REINDEER

D5.1

Detailed experimental validation plan, design of experiments, and definition of common formats

Project number:	101013425
Project acronym:	**REINDEER**
Project title:	REsilient INteractive applications through hyper Diversity in Energy Efficient RadioWeaves technology
Project Start Date:	1st January 2021
Duration:	42 months
Programme:	H2020-ICT-52-2020
Deliverable Type:	Report
Reference Number:	ICT-52-2020 / D5.1 / 1.00
Workpackage:	WP5
Due Date:	31st May, 2023
Actual Submission Date:	31st May, 2023
Responsible Organisation:	KU Leuven
Editor:	Gilles Callebaut
Dissemination Level:	PU
Revision:	1.00
Abstract:	Detailed experimental validation plan, design of experiments, and definition of common formats. This report provides a detailed experimental validation plan. The design of experiments describes the distributed RadioWeaves topologies to be realized in the KU Leuven testbed and the particular experiments to be performed in the ULund testbed. D5.1 also defines formats for interfacing with RadioWeaves panels and for the exchange of experimental data.
Keywords:	RadioWeaves, experimental research, testbeds, energy-neutral devices, Data Storage Standard

The REINDEER project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101013425.

Editor

Gilles Callebaut (KU Leuven)

Contributors

Gilles Callebaut, Jarne Van Mulders, Bert Cox, Liesbet Van der Perre (KU Leuven)

Liang Liu, Xuhong Li, Christian Nelson, William Tärneberg, Emma Fitzgerald, Ove Edfors (ULund)

Benjamin J. B. Deutschmann, Thomas Wilding, Klaus Witrisal (TU Graz)

Koen Deforche (BlooLoc)

Ulrich Mühlmann (NXP)

Internal reviewers

Andres Reial (EAB)
Sai Subramanyam Thoota (LiU)

Disclaimer

Executive Summary

This deliverable provides an overview of the experimental validation plan and the various testbeds and simulators used in the Reindeer project. The move towards a distributed deployment in RadioWeaves (RW) presents new challenges related to the geographical dispersion of processing resources and a high number of contact service points (CSPs). To address these challenges, a detailed experimental validation plan has been developed.

To facilitate data exchange and processing across different testbeds, a common data format and interface have been designed, *i.e.*, Dataset Storage Standard (DSS). This allows for convenient creation of processing scripts in a testbed-agnostic manner, enabling sharing of scripts among developers and researchers without the need for intermediate adaptations. Additionally, the DSS has been developed to store data from both experiments and simulations, enabling the processing of results using the same scripts and tools.

The Reindeer project involves multiple testbeds and energy-neutral devices (EN devices) to validate and assess proposed algorithms, protocols, and architectures. These testbeds and EN devices are categorized based on their scope and intention. In addition to the testbeds, two complementary simulators, TugSim and LuSim, are being developed to extend experiments to complex and varying scenarios, including hardware in the loop simulations.

The herein-defined experiments aim to evaluate the key performance indicators (KPIs) defined in D1.1. The experimental results obtained in WP5 will serve as inputs for broader and more complex deployments, as outlined in the use case evaluation in T1.3. Therefore, WP5 plays a crucial role in experimentally investigating the KPIs of the RW concept. Overall, the combination of testbeds, simulators, and experiments conducted in the Reindeer project will contribute to the advancement and validation of RadioWeaves technology and its potential for distributed deployment in 6G systems.

Contents

Glossary

1PPS pulse per second.

3D three-dimensional.

6G sixth-generation.

ADC Analog-to-digital converter.

AGC automatic gain controller.

AOA angle-of-arrival.

AP access point.

API application program interface.

AR augmented reality.

ASK amplitude-shift keying.

BS base station.

CCS correlative channel sounder.

CF cell-free.

CPU central-processing unit.

CSI channel state information.

CSP contact service point.

DAC digital-to-analog converter.

DAQ data acquisition system.

