVL (blue) stands as a wrapper around the radio hypervisor (grey).

At present, XVL supports virtualisation in the frequency domain. However, XVL separates the resource management from the underlying technicalities of the radio hypervisor, which allows the resource manager to be extended and modified at ease. Thus, XVL can easily be extended for supporting time or space instead of frequency resources, or even incorporate time and space as new degrees of freedom in the resource definition. It only depends on the types of radio virtualisation that the radio hypervisor supports. Currently, the radio resources are defined in terms of a centre frequency, a bandwidth, a client ID, and a UDP port.

6.1.2.1 Provisioning Delay

In this analysis, we are interested in the delay for provisioning virtual radios with XVL. The provisioning delay is crucial for planning and evaluating a RAN deployment using the RANaaS paradigm, as it dictates the time interval required to provision the virtual radios that realise the RAN. This interval comprehends the time between the client sending a resource request, and XVL returning the resource allocation confirmation. Figure 42 shows the results of our measurements for different configurations of the virtual radio and radio hypervisor. The exponential growth in the semi-log scale denotes a linear behaviour in relation to the number of resources allocated per virtual radio, regardless of the FFT size of the radio hypervisor. We attribute the linear behaviour to the time for allocating memory for XVL's buffers. The worst case scenario where a virtual radio required 8192 FFT bins took less than 300ms to be fulfilled. Hence, this result indicates that XVL is fast enough to provision virtual radios on-the-fly.

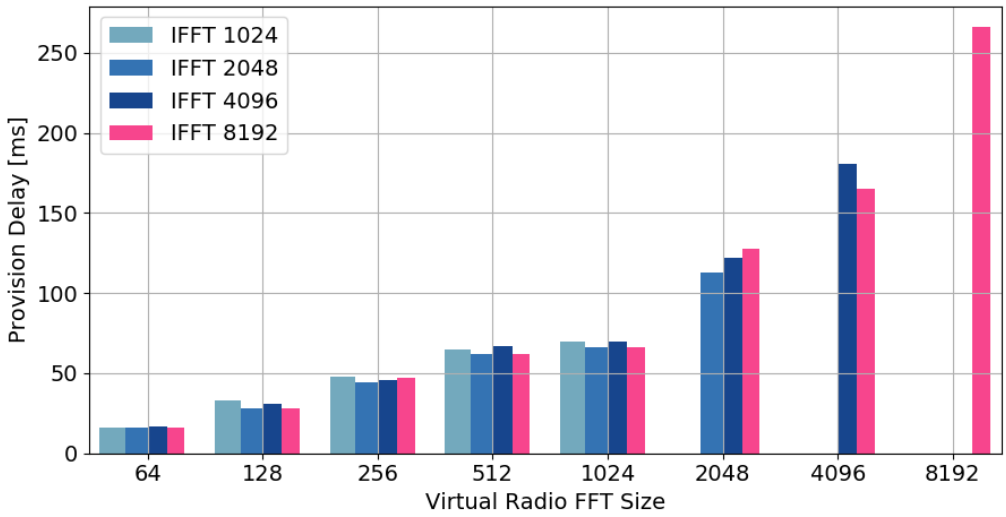


Figure 42: The provisioning delay to fulfil a virtual radio request. The time taken for creating virtual radios is proportional to the number of allocated FFT bins, but not to the overall IFFT size of the radio hypervisor.

6.1.3 Testbed integration

XVL was developed and validated in the IRIS testbed, using USRP N210s. XVL's source code will be made available through a git repository.

6.2 Relation to Showcase

The Showcase 3 will be divided into three stages: the demonstration of TCD's FFT-based radio slicing capabilities, the demonstrations of IMEC's OFDM-based radio slicing capabilities, and the combination of the two radio slicing approaches. The XVL will empower TCD's radio slicing (MySVL) to allow the creation of radio slices on the fly.

6.3 Risk analysis

There is not any identified risk at this stage, with regard to the PHY and higher layer integration of the proposed solution.

6.4 Implementation plan

This section describes TCD's plans for implementing Radio Function Virtualisation (RFV) on SDRs for Year 3. We plan to extend XVL to instantiate radio functions on top of its virtual RF front-ends. Ultimately, allowing the creation of virtual radios with both a virtual RF front-end and a virtual radio stack, as shown in Figure 43.

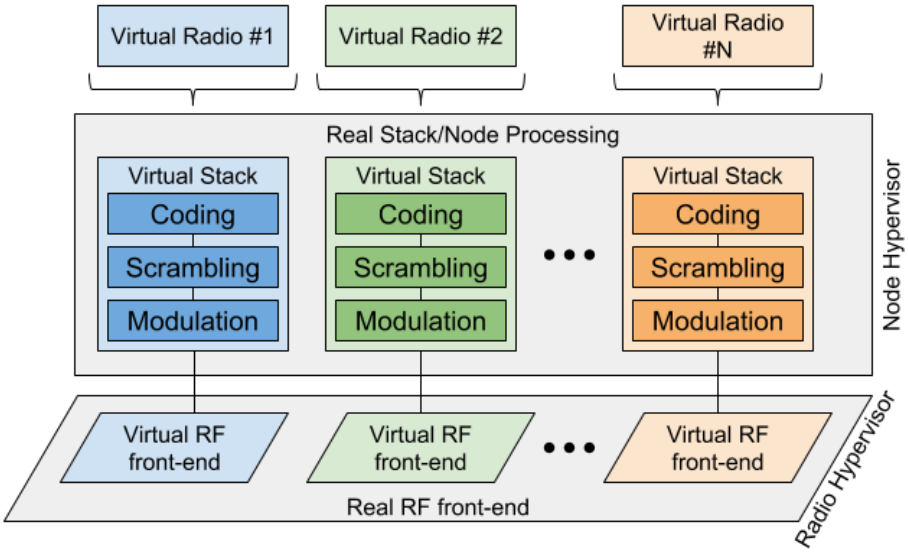


Figure 43: Illustration of virtual radios with virtual RF front-ends and virtual radio function stacks.

6.4.1 PHY improvements

The chain of radio functions that constitute a Radio Access Technology (RAT), e.g., coding, scrambling, and modulation, can be optimised when running on specialised hardware, e.g., FPGAs, DSPs. We plan to investigate the latency improvements that we can achieve by leveraging heterogeneous processing to run particular virtual radio functions of a radio function chain.

6.4.2 Higher layer protocol integration

The provisioning of virtual radio functions on top of a virtual RF front-end will be realised through an extension of the existing virtual radio provisioning procedure. The clients, e.g., verticals, MVNOs, will be able to request not only virtual radio resources, but also request some radio functionality to operate on top of their respective virtual RF front-ends.

6.4.3 Testbed integration

We plan to make available to future experimenters the following functionality:

1. On-demand instantiation of virtual radio radios with only a virtual RF front-end, operating similar to Remote Radio Heads (RRHs).
2. On-demand instantiation of virtual radios with a virtual RF front-end and a virtual radio stack.

7 NS-3 BASED PROTOTYPING PLATFORM FOR RAT INTERWORKING

The goal in Year 2 was to further enhance the Multi-RAT platform for real-time use towards the envisaged platform overview shown in Figure 44.

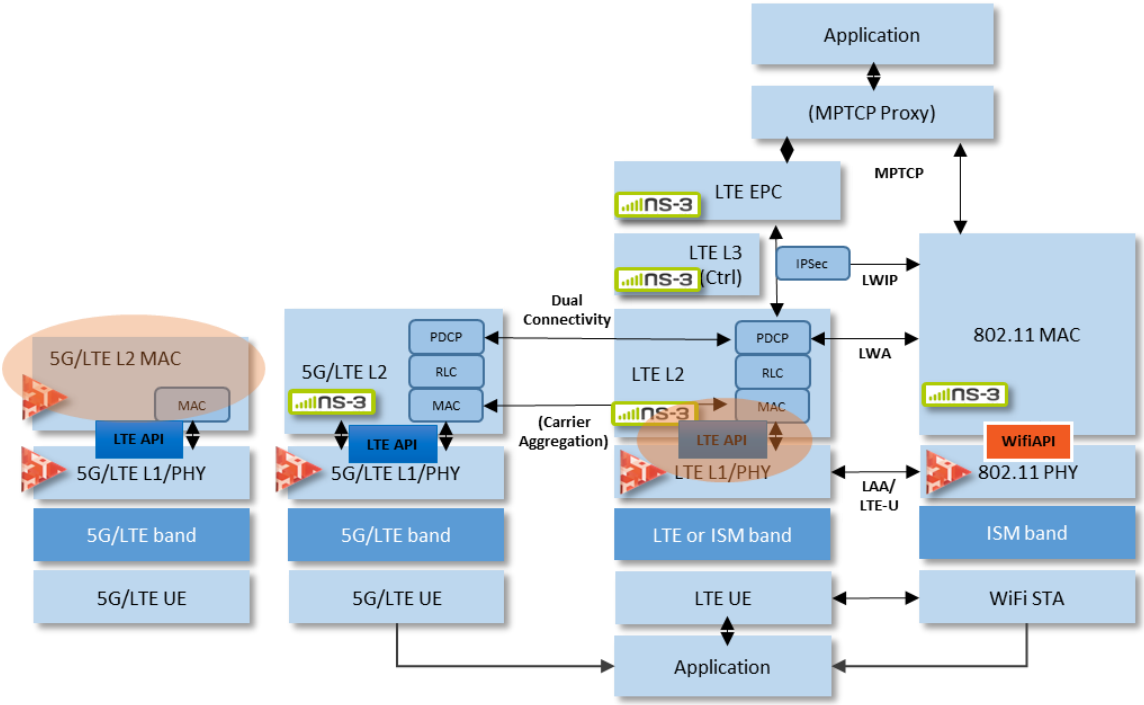


Figure 44: Envisaged Platform for Multi-RAT experimentation

Specific focus in Year 2 was drawn on two parts of this platform.

- The adaption of the NI LTE Application Framework and corresponding API and ns3 integration to become real-time capable in order to meet LTE timing requirements and dynamic configuration in DL based on TTI (1ms) timing. This interfacing and the application framework itself were no real-time implementations beforehand.
- As the ns3 platform doesn't offer a 5G MAC implementation, a LabVIEW based MAC implementation supporting 5G-NR MAC protocol format was added as an option for future inclusions of real 5G PHY implementations into the platform

The technical details and insights of these developments are described in the subsequent sections.

7.1 Implementation results

7.1.1 PHY improvements

The NI LTE Application Framework PHY layer implementation was completely reworked to become real-time capable when running on a real-time CPU. This effort was done in conjunction with the implementation of the generic L1/L2 API. These efforts are described in detail in Section 7.1.2.1.

7.1.2 Higher layer protocol integration

One main focus of NI in Year-2 was on implementing the generalized L1-L2 API in the LTE link of the NI Multi-RAT platform described in deliverable D3.1. The goal of this generalized API (see ORCA

functionality [22]) is to support different implementations of the LTE Layer 2 (and above) with the use of the NI LTE Application Framework in order to enable an end-to-end wireless system used for the ORCA experimentation platform.