DMA Direct Memory Access.

D-MIMO distributed MIMO.

DPDK Data Plane Development Kit.

DSS Dataset Storage Standard.

ECSP edge computing service point.

EN energy neutral.

EN device energy-neutral device.

FD front-haul distance.

FPGA field-programmable gate array.

GNSS global navigation satellite system.

GPU graphical processing unit.

HAT hardware attached on top.

ICCS Ilmsens correlative channel sounder.

IO input/output.

IQ in-phase and quadrature.

KPI key performance indicator.

LIS large intelligent surface.

LoS line-of-sight.

LSA large synthetic array.

MCU microcontroller unit.

MIMO multiple-input multiple-output.

MISO multiple-input single-output.

mMIMO massive MIMO.

mmWave millimeter wave.

MPPT maximum power point tracking.

MRC maximum ratio combining.

MRT maximum ratio transmission.

NI National Instruments.

NLoS non-line-of-sight.

NTP network time protocol.

OFDM orthogonal frequency-division multiplexing.

OOK on–off keying.

P2P point-to-point.

PAPR peak-to-average power ratio.

PD powered device.

PEB positioning error bound.

PHY physical.

PLA physically large array.

PoE power-over-Ethernet.

PS Processing System.

PTP precision-time protocol.

Rb Rubidium.

RF radio frequency.

RFSoC Radio Frequency System-on-Chip.

RIS reflective intelligent surface.

RLS recursive least squares.

RPi Raspberry Pi.

RSSI received signal strength indicator.

RW RadioWeaves.

RX receiver.

SDR software-defined radio.

SMC specular multipath component.

SNR signal-to-noise ratio.

SoC state of charge.

TCP Transmission Control Protocol.

TDD time division duplexing.

TDMA time-division multiple-access.

TDOA time-difference-of-arrival.

TOSM through-open-short-match.

TSN time-sensitive networking.

TX transmitter.

UART universal asynchronous receiver-transmitter.

UE user equipment.

URA uniform rectangular array.

USRP universal software radio peripheral.

UWB ultrawideband.

VEP virtual edge platform.

VNA vector network analyzer.

VR virtual reality.

WPT wireless power transfer.

WR White Rabbit.

XETS cross exponentially tapered slot.

ZF zero-forcing.

Chapter 1

Introduction

In this chapter, we introduce a testbed designed for investigating the capabilities of distributed MIMO (D-MIMO) systems in the context of sixth-generation (6G) networks. We discuss the requirements and challenges associated with real-time signal processing and the move towards distributed processing architectures. Additionally, we highlight the importance of open testbeds and the need for common data formats and interfaces to facilitate data exchange and analysis. The chapter also outlines the structure and contents of the dataset storage interface.

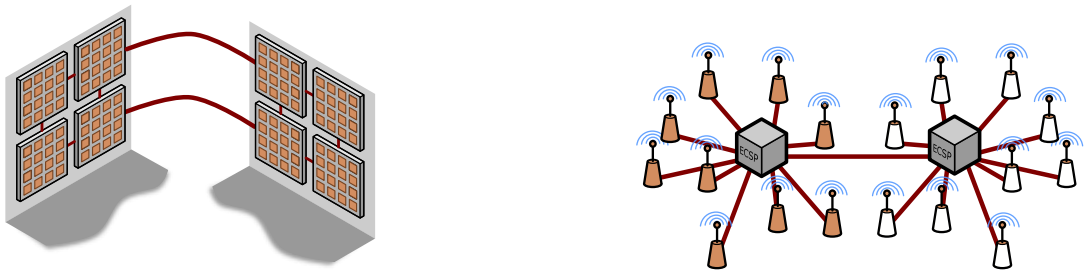
1.1 Deliverable Outline

The subsequent sections provide an explanation of the testbed requirements and the common data format. This information aims to provide context and facilitate understanding of the intended functionalities and data exchange methods of the testbeds. In Chapter 2, we present the experimental RadioWeaves (RW) testbeds and energy-neutral devices (EN devices). We begin by introducing various testbed levels that indicate the capabilities of our platforms, ranging from channel sounding to real-time applications. Each available testbed and EN device is described in detail. Chapter 5 introduces two complementary RW simulators. In Chapter 4, we highlight the key performance indicators (KPIs) of the REINDEER project and specify the testbeds capable of assessing each KPI. We explain five main experiments designed to address these KPIs. Finally, Chapter 5 concludes the discussion and provides insights into the planned future work.

1.2 Testbed requirements

The move towards a distributed deployment in RW, necessitates a number of new methods to cope with, including i) the geographical dispersion of processing resources and ii) the high number of contact service points (CSPs). On top of this, providing real-time operation of a D-MIMO testbed imposes new challenges. We briefly highlight some critical requirements for the testbed, before introducing the developed testbeds in the Reindeer project.[1]

[1] This section contains work published in **empty citation**



(a) Decentralised antenna arrays with distributed processing resources in each array.

(b) Fully distributed antennas connecting to edge computing service points (ECSPs) for centralized processing.

Figure 1.1: Distributed and centralized processing architectures applied in mMIMO systems with different flavours of antenna distributions.

1.2.1 Offline vs. real-time signal processing

Both offline and real-time signal processing are important features for a 6G testbed. Offline processing requires the capability of recording data for a certain time interval for later analysis, giving the flexibility to develop and evaluate a wide range of signal-processing algorithms. On the other side, real-time processing enables system measurement and testing in fast-changing environments. The real-time testbed can also serve as the system-level tester for hardware solutions of specific function blocks (e.g., digital signal processors, data converters, and transceivers).

The first challenge for real-time signal processing is the high throughput requirement. For instance, depending on the selected algorithms, the required processing capability to support real-time multi-user detection can grow significantly with the number of base station (BS) antennas and exponentially with the number of simultaneously served users.

Processing latency is another challenge for real-time testbeds, especially for time-critical use cases and/or in fast-changing environments. There are two processing latency constraints when developing a real-time testbed. The first one is the end-to-end latency, e.g., transmitter (TX)-to-receiver (RX), which is bounded by application requirements. For applications such as real-time digital twins in manufacturing, and human and robot co-working, the end-to-end latency should be around $1\,\mathrm{ms}$ [1]. The second latency constraint relates to time division duplexing (TDD) switching time in reciprocity-based massive MIMO (mMIMO). The process of channel estimation using uplink pilots, calculating precoding coefficients, and transmitting the precoded data all need to be finished within the time interval between the uplink pilot and downlink data in a radio frame [2]. This time interval can be in the sub-millisecond range, depending on the supported coherence time and thus the radio frame structure. For instance, the processing latency constraint is around $0.4\,\mathrm{ms}$ to support $70\,\mathrm{km\,h^{-1}}$ mobility at $3.7\,\mathrm{GHz}$ [2]. The latency requirement is even more critical for decentralised processing architectures, where the data transfer between processing units may take a substantial time.

1.2.2 Moving to distributed processing architectures

Independent of the physical locations of the antennas (colocated or distributed), there are various processing strategies that can be applied, affecting the data transfer between processing units in different ways. Here, we will focus on the two main processing strategies, namely centralised and decentralised processing [3], [4], as illustrated in Figure 1.1. The centralised solution faces critical implementation challenges when up-scaling the number of antennas and the sig-

nal bandwidth, especially considering the aggregated data rate to the central processing unit. For a 6G mMIMO system with 256 BS antennas, 25 single-antenna users, and 100 MHz signal bandwidth, the aggregated data rate reaches around $750\,\mathrm{Gb/s}$, exceeding the data rate limit of commonly available interconnection solutions. The aggregated data may be distributed over different physical links; however, the number of input/output (IO) connections are generally limited on a practical processing hardware platform. In addition to dedicated interconnects, a candidate medium for transmitting the data is Ethernet, where software is being developed, for example, Data Plane Development Kit (DPDK) [5], to enable high-throughput real-time data transfer.