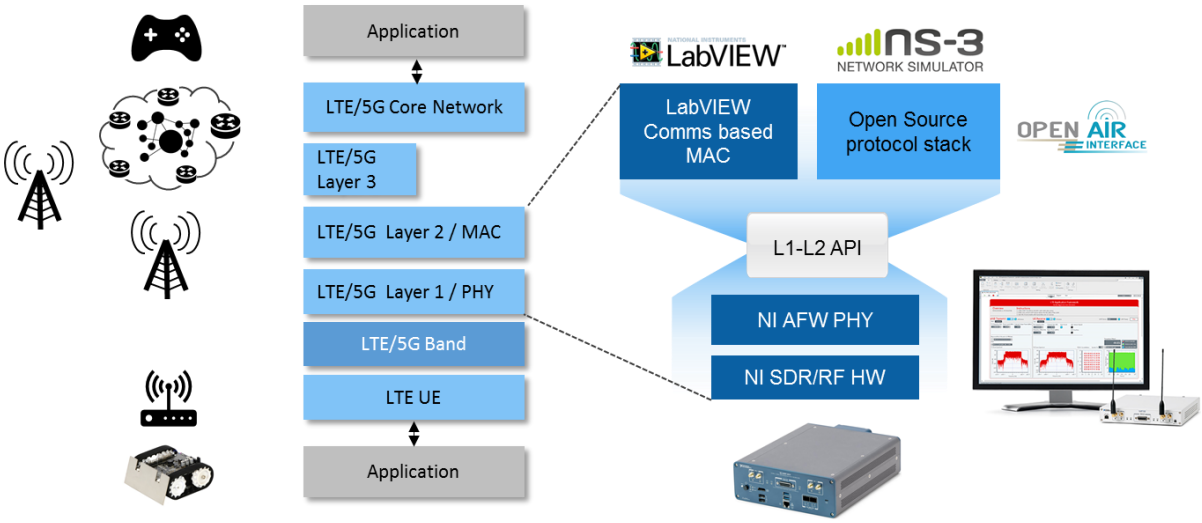


Figure 45: Overview about the generalized NI L1-L2 API.

As depicted in Figure 45 with the used L1-L2 API on the one hand side one can use a LabVIEW based MAC implementation or an existing MAC implementation available in an open source stack like ns-3 or Open Air Interface. As described in D3.1 a similar API is also used to connect the ns-3 WiFi module with the NI 802.11 Application Framework. In the future, it is planned to also integrate a 5G system using a similar API that is derived from the LTE link. In the following we describe how the L1-L2 API is implemented in the NI LTE Application Framework as well as in the ns-3 LTE module.

7.1.2.1 Integration of L1/L2 API into NI LTE Application Framework

An overview about the NI LTE Application Framework is given in [23]. In the default setting this system can be used for doing research on PHY algorithms having an LTE OFDM system as reference supporting an FPGA based real-time processing and bandwidths up to 20 MHz allowing throughput values of up to 75 MBit/s. In order to make use of this powerful PHY implementation for MAC layer researchers in Year-2 NI implemented the L1-L2 API in its LTE Application Framework and made it available as default feature. Further the CPU software architecture was completely redesigned to be real-time capable in order to meet LTE timing requirements at MAC layer. The API implementation specification has already been recently published in [24] by NI and will be summarized in the following.

The API implementation adds a PHY service access point (SAP) on top of the LTE Application Framework PHY. The PHY SAP provides a message-based interface between the LTE L2 and L1 stack. It includes the eNodeB and UE with DL and UL transmission.

The PHY SAP follows a general API concept and is based on a common set of 3 general message types

- REQ: Service request from a higher layer to lower layer
- CNF: Confirmation of the received service request from lower layer to higher layer
- IND: indication (status, data, error, and so on) from lower layer to higher layer

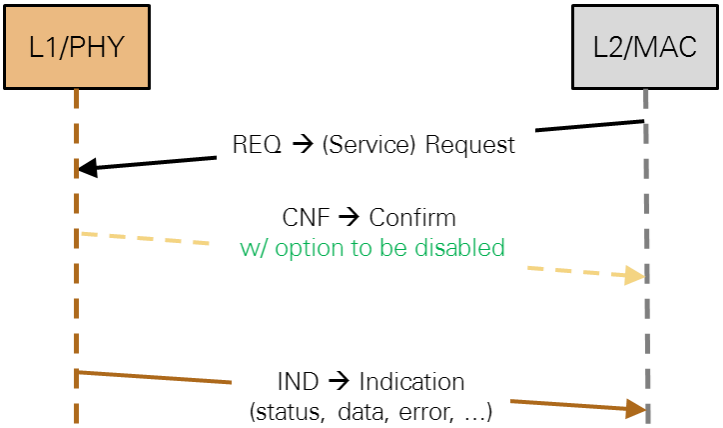


Figure 46: Simplified communication between L1 and L2 based on API message types.

The REQ messages are separated into config and payload message with the former including PHY control parameter (DCI/UCI) like MCS, RB allocation, transport block size and the latter including transport block bits.

The general message structure for all messages is shown in Figure 47.

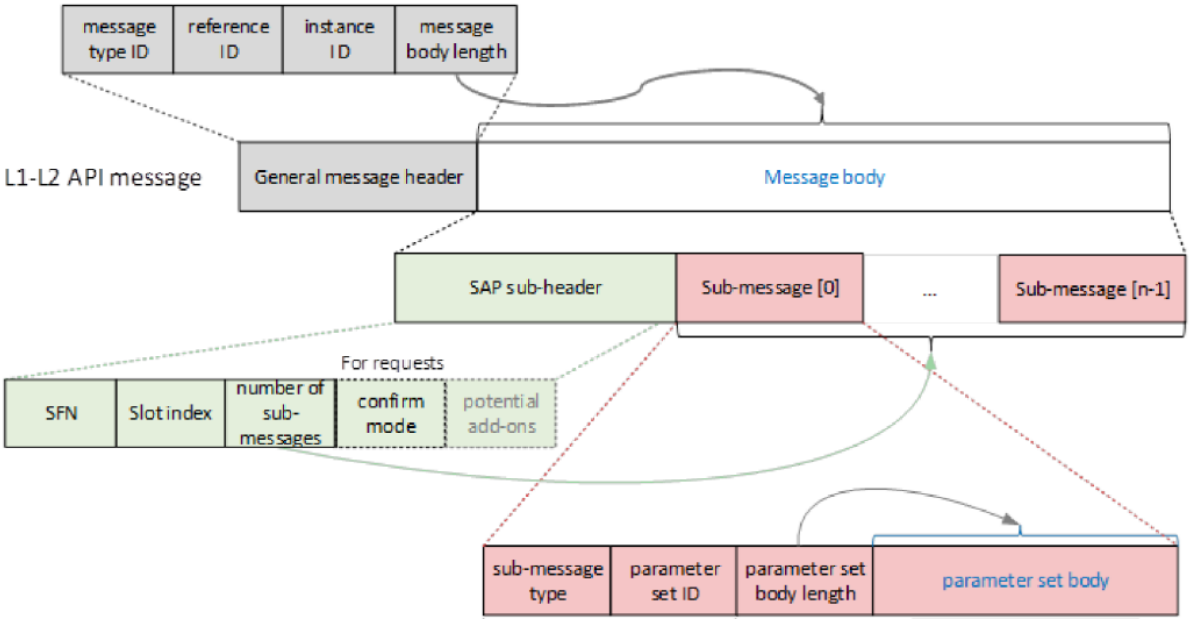


Figure 47: PHY SAP General Message Structure.

It is worth noting that this message structure is derived from the well-known Small Cell Forum FAPI specification [25] with some extensions and modifications. Furthermore, as described in [26] the same message format is also used for the API implementation in the 802.11 Application Framework implementation that is used to connect the ns-3 WiFi module.

As for the concrete LTE PHY SAP the following set of messages are used to access and run the LTE Application Framework PHY:

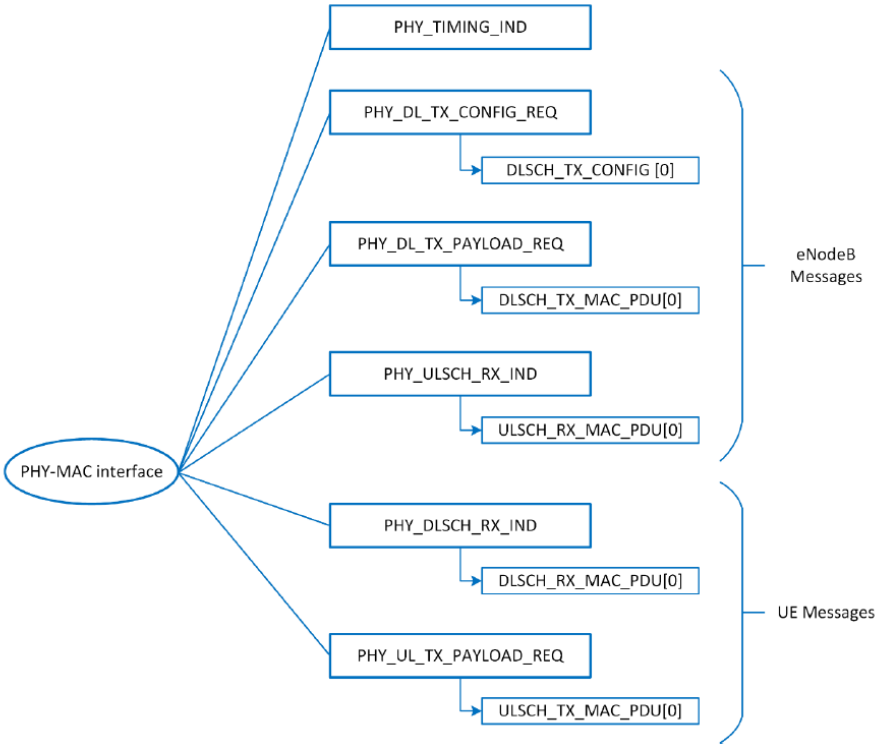


Figure 48: LTE PHY SAP Message Definitions.

For the LTE downlink specifically, the DL TX Config Request message is used to configure the data channel (PDSCH) as well the control channel (PDCCH) while the DL Tx Payload request message includes the MAC PDU that will be transmitted through the PDSCH. At the receiver the PDSCH data content will be passed to the MAC via the DLSCH Rx Indication message. As the uplink can only be semi-statically configured there is only a UL Tx Payload request and ULSCH Rx Indication message implemented. A very important message represents the PHY Timing Indication message that is used to synchronize the MAC implementation regarding the LTE frame timing of 1 ms as well as the used PHY TTI counters. This is crucial to let the MAC schedule data for specific PHY data frames and to achieve the required LTE DL/UL round trip latencies. A timing diagram of the message sequence is shown in Figure 49.