Decentralised processing (including distributed processing algorithms and the corresponding distributed architectures) enables closer-to-antenna processing and avoids extensive data aggregation, and thus is more suited for scalable implementations. This method follows more closely the generic architecture, using ECSPs and CSPs as shown in Figure 2.1. For example, the recursive least squares algorithm is a decentralised implementation of the zero-forcing (ZF) detection, which is mapped to a Daisy chain topology to distribute data transfer only between neighbouring processing units in the chain [3]. However, this implies a long processing latency to propagate the signal processing serially through the entire chain. This could be partially overcome by applying more parallel topologies, e.g., mesh and tree [4]. In physically distributed mMIMO systems, numerous antennas are expected to be available around (and close to) the users. Algorithms with local-only processing, e.g., maximum ratio transmission (MRT), are likely to provide good enough system performance and approach the centralised ZF performance. The corresponding data transfer between processing units can then be minimised.

Data storage is another challenge for implementation in the centralised solution. The required memory size to store the channel matrices of 100 MHz 256×25 mMIMO system is 380 Mb (24 bits per channel matrix element). In the distributed processing architecture, data storage can be distributed over many processing units, making it possible to use more cost and energy-efficient on-chip memory solutions for each processing unit.

1.2.3 Open Testbeds

To facilitate widespread adoption and interoperability, 6G systems require open standards, interfaces, data sets, hardware, and software. There is a need for interoperability and openness in next-generation networks. Following these principles, the developed testbeds will also pursue an open approach. Consequently, a publicly available website is developed `https://6g-testbeds.github.io/`, allowing to constantly update it to reflect the current status of the testbeds. Furthermore, as these testbeds transcend the Reindeer project, this webpage permits us to keep updating this page with the latest information, even after the finalisation of the project.

Accompanying open testbed infrastructure and interfaces, data and open tools to process the generated (raw) data should be accessible. Furthermore, the data should satisfy the FAIR principles [6], that is, findability, accessibility, interoperability, and reusability. This entails that the data is accompanied by well-described metadata, documentation, and (preferably) open licences. This information, together with the data, will be hosted on an available publicly-accessible server of an Open data platform, e.g., IEEE DataPort, Open Data on AWS, Kaggle and data.europa.eu (DEU).

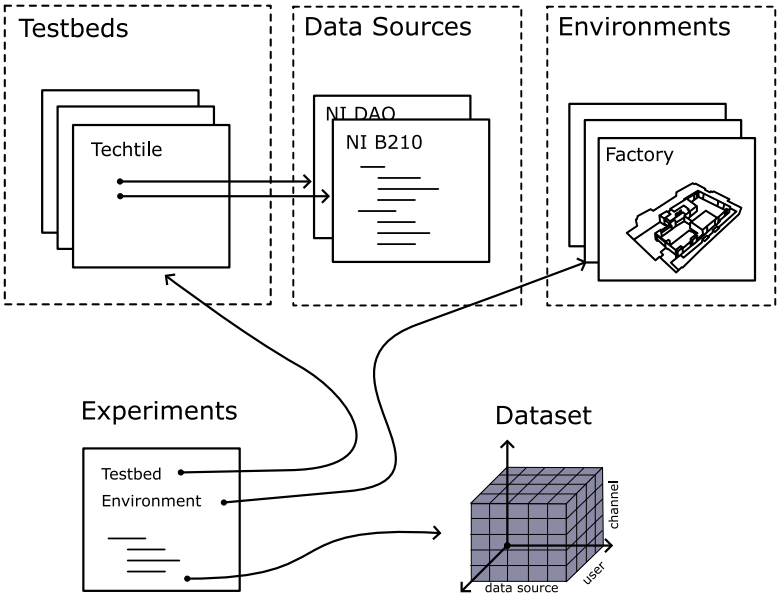


Figure 1.2: Illustration of the proposed standard, containing the description files and the dataset.

1.3 Common Data Formats and Interface

A common data format and interface are designed to i) facilitate exchanging data and ii) to conveniently create processing scripts in a testbed-agnostic fashion. The latter entails that processing scripts can be shared among the testbed developers and researchers, without intermediate scripts to tailor it to the format of the data and metadata. Dataset Storage Standard (DSS) has been developed due to a lack of reference data sets, common tool sets and standardized metadata. Moreover, the manual process to describe the experiment and the stored data with comprehensive metadata is cumbersome and therefore lacks sufficient metadata in existing datasets.

In contrast to existing endeavours, DSS focuses on testbed-agnostic data storage, originating from experiments or simulations, in contrast to e.g., of National Instruments (NI) NI RF Data Recording **empty citation** which is tailored for NI products.

While originating from the Reindeer project, the standard is currently being designed by both Reindeer-internal and external researchers. The open-source interface documentation, including examples, is hosted on GitHub: `https://github.com/6G-Testbeds/Dataset-Storage-Interface`. A short summary of the interface is included below, for more details, consult the documentation on the GitHub page.

1.3.1 Architecture

The standard is divided in human-readable files, describing the testbed and experiments, a data storage format and an application program interface (API). These are illustrated in Figure 1.2.

1.3.1.1 Description Files

A number of files are required in order to interpret and explain a conducted experiment. The files are structures such that reuse is possible among different experiments and testbeds. For instance, a *data source* can describe a software-defined radio (SDR), as well as a data acquisition system (DAQ) system. A testbed exists out of several *data sources*. An experiment can

make use of an environment file. The latter describes, e.g., the room dimensions and 3D scans. The experiment includes the testbeds used, a number of measurements each having different parameters. An example is provided below:

```
name: "same name as file"
description: "Describing the file"
testbed: *ref_testbed
scenarios:
 - data_source: [0,100] #optional list of data_source indexes
   data_source_type: "SDR"
   sampling_rate: 250e3
   domain: "freq"
   freq: 433e6
   pos_type: "GPS" # or relative
 - data_source_type: "ADC"
   sampling_rate: 10e3
```

1.3.1.2 Dataset Format

The description files are utilized when reading/storing the data in a common format. A simple data set structure is used to store the data[2] of a specific experiment scenario. A tensor is used with the following dimensions:

- data source
- data
- channel (optional)
- time (optional)
- user (optional)
- positions (optional)

Each dimension can have a coordinate associated to it, i.e., each dimension can have label-based indexing through coordinates. For example, for each position entry in the tensor, a position, e.g., relative or global navigation satellite system (GNSS)-based, can be associated with it.

The data types should be inferred from the dataset and is not imposed by the standard. The timestamp could be inferred from the `start_time` in the experiment description file, but this is not required. Based on the start time and the sampling rate, the timestamps can be generated. The channel dimension indicates which channel of the data source is used. For example, some SDRs have multiple RX chains on the same SDR board, and each RX chain is seen as a separate channel. The position of the user can be included in the dataset or in the scenario file, depending on the user's mobility, for instance. If the user is fixed during the scenario, it should be included in the scenario description, otherwise, it can be included in the dataset.

The dataset is stored in an HDF5 to keep interoperability with a range of programming languages.

[2]Note, that we do not impose where the data is stored.

1.3.2 Examples

By standardizing the dimensions of the data, we can conveniently select the source, time, and channel and plot the associated data, as indicated below:

```
# plotting the data at time=0 and the first source
20*np.log10(abs(da.isel(source=0, time=0))).plot(x="data", hue="channel");

# from time to frequency along the data dimension
da_fft = xrft.fft(da, dim='data', shift=True)
plt.plot(20*np.log10(abs(da_fft.isel(source=0, channel=0, time=0))));
```

Chapter 2

Experimental RadioWeaves Testbeds and Devices

2.1 Introduction

Several testbeds and user equipment (UE) devices have been designed to validate and assess the proposed algorithms, protocols, and architectures in Reindeer. An overview of the testbeds can be found in Table 2.1.