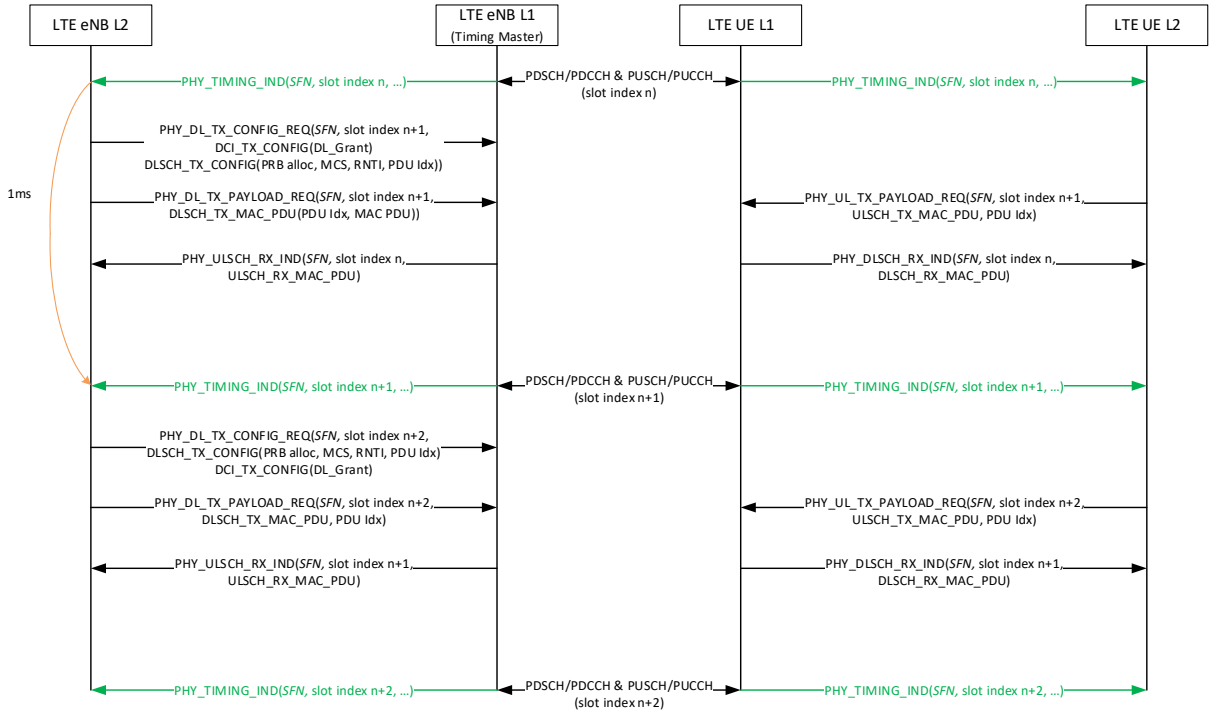


Figure 49: LTE PHY SAP Message Sequence Chart.

The content of these messages (i.e. the specific parameters) is also provided in [24] but will not be discussed in more detail here.

One important API requirement is to ensure a reliable communication channel between the LTE MAC and the LTE PHY to achieve the required 1 ms LTE TTI timing with a low jitter. As the underlying real-time CPU system is based on Linux, in this implementation Linux pipes are chosen that provide the required flexibility to have a generalized API. Linux pipes are used to connect to either a MAC written in LabVIEW or a MAC written in C/C++ programming language (refer to Figure 45). It should be noted that for the WiFi API implementation UDP has been used to connect the ns-3 WiFi High MAC module with the 802.11 Application Framework middle MAC. But this is not possible in the LTE case due to the stricter timing requirements of PHY and MAC interactions. For the implementation LabVIEW and C functions are implemented to provide access to this API transport channel. Figure 50 shows a block diagram of the API implementation.

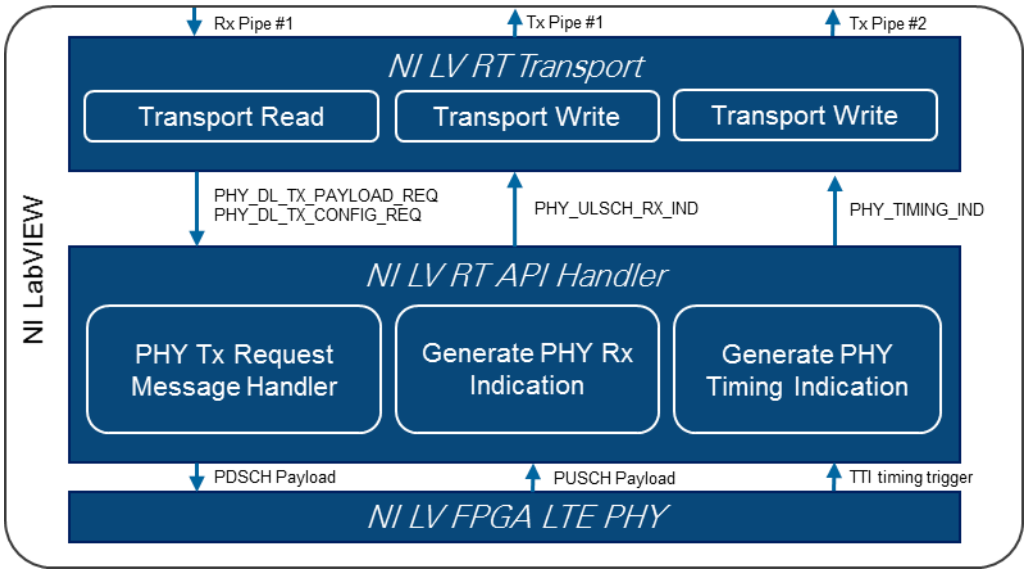


Figure 50: API Implementation Diagram (eNB example).

There is an API handler responsible for handling the support of API messages and the corresponding message sequences. It should be noted that the API message handler can be configured to act as eNB or UE which allows flexible use of the same system components.

In addition to the implementation of the L1-L2 API in the LTE Application Framework NI also added the support of the new USRP-2974 hardware platform. The new USRP-2974 is a stand-alone device including an Intel Core i7 CPU and a Xilinx Kintex-7 FPGA as well as the ADC/DAC + RF. The basic hardware system will cover a frequency range of 10 MHz-6 GHz with a maximum instantaneous bandwidth per channel of 160 MHz. Therefore, it perfectly fits to the testbed needs utilizing multiple nodes in a distributed manner connected and remotely controlled via Ethernet. Furthermore, it supports also the extension with an additional legacy USRP-RIO connected via a PCIe expansion port. This shall in principle allow to run an NI LTE and 802.11 Application Framework on the two available FPGAs. The proof of concept will be part of the activities in Y3. As the new USRP device includes an Intel CPU where an NI Linux RT system is operating one can also run the ns-3 code on the same embedded device as the Application Frameworks. This results in a significantly reduced form factor and cost for such a complex prototyping system.

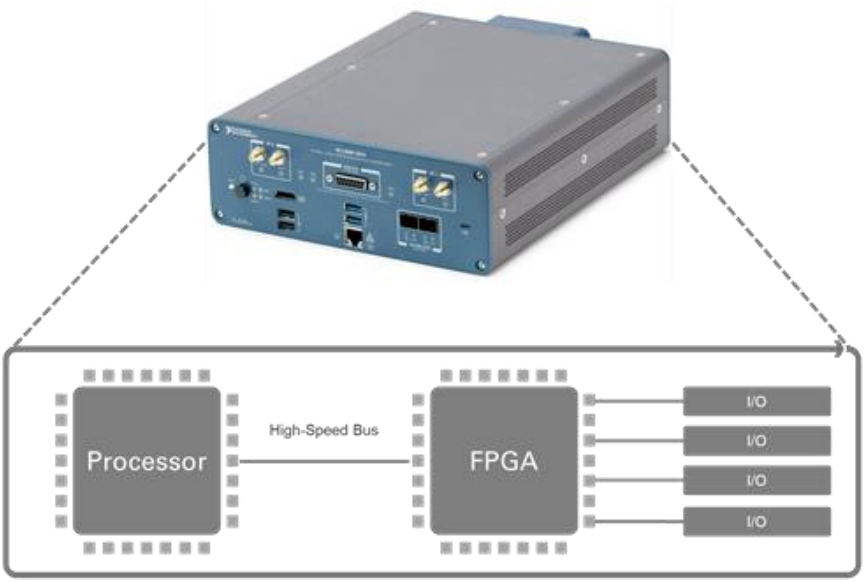


Figure 51: USRP-2974 Hardware Device.

7.1.2.2 NI LTE L1/L2 API Interface for ns3

As described earlier, the goal of the API implementation is to connect the NI LTE Application Framework PHY also with external open source stacks providing the MAC and higher layer functionality. In this work, the ns-3 simulator is used as it includes a rich set of modules that provide functionalities for testing wired and wireless communication system, e.g. LTE and WiFi but also CSMA/ETH and Point to Point links. 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. In Year-2 one focus of NI was on redesigning an existing API implementation of the ns-3 LTE module with the goals:

- Connect to the new L1-L2 API part of the NI LTE Application Framework
- An increased stability by synchronizing the NI LTE PHY implementation with the ns-3 simulator utilizing the PHY timing indication API message allowing real-time TTI base configuration in DL
- Clean separation between the original ns-3 code and the NI API code to allow other users to use and modify as well as to ultimately publish the code
- Allow the ns-3 stack to connect also to other PHY modules using the same API in the future e.g. the TU Dresden 5G Transceiver in Year-3

A detailed description of the ns-3 LTE module is provided in [27]. The ns-3 simulator represents a discrete event simulator completely written in C++ following a rigorous object system to model the individual network components. The ns-3 LTE model provides all components required to configure a complete Core Network as well as eNBs and UEs.

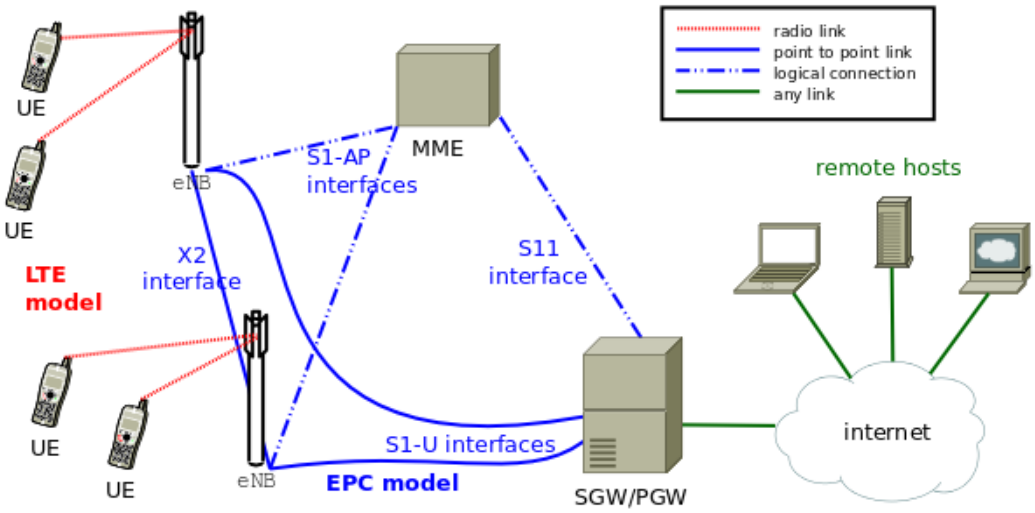


Figure 52: Overview of the LTE EPC simulation model [27].