RW presents a myriad of new research challenges, each having a different scale, aim, and performance metrics. In order to discuss the different testbeds, we distinguish them based on their scope and intention. An overview of these different levels and a common architecture is presented in Figure 2.1. Following the terminology introduced in [7], a 6G system architecture could exist out of several connected ECSPs, which coordinate one or multiple CSPs and perform joint processing. The term CSP was introduced to indicate that the conventional access point (AP) is expected to support more than just communication in 6G, e.g., wireless power transfer (WPT). This entails that these CSPs will host a variety of hardware such as radio, charging, processing, data storage and other sensing elements. We categorized four levels of testbeds based on their capabilities and intended use. This is illustrated in Figure 2.1.

An overview of the testbeds developed in Reindeer as well as external testbeds are hosted on `https://6g-testbeds.github.io/`. Through this Github repository, in combination with the proposed data storage standard, we aim to address the larger research community to create open and accessible 6G testbeds.

Level 1 - Point-to-point (P2P) Link Evaluation Level 1 testbeds are used to assess one specific component of a system. This can relate to a flexibel antenna configuration setup, as indicated in blue in Figure 2.1. L1 testbeds can also be used in conjunction with simulations to perform “calibrated simulations”, where an experiment consist of *hardware in the loop*.

Level 2 - Channel Sounders To study the new propagation conditions as an effect of distributing resources and having an increased number of antenna elements with respect to conventional mMIMO, channel sounders are used. In essence, a channel sounder consists of one or more TX and RX chains, where the transmitted signals are recorded at the RX, and processed to investigate the propagation channel. Different approaches are used to sample the channel. Virtual

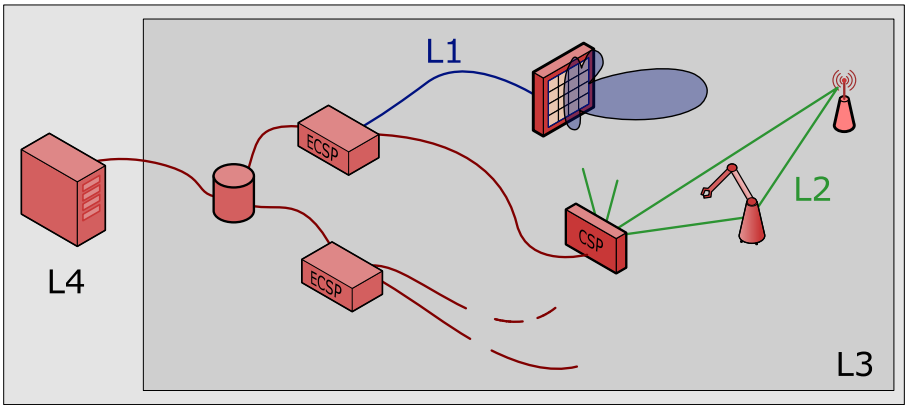


Figure 2.1: Overview of a 6G radio system with different testbed scopes. At level 1 (L1), new hardware and configurations are evaluated. Level 2 (L2), i.e., channel sounders, study the novel propagation conditions in RW scenarios. Level 3 (L3) and 4 (L4) are designed to test, e.g., distributed processing in real-life and real-time conditions, respectively.

arrays can be created by moving a single antenna to multiple locations. Another approach is to have multiple antennas connected to one radio frequency (RF) chain, where the antennas are time-multiplexed via an RF switch.

Level 3 and 4 Testbeds In contrast to the channel sounders, real-life (L3) testbeds do not focus on fine-grained propagation channel knowledge, such as for example angle-of-arrival (AOA), but are designed to investigate end-to-end communication performance, as for instance the network's energy-efficiency. In L3 testbeds, real-life measurements are conducted and processed in an offline stage. L4 testbeds extend L3 by supporting real-time processing of signals to evaluate algorithms requiring real-time interaction between the network and the UEs, unable to be performed offline, cfr. Section 1.2.1.

2.2 RadioWeaves Testbeds

All REINDEER-related testbeds available in the REINDEER consortium are provided in this chapter. It serves as a complete overview, ranging from Level 1 to Level 4 testbeds. To perform the experiments, as detailed in Chapter 4, mainly the Level 3 and 4 testbeds will be used. To mitigate risks, other testbeds can be introduced, potentially complemented with the simulations, to perform the experiments.

Table 2.1: Overview of available testbeds in REINDEER and their specifications. Also available at 6g-testbeds.github.io.

Name	Array type	Number of antennas	bandwidth	Frequency range	Testbed Level	Aim/Focus
LARVA	Synthetic	1 TX / 2 RX	7 GHz	3–10 GHz	L2	Synthetic aperture measurements for positioning and WPT.
RUSK	Virtual	-	240 MHz	3 bands	L2	Synthetic RW and many users for WPT and positioning.
Techtile	140 CSPs	280	56 MHz	70 MHz-6 GHz	L3/L4	Room-sized RW infrastructure with dense CSPs deployment.
KULMaMi	-	64	18 MHz	400 MHz-4.4 GHz	L4	Real-time processing for communication
LUMaMi	-	100	200 MHz	3.7 GHz	L4	Real-time processing for communication, data capture for positioning
LuLIS	16-64 4x4 panels	256-1024	100 MHz	3.8 GHz	L4	-
LUDyna	-	1-N	40 MHz	1.2–6 GHz	L2	Dynamic and multi-link measurements for sensing.

2.2.1 Techtile

The Techtile testbed features a support structure comprising standardized wooden parts and 140 detachable tiles, see Figure 2.2. The infrastructure is built around an Ethernet-based backbone, providing power, data, and synchronisation. Each tile in the testbed can accommodate various sensors, actuators, transmitters, receivers, and processing resources. Overall, Techtile provides a flexible and scalable platform for conducting experiments and research in areas such as network communication, synchronization, sensor fusion, multi-modal sensing, and distributed computing. An overview of the architecture of the testbed including logical links between components is shown in Figure 2.3.

Figure 2.2: Left: The Techtile support structure – Right: The back of three tiles, equipped with the default setup, *i.e.*, a SDR (universal software radio peripheral (USRP) B210), processing unit (Raspberry Pi (RPi) 4) and power supply with power-over-Ethernet. Each tile is connected to the central unit with an Ethernet cable, providing both power and data.