The eNB and UE model features an implementation of the L2/L3 including RRC, PDCP, RLC as well as MAC including a Femto Forum Scheduler API based interface that allows the use of a variety of different MAC scheduler algorithm implementations. Because ns-3 is mainly intended to be a simulator the PHY is implemented as model that maps SINR values coming from the channel model to a certain bit error rate which also known as MIESM.

The NI L1-L2 API has been implemented to bypass this PHY model and use a real-time PHY instead. This is possible because also the ns-3 simulator includes a real-time mode where the simulator timing is synchronized to the CPU timing. The main changes to the original ns-3 LTE code can be summarized as:

- Synchronizing the ns-3 simulator time to the FPGA timing
- Disable PHY emulation and connect to real-time PHY
- Separate eNB and UE instances to run on the different hardware devices

As described above, the intended system architecture combines a powerful LTE PHY real-time implementation running on an FPGA with an ns-3 LTE upper layer system running on a CPU. In order to achieve a stable operation with these two stack components a synchronization between the CPU and FPGA clock is required. In this system the API PHY timing indication message provides an accurate 1 ms timing of the FPGA clock that can be evaluated in the ns-3 simulator to synchronize to the CPU clock. A basic block diagram that shows the synchronization procedure is shown in the right-hand side of Figure 53.

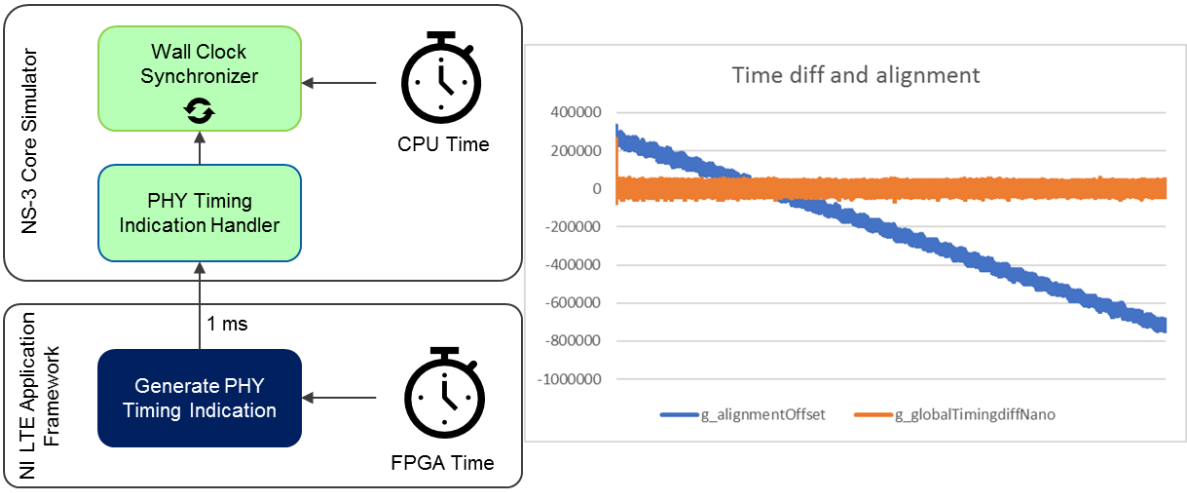


Figure 53: Synchronization between CPU and FPGA clock.

A new PHY Timing Indication message handler has been implemented in ns-3 that evaluates the CPU timing whenever a new message arrives. If the real-time mode is enabled in the ns-3 core simulator the simulator timing is usually synchronized to the wall clock (CPU) using a specific function *GetNormalizedRealtime*:

```
uint64_t
WallClockSynchronizer::GetNormalizedRealtime (void)
{
  NS_LOG_FUNCTION (this);
  return NiCalcNormalizedRealtime (GetRealtime (), m_realtimeOriginNano);
  //return GetRealtime () - m_realtimeOriginNano;
}
```

In the original code the function *GetRealtime* derives the current CPU time so that the simulator time can be calculated accordingly. The newly implemented function *NiCalcNormalizedRealtime* provides a synchronized time using the CPU time and FPGA time received through the PHY timing indication messages as reference. As a result, a very good synchronization between the CPU and FPGA clock can be achieved as shown in the left-hand side of Figure 53. The blue curve represents the offset between the CPU and FPGA clock in ns. As it can be observed, the difference over 60 minutes simulation time becomes almost 1 second and a reliable operation between the ns-3 LTE MAC and NI LTE PHY would not be possible anymore. The orange curve represents the synchronized timing which shows that a high accuracy with only a small offset / jitter between the CPU and FPGA clock can be achieved.

Figure 54 shows a system sketch of the API implementation architecture.

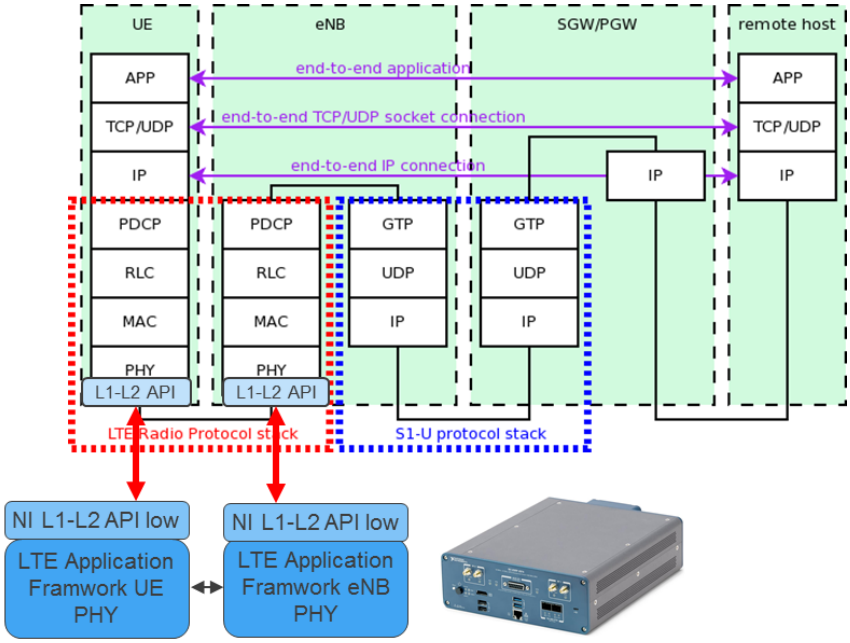


Figure 54: NI L1-L2 API implementation in the ns-3 LTE module (based on [27]).

A detailed implementation diagram of the L1-L2 API as part of the ns-3 eNB stack is shown in Figure 55. For the UE the implementation is done in the same way. As one can see in the figure, on the *LteEndPhy* component level we implemented a switch to chose between the legacy ns-3 simulation mode and the use of the NI L1-L2 API.

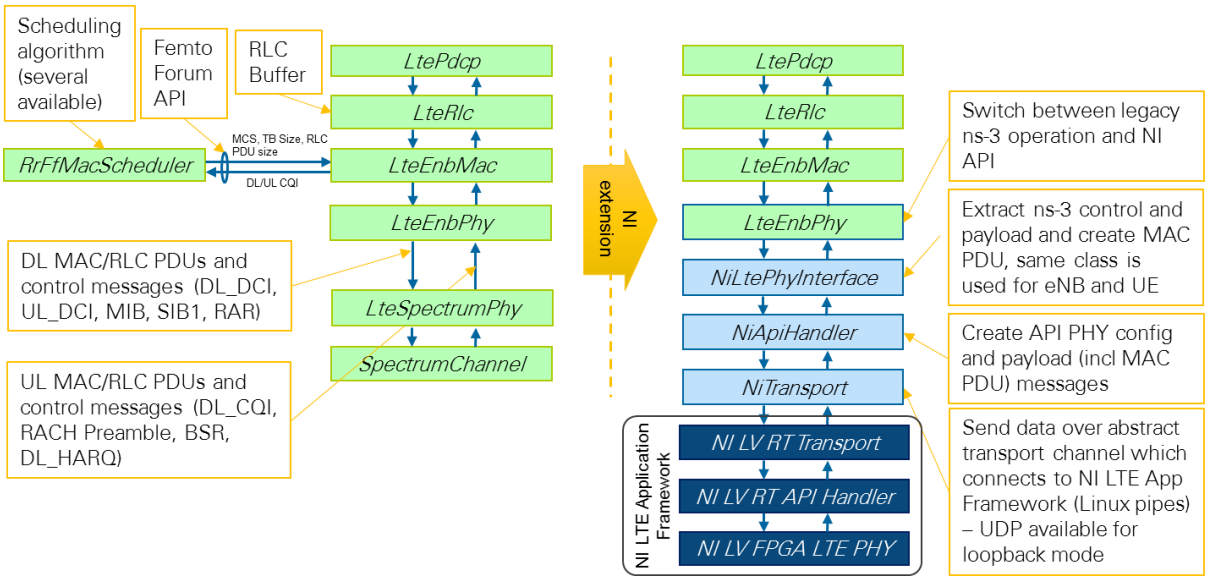


Figure 55: ns-3 LTE eNB Implementation Diagram of NI API.

If the NI L1-L2 API is enabled a newly created *NiLtePhyInterface* class is called that represents the convergence layer between the ns-3 control and payload messages that need to be translated into the previously introduced L1-L2 API messages. This is done by the *NiApiHandler* class that creates the DL Tx Config and Payload Request messages and send them through the NI transport channel. For the API message transport channel on the one hand side Linux pipes are used but on the other hand also a UDP channel can be used to connect the ns-3 eNB instance with a second ns-3 UE instance in a loopback mode. This is in particular useful for debugging without using the SDR hardware.

A detailed block diagram of the new object hierarchy in the ns-3 LTE eNB and UE models including the corresponding Tx and Rx functions is depicted in Figure 56 for the UDP loopback case.