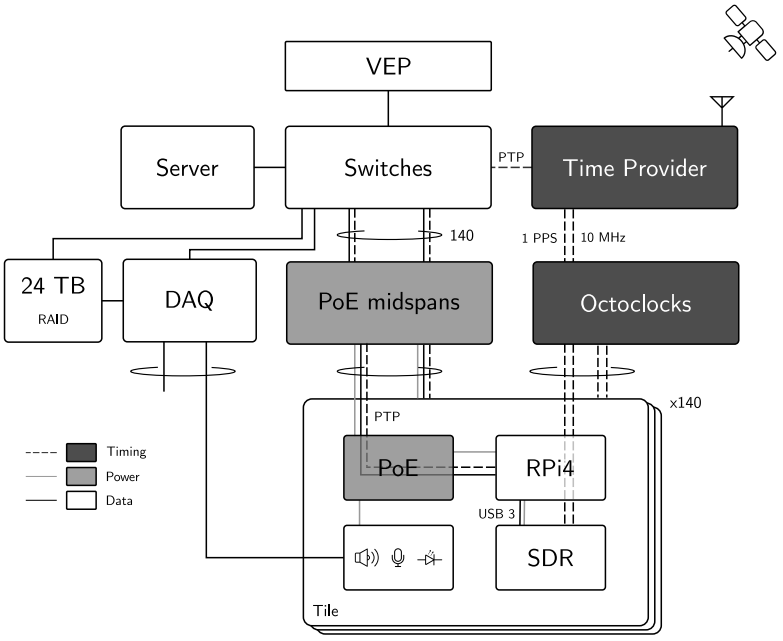


Figure 2.3: Overview of the architecture of the testbed including logical links between components. Blocks highlighted in dark gray distribute synchronization information. Light gray blocks represent provision of power through Ethernet or dedicated terminals and connectors. The solid lines depict connections exchanging data, *e.g.*, Ethernet-based or USB. Dashed lines represent links transferring synchronization information.

2.2.1.1 Construction

Figure 2.2 illustrates the Techtile support structure. It is based on the open-source building concept WikiHouse[1]. Such constructions are made of standardized wooden parts. Building further on the modular design, the walls, floor, and ceiling of the structure are comprised of 140 detachable tiles. The two walls of the building support 28 tiles each. The ceiling and ground floor support 42 and 52 tiles, respectively. The room's dimension is $4\,\mathrm{m}$ by $8\,\mathrm{m}$, with a height of $2.4\,\mathrm{m}$. A tile has a size of $120\,\mathrm{cm}$ by $60\,\mathrm{cm}$.

2.2.1.2 Ethernet-based Backbone

The backbone of the infrastructure consists of a central server, connecting all tiles via Ethernet. $90\,\mathrm{W}$ of power is supplied to each tile by means of power-over-Ethernet (PoE) midspans. The employed Ethernet switches support the IEEE-1588 precision-time protocol (PTP), enabling high-accuracy clock distribution to all connected devices. The PoE technology also keeps cable management practical. Our PoE architecture consists of several PoE midspans[2] supporting a total of 156 powered devices (PDs) (PoE clients). Each midspan has a 10/100/1000 Mbps data rate pass-through. In total, a PoE budget of approximately $9\,\mathrm{kW}$ is available.[3] The testbed, thus, provides communication, synchronization, and power over Ethernet. As everything is connected over Ethernet, the developed system is easily scalable and flexible in the manner in which devices are added and removed from the network. Furthermore, by connecting all tiles using Ethernet,

[1] www.wikihouse.cc

[2] The PD-96XXGC series midspans from Microchip is used.

[3] While $9\,\mathrm{kW}$ is available over PoE, current preliminary calculations indicate a power consumption of $3\,\mathrm{kW}$ under full load. Approximately $1.5\,\mathrm{kW}$ to power the RPis and USRPs, and the remainder for processing and switching.

all different network topologies can be emulated, *e.g.*, tree or mesh. This facilitates investigating the impact of different network topologies discussed in [8].

2.2.1.3 Central Processing and Networking

The central server (Dell PowerEdge R7525) has 512 GB RAM, two NVIDIA Tesla T4 16 GB graphical processing units (GPUs) and two AMD 7302 3 GHz central-processing units (CPUs), running Ubuntu Server 20.04 LTS. At the networking side, four Dell S4148T-ON switches are used, featuring each 48 10 Gbit Ethernet ports with IEEE 1588v2 support. A Dell virtual edge platform (VEP) 1485, running VyOS, handles routing, DHCP, VPN and firewall.

2.2.1.4 Synchronization

Multiple manners to distribute synchronization are available in the testbed to investigate different levels of synchronization, *i.e.*, Ethernet-based, dedicated cabling or over-the-air synchronization.

Ethernet-based By default, all tiles are time synchronized with PTP IEEE 1588v2 [9]. This protocol achieves a clock accuracy in the sub-microsecond range –and depending on the network, configuration, and version even the sub-nanosecond range. The protocol incorporates a master-slave architecture. The root time reference