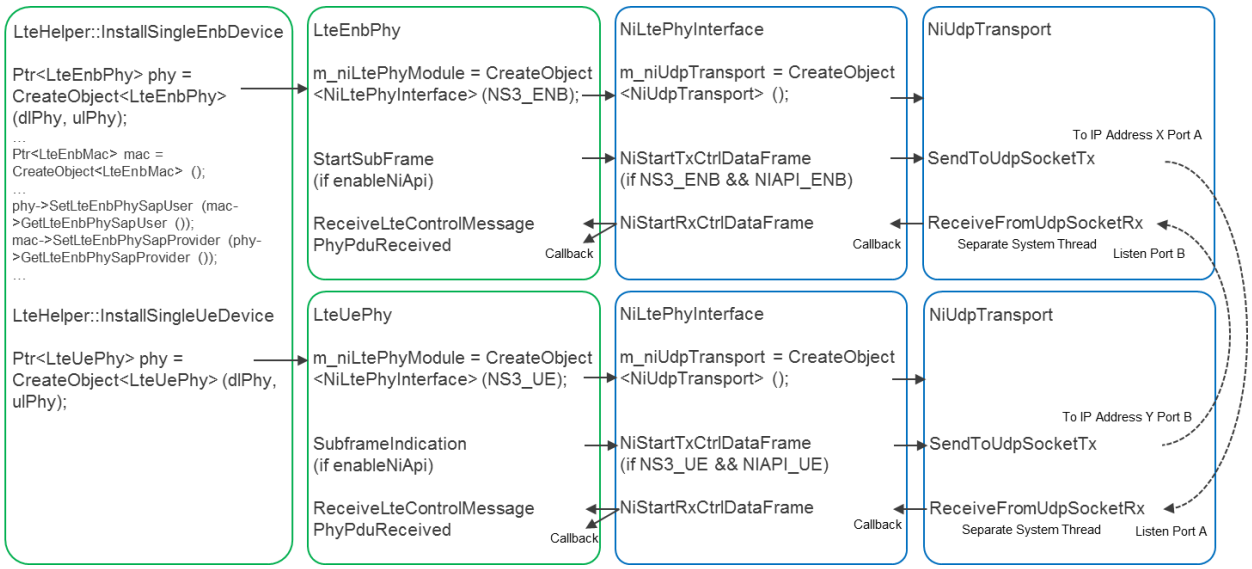


Figure 56: New ns-3 LTE eNB and UE Object hierarchy.

The new *NiLtePhyInterface* class can be configured as eNB and UE which allows to flexibly reuse functions and procedures between downlink and uplink. It should be noted that in the ns-3 LTE module some simplifications are used because of the PHY emulation. For example, for the random access procedure the RACH preamble as well as the MIB and SIB is realized and transmitted as a control message. Therefore, within the *LteEndPhy* components (and similarly for the *LteEndUe*) the *StartSubFrame* function provides a list of control messages and the RLC/MAC PDU packets to the *NiLtePhyInterface* Tx function that extracts the required DL configuration information but includes all control messages in the MAC PDU. At the receiver the *NiLtePhyInterface* Rx function extracts the control messages from the RLC/MAC PDU packets again and provides that data to the *ReceiveLteControlMessage* and *PhyPduReceive* functions so that they don't have to be changed.

The main attributes that can be configured in the new *NiLtePhyInterface* class are:

- *niApiDevType* – to configure the object as eNB or UE
- *enableNiApi* – to enable the NI API
- *enableNiApiLoopback* - to use the UDP loopback mode for debugging

It should be noted that the Wifi system that was described in D3.1 also have been changed to follow the same object hierarchy as in the LTE system in order to allow the users to quickly access and use the NI API for both setups. Figure 57 shows a detailed block diagram about the new object hierarchy in the ns-3 WiFi Access Point and Station MAC models.

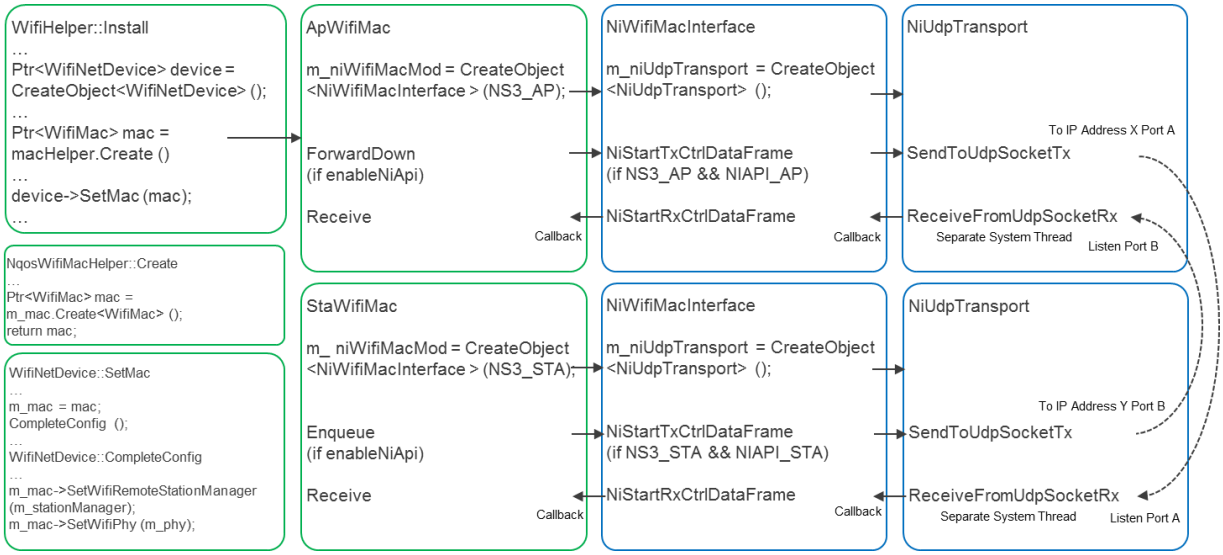


Figure 57: New ns-3 WiFi AP and STA Object hierarchy.

As a key result it is now possible to connect the ns-3 WiFi and LTE modules to the real-time capable NI 802.11 and LTE Application Frameworks utilizing the same generalized API framework. Figure 58 shows a possible configuration showcasing a Multi-RAT base station supporting an LTE as well as a WiFi RAT. This allows the first time that researchers can investigate and experiment on LTE-WiFi interworking techniques using the same SDR prototyping platform. For Year-3 it is planned to extend this Multi-RAT system with a 5G link utilizing the same generalized API concept as done for LTE and WiFi.

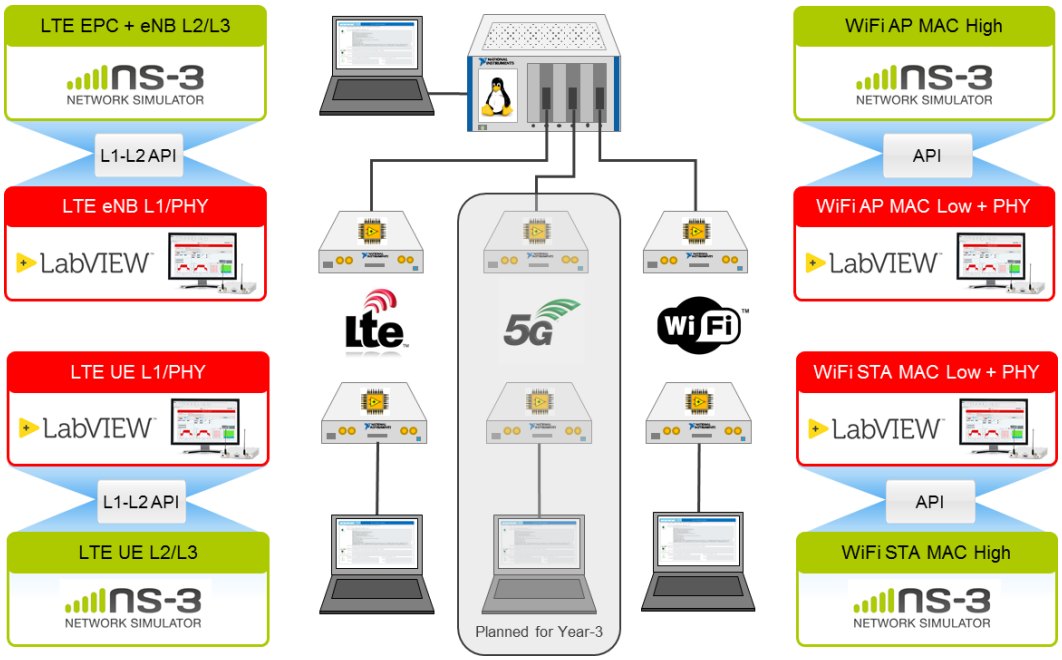


Figure 58: LTE WiFi Multi-RAT Prototyping System.

7.1.2.3 LabView 5G MAC Implementation

As outlined in Figure 45 the generalized L1-L2 API in the LTE link can be used to connect different higher layer protocols. A full stack example written in C/C++ was described in the previous subsections. This section will focus on a LabVIEW based implementation of a MAC layer which supports 5G-NR

MAC protocol format.

The main motivation is to extend the described Multi-RAT platform with a 5G system. As described in the implementation plan for the next year the TUD 5G GFDM PHY will be used as physical layer. So far 5G higher layer protocols are missing in NS-3. That's why we plan to extend the TUD 5G GFDM PHY with an own MAC layer implementation which supports 5G-NR protocol format. As starting point for the development, we used the NI LTE Application Framework 2.2 [24] based on 3GPP Rel.10 [28]. Since the support of the generalized L1-L2 API and real-time capability [see Section 7.1.2.1] was implemented this year, it is now possible to develop real-time MAC functionality on top. The focus of this MAC implementation is on downlink scheduling including link adaptation, supporting end-to-end user plane data transfer. Since this proof-of concept was successfully demonstrated at Globecom conference, the plan for the next year is to connect the LabVIEW based MAC with the LabVIEW based TUD 5G GFDM PHY.

Because of the high effort and complexity, a full featured 3GPP standard compliant MAC layer for eNodeB and UE is not achievable within the ORCA timeframe. Thus, we pointed out and successfully implemented the following main features which are from high interest for MAC research:

- DL scheduling for multiple users in time division multiplex (TDM).
- FAPI [29] based MAC Scheduler supporting two algorithms.
    - Round robin (RR).
    - Time Domain Maximum Throughput (TDMT).
- Dynamic Link adaption based on measurement reporting in uplink.
- Support of 5G-NR and LTE MAC protocol format.
- User Plane traffic for multiple users.
- NI L1-L2 API support on top of NI LTE AFW 2.2.

In contrast to WIFI the cellular MAC Layer in LTE is asymmetric between eNodeB and UE. It coordinates access to the physical layer for all users in a centralized way. At eNodeB the central element for the scheduling mechanism is the MAC Scheduler which is not specified by standardization body and thus an important IP differentiator. The UE has to follow the scheduling decisions provided by the eNodeB and has to report feedback as outlined in Figure 59. Scheduling can be done in both directions (downlink and uplink). Downlink scheduling is initiated by the eNodeB which analyzes scheduling feedback such as channel quality and other UE parameters. Uplink scheduling will be explicitly requested by the UE initiating a scheduling request which is out of scope of this implementation.

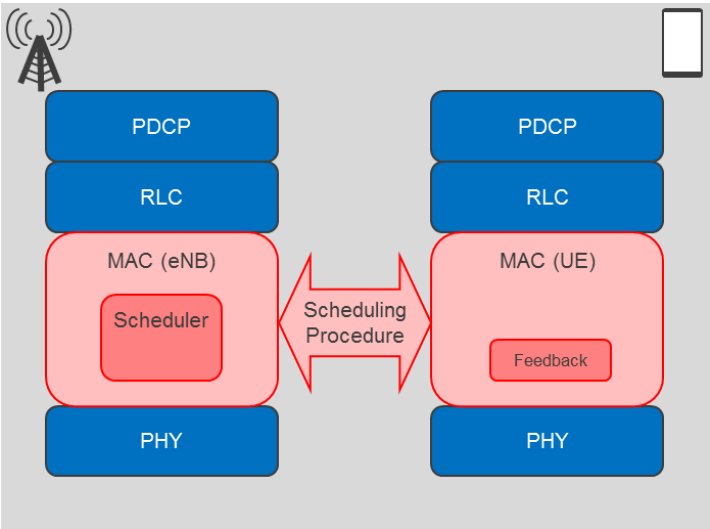


Figure 59: Multiuser Scheduling main elements.

Derived from the planed feature set a MAC Architecture was designed as outlined in Figure 60.

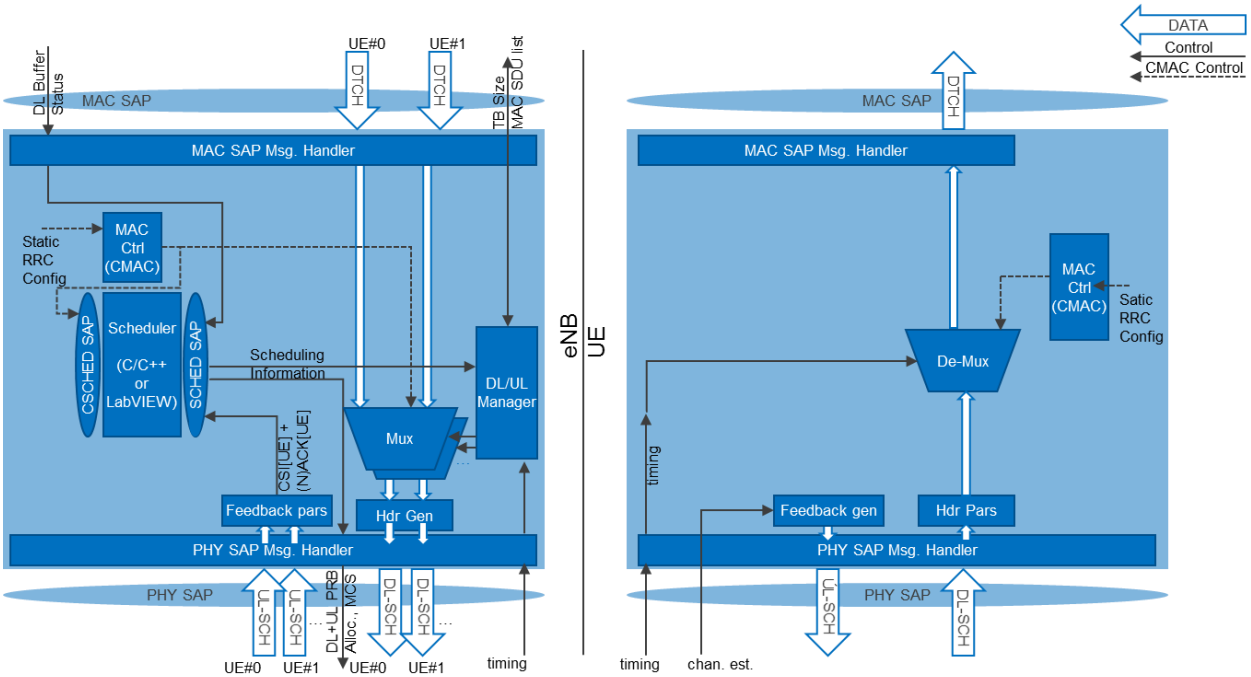


Figure 60: High Level MAC Architecture.

On the eNodeB (eNB) important building blocks such as Scheduler, MAC PDU (Dis-)Assembly (Mux, DL/UL Manager) including 5G-NR Header parsers and generators were implemented. On UE, also MAC PDU (Dis-)Assembly blocks including measurement feedback reporting based on channel estimate are implemented. The PHY service access point (SAP) provides the interface to the physical layer using the L1-L2 API. The MAC service access point is envisaged and currently the UDP data interfaces are located at this level.

In the following important modules are described in more detail:

<u>FAPI based Scheduler</u>

As already described the eNB MAC scheduler and its scheduling algorithms are not specified by standardization body. But there are many scheduler algorithms available in the open-source community. Here the FemtoForum (today Small Cell Forum [25]) originated LTE MAC scheduler API (FAPI) plays an important role. The goal of this interface specification is to allow the use of a wide range of schedulers which can be plugged into the eNodeB and to allow for standardized interference coordination interface to the scheduler. An updated version 2.0.0 [29] with enhanced feature set and bug fixes was developed by IS-Wireless [30] and Eurecom [31] in the context of the EU FP7 project SOLDER. To benefit from this generalized, widely spread interface the NS-3 LTE MAC scheduler which is also based on FAPI was separated from the NS-3 simulator into a linux shared library and integrated via the LabVIEW shared library interface to the LabVIEW based MAC following a test driven approach outlined in Figure 61 and Figure 62.

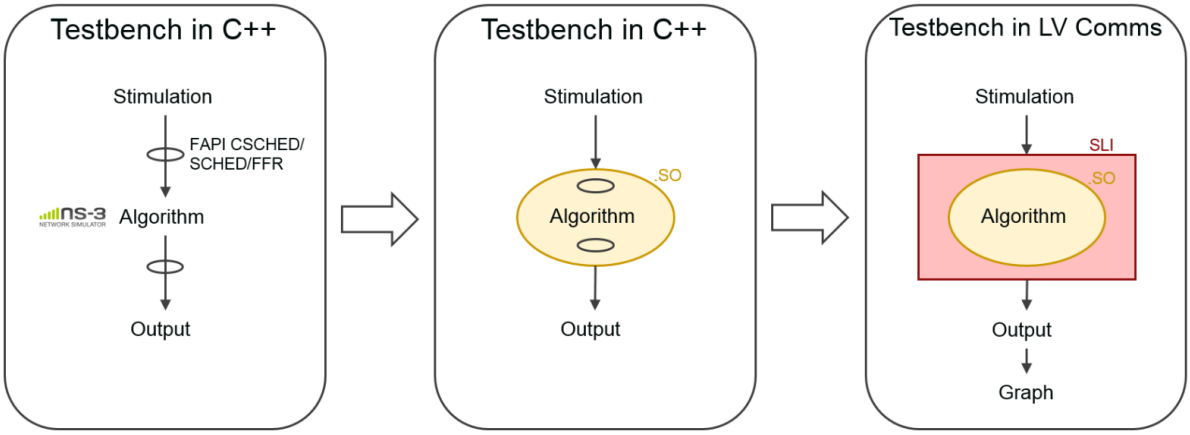


Figure 61: Integration process of the C++ based ns-3 scheduler into LabVIEW.

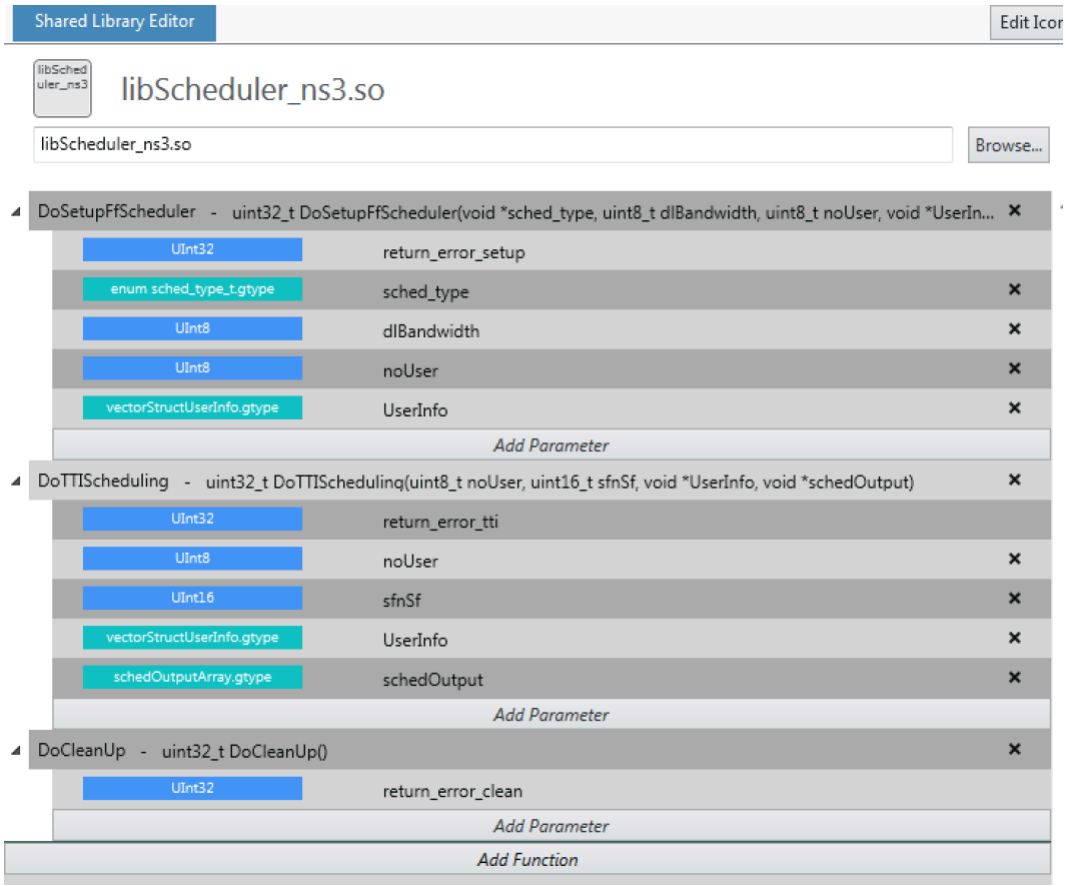


Figure 62: Shared Library Interface (SLI) declares the functions, parameters and return values of the C-functions.

The scheduler architecture and its interface primitives are visualized in Figure 63. The scheduler interface is connected via two service access points (SAPs) to the LabVIEW implementation. The green highlighted items are implemented in order to achieve scheduling in downlink.

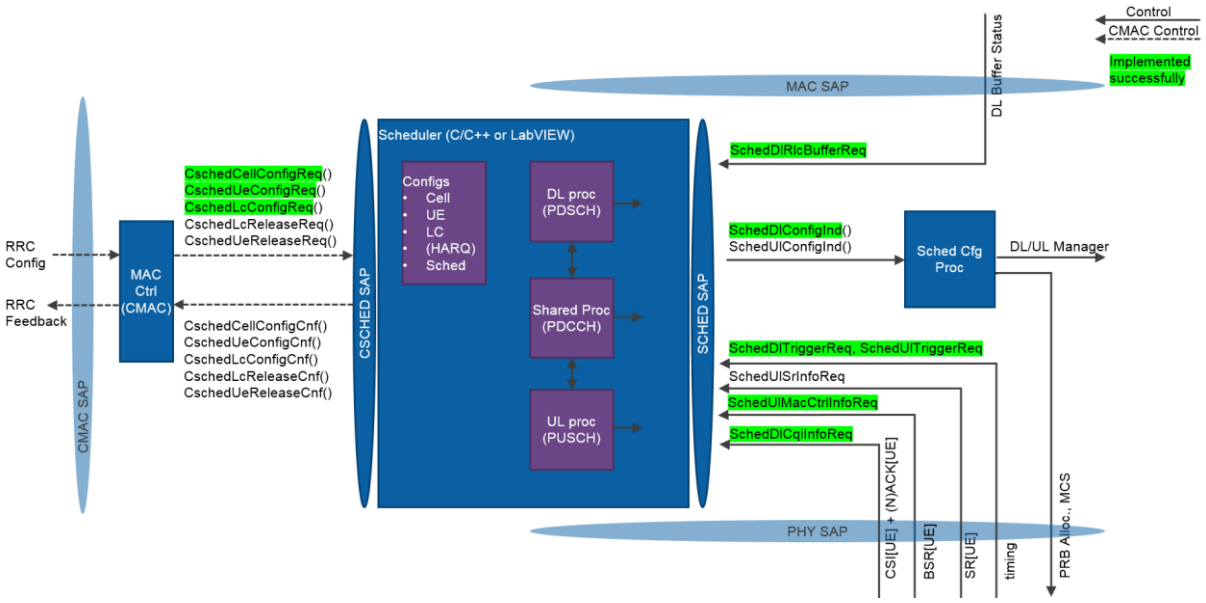
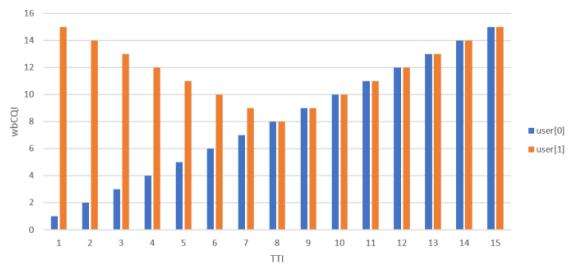
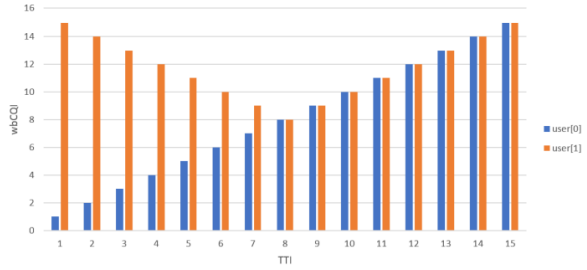


Figure 63: Scheduler architecture and interfaces.

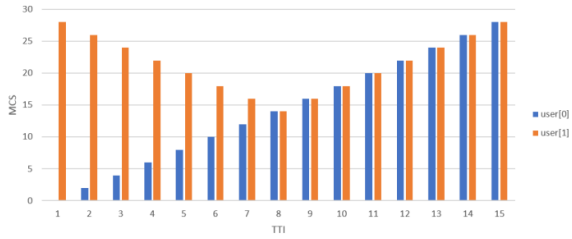
As a starting point, two different scheduling algorithms are made available. The Round Robin algorithm is one of the simplest scheduling algorithms. It divides the available resource blocks equally among all active users and calculates the MCS depending on the received wideband Channel Quality Indicator (wbCQI). As a second algorithm the Time Domain Maximum Throughput scheduler shall maximize the throughput. Therefore, the resources are allocated to the user, who can achieve the highest rate calculated by wbCQI. In the case where several UEs having the same achievable rate, the ns-3 implementation selects the first UE, which has been configured at first. The algorithm does not provide any fairness. Scheduling results for these tow algorithms are exemplary compared in Figure 64.



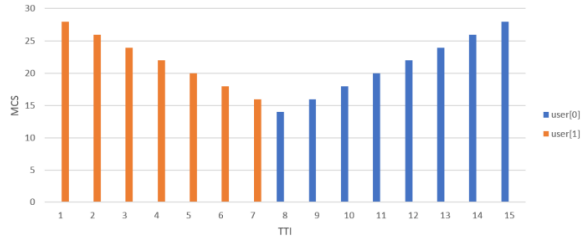
Input parameter: varying CQI and 100 PRBs for both users.



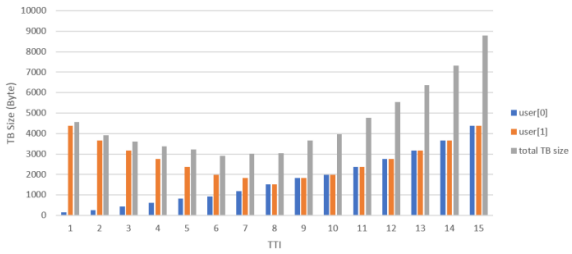
Input parameter: varying CQI and 100 PRBs for both users.



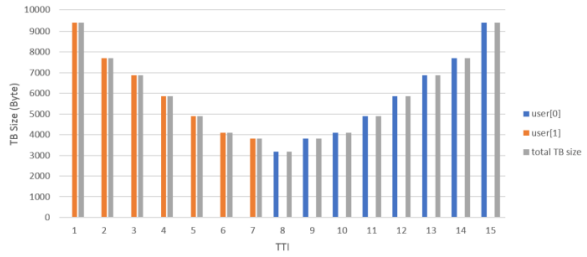
Output parameter: MCS maps exactly to the CQI and is independent from the number of users.



Output parameter: MCS maps to the CQI but only the user with the better channel quality gets resources.



Output parameter: The resources are divided into equal parts. The MCS and the number of PRBs decide about the TB size.



Output parameter: Only one user is scheduled and gets 100 PRBs. The actual TB size is determined by the MCS.

Figure 64: RR Algorithm (left) vs. TDMT algorithm (right).

MAC PDU Assembly/Dis-Assembly

In order to transfer data between eNB and UE over the downlink shared channel (DL-SCH) the MAC layer has to handover a MAC PDU also called transport block (TB) to the physical layer. The size of MAC PDU and modulation and coding scheme (MCS) is defined by MAC scheduler described above. Two MAC protocol formats are supported. First LTE MAC protocol format was implemented according Release 10, 3GPP TS 36.321 [32]. To support also a 5G flavor of medium access control, 5G-NR MAC protocol format was implemented according Release 15, 3GPP TS 38.321 [33]. An overview about the 5G-NR MAC PDU Assembling implementation is given in Figure 65 where an incoming user plane data (UDP packets) are assembled to a MAC PDU in order to utilize bandwidth of the given radio access technology (RAT). With this real end-to-end data transfer is possible to drive end-to-end applications required by ORCA experiments.

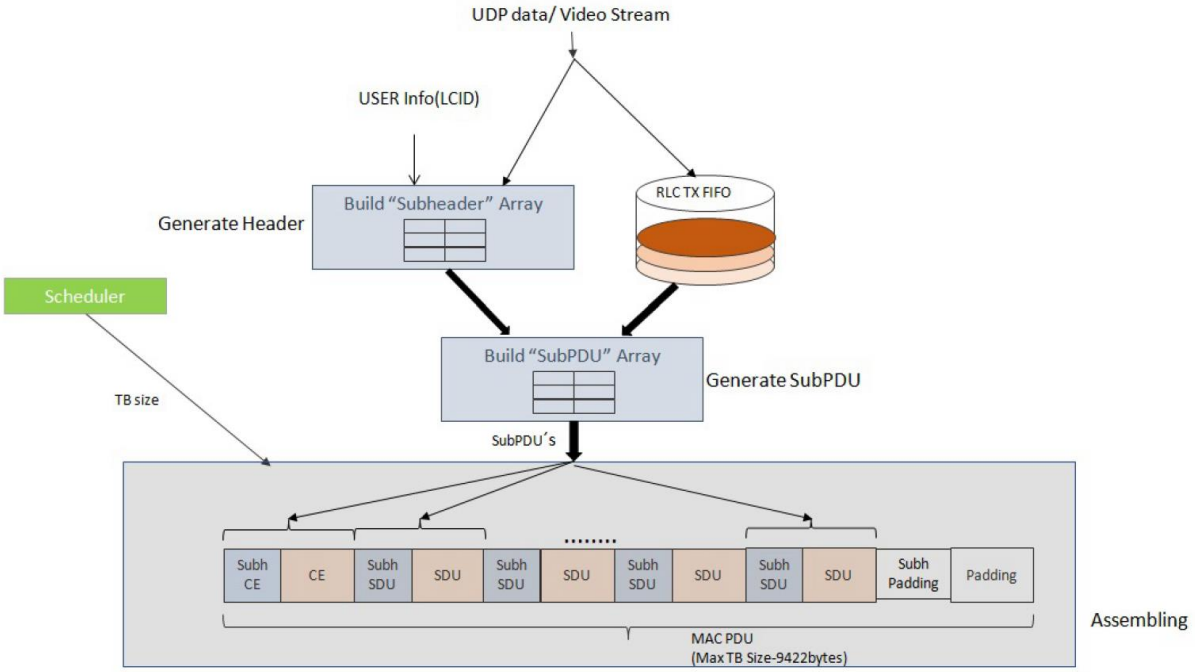


Figure 65: 5G-NR MAC PDU assembling.

The complete 5G-NR LabVIEW MAC implemented in LTE Application Framework v2.2 is shown in Figure 66. In the downlink user plane data can be provided from the eNB to different UEs while channel measurement reports are transmitted in the uplink from the different UEs back to the eNB. With this closed loop operation, the scheduler can assign dynamically the corresponding resources to the UEs in order to achieve dynamic link adaptation. Since dynamic UL control via UL-Grants is currently not implemented in the physical downlink control channel (PDCCH) the UEs permanently transmit UL by a predefined pattern. The eNB switches accordingly between the users by dynamically reconfiguring Radio Network (Temporary) Identifier (RNTI) in order to receive this control data.

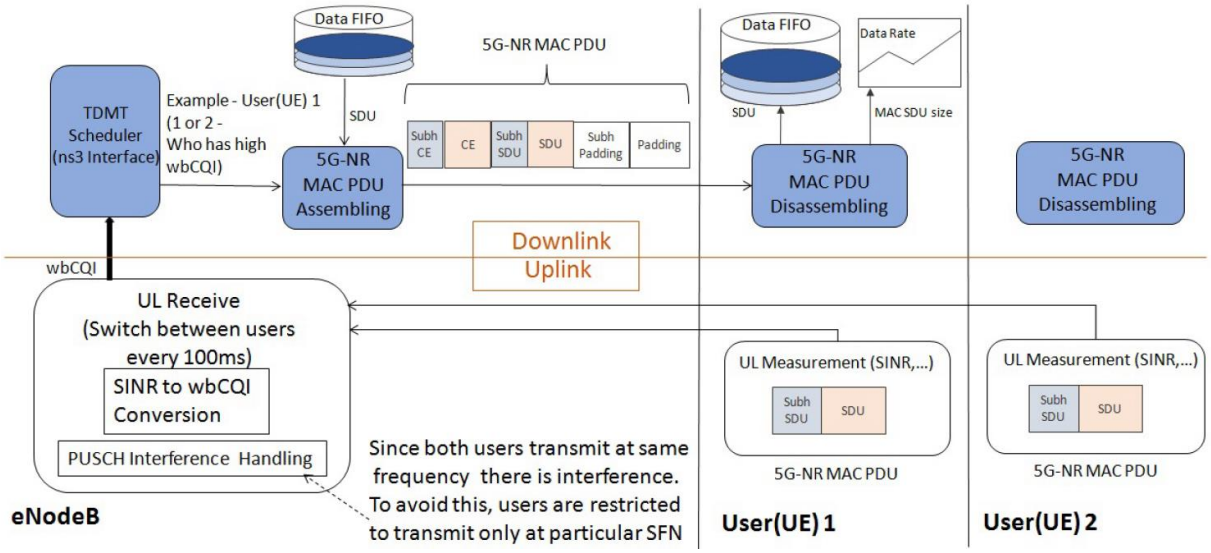


Figure 66: 5G-NR LabVIEW MAC in LTE AFW 2.2.

The complete running demo setup with parallel streaming of two full HD videos from eNB to the UEs is shown in Figure 67. For eNB and UE different graphical user interfaces (GUIs) where designed to visualize the real-time MAC behavior.

Figure 67: LabVIEW 5G MAC demo at Globecom Conference 2018.

7.1.3 Testbed integration

The abovementioned extensions of the Multi-RAT platform are made available in the TUD testbed. Especially, the L1/L2 API and real-time capability in the NI LTE Application Framework is rolled out at the testbed in form of a software update. The ns3 integration of the L1/L2 API for LTE will also be made available by rolling out the latest stable version of the ns3 Multi-RAT platform software trunk to the TUD testbed. The use of the LabVIEW MAC layer implementation will be made available on demand depending on the need of an experimenter for this type of functionality.

7.2 Relation to showcase

As described in deliverable D2.1 [34] and D2.4 [35], the showcase for the NS-3 based prototyping platform is showcase 4: “Interworking and Aggregation of Multiple Radio Access Technologies”.

7.3 Risk analysis

A risk during the development can be the GFDM PHY layer integration towards the LabVIEW 5G MAC implementation for Y3. The LTE API used as baseline for this integration needs to be checked whether it can fulfil the GFDM requirements.

7.4 Implementation plan

7.4.1 PHY improvements

For a possible integration of a 5G PHY layer into the Multi-RAT system, the Application Framework implementation might need to be adapted. This will be the focus regarding PHY improvements for Year 3.

7.4.2 Higher layer protocol integration

The plan for final year 3 of ORCA is to further integrate the Multi-RAT platform to become a concise and complete solution for experimenters working on Multi-RAT interworking research. The focus will be on the integration of the GFDM 5G link including MAC functionality into the overall Multi-RAT scenario. An analysis will be carried out on the best approach to interface the 5G link into the setup.

The platform should also be further enhanced regarding platform independence. In Year 3, the goal is to analyse other software stacks besides ns3 for the potential to be used. This analysis again incorporates, to assess options for interfacing these stacks to the NI SDR platform deployed in the TUD testbed.

The implementation of interworking functionality between different LTE stacks is part of the Open Call 2 for Extensions where Dual Connectivity will be implemented. After this effort is finished, the goal for Year 3 is the final integration of this implementation in the overall Multi-RAT experimentation stack that includes the previously implemented interworking functionality.

7.4.3 Testbed integration

If the 5G Transceiver is successfully integrated into the Multi-RAT platform, the whole setup should be made available for experiments to researchers. With this, a broad variety of experiments between three different RATs can be envisaged.

Furthermore, an analysis will be carried out, on how to efficiently use the underlying SDR hardware in terms of computational complexity. While the current assumption is, that each entity in the system needs an own Real-time controller, further investigations are drawn to options for efficiently running, e.g., two NI Application Frameworks on one CPU, e.g., as a single Multi-RAT base station.

© ORCA Consortium 2017-2020

8 CONCLUSIONS

In conclusion, this deliverable reports on essential data plane SDR functionality to achieve low latency and high throughput operation by allowing real-time operation. It focuses especially on the progress and extensions made in Year 2. Therefore, ORCA extends the current state-of-the-art SDR capabilities by designing 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 partner has brought in new functionalities which can be summarized as follows:

- KUL – a new multi-cell combining vector which reduces computational cost without sacrificing spectral efficiency, and an analysis of spectral efficiency under different levels of cooperation between two networks.
- TUD – described the improvements of the GFDM PHY, e.g., RF control and algorithms. In addition, latency measurements were performed where the GFDM PHY proved to be suitable for low-latency applications.
- IMEC – a new feature in physical layer to improve the latency performance of wireless link formed on the ZYNQ SDR by reducing the Tx/Rx switching time to virtually zero.
- TCD – a software layer that sits on top of radio hypervisors and enables their use for enabling RANaaS. It allows the negotiation of radio resources and the instantiation of radio slices on demand.
- NI – a real-time capable L1/L2 API to interface NI SDRs to ns3 with addition of a LabVIEW based 5G MAC implementation. mmWave phase noise compensation algorithms and a 2D codebook beam characterization.

© ORCA Consortium 2017-2020

9 REFERENCES

[1] ORCA Deliverable D4.3, "Enhanced operational SDR platforms with end-to-end capabilities", December 2018

[2] U. Mengali und A. N. D'Andrea, Synchronization Techniques for Digital Receivers. 1997.

[3] X. Wang et al, "28 GHz Multi-Beam Antenna Array Based On A Compact Wideband 8×8 Butler Matrix" 2018 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting, 8-13 July 2018 Boston, Massachusetts.

[4] ORCA Deliverable D2.2, "Technical requirements of the ORCA test facility", June 2017

[5] ORCA Deliverable D3.1, "First operational real-time SDR platforms", December 2017

[6] Li, X., Bjornson, E., Larsson, E. G., Zhou, S., and Wang, J. (2015, December). A multi-cell MMSE detector for massive MIMO systems and new large system analysis. In Global Communications Conference (GLOBECOM), 2015 IEEE (pp. 1-6). IEEE.

[7] H. Bolcskei and F. Hlawatsch, "Discrete zak transforms, polyphase transforms, and applications," IEEE Transactions on Signal Processing, vol. 45, no. 4, pp. 851–866, April 1997.

[8] M. Matth´e et al., "Precoded GFDM transceiver with low complexity time domain processing," EURASIP Journal on Wireless Communications and Networking, vol. 2016, no. 1, p. 138, 2016.

[9] D. Guo and L. Zhang, "Virtual full-duplex wireless communication via rapid on-off-division duplex," in 2010 48th Annual Allerton Conf. on Communication, Control, and Computing (Allerton), Alberton, IL, USA, 2010, pp. 412–419.

[10] ANALOG DEVICES: user guide of AD-FMCOMMS2-EBZ an FMC board for the AD9361. Accessed: July 26, 2018. [Online]. Available: https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms2-ebz

[11] Ettus Research: the Universal Software Radio Peripheral (USRP). Accessed: Aug. 16, 2018. [Online]. Available: https://www.ettus.com/

[12] ANALOG DEVICES: user guide of AD-FMCOMMS3-EBZ an FMC board for the AD9361. Accessed: July 26, 2018. [Online]. Available: https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms3-ebz

[13] ANALOG DEVICES: user guide of AD-FMCOMMS5-EBZ an FMC board for the AD9361. Accessed: July 26, 2018. [Online]. Available: https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms5-ebz

[14] TEXAS INTSTRUMENTS: datasheet of CC2538 Powerful Wireless Microcontroller System-On-Chip for 2.4-GHz IEEE802.15.4, 6LoWPAN, and ZigBee® Applications. Accessed: July 26, 2018. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2538.pdf

[15] Standard for Low-Rate Wireless Networks, IEEE 802.15.4, 2015.

[16] XILINX: an overview of Zynq-7000 SoC Data Sheet. Accessed: July 26, 2018. [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf

[17] MINICIRCUITS: data sheet of Mini-Circuits' ZN2PD2-63+ high-power splitter/combiner. Accessed: July 26, 2018. [Online]. Available: https://ww2.minicircuits.com/pdfs/ZN2PD2-63+.pdf

[18] ANALOG DEVICES: data sheet of AD9361 RF Agile Transceiver IC. Accessed: July 26, 2018. [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/AD9361.pdf

© ORCA Consortium 2017-2020

[19] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE 802.11- Part 11, 2016.
[20] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 19. IEEE 802.11ad™ - Part 11, 2012.
[21] ANRITSU: operational manual of MS2690A/MS2691A/MS2692A signal analyzer. Accessed: July 26, 2018. [Online]. Available: https://dl.cdn anritsu.com/en au/test measurement/files/Manuals/Operation Manual/MS269xA/MS269xA_Mainframe_Operation_Manual_e_40_0.pdf.
[22] ORCA Functionality: Generalized API, https://orca-project.eu/wp-content/uploads/sites/4/2017/09/ORCA_2.2-08-v1.1.pdf, Dec 2018
[23] LabVIEW Communications LTE Application Framework 1.1, White Paper, online available: http://www.ni.com/white-paper/52524/en/, March 2017
[24] LabVIEW Communications LTE Application Framework 2.5, Manual, October 2018
[25] Small Cell Forum, FAPI and nFAPI specifications, http://www.scf.io/en/documents/082_-_nFAPI_and_FAPI_specifications.php, May 2017
[26] LabVIEW Communications 802.11 Application Framework 2.1, Manual, March 2018
[27] ns-3 Model Library, Manual, September 2018
[28] 3GPP TS 36.211 "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation", Rel 10, March 2013
[29] FAPI 2.0.0 - Small Cell Forum originated LTE MAC scheduler API, http://www.eurecom.fr/~kaltenbe/fapi-2.0/index.html, Dec 2018
[30] IS Wireless, https://www.is-wireless.com, Dec 2018
[31] Eurecom, Sophia Antipolis, http://www.eurecom.fr/en, Dec 2018
[32] 3GPP TS 36.321" Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification", Rel 10.10, January 2014
[33] 3GPP TS 38.321 "NR; Medium Access Control (MAC) protocol specification", June 2018
[34] ORCA Deliverable D2.1, "Definition of showcases", March 2017
[35] ORCA Deliverable D2.4, "Development and integration of showcases in Year 2", December 2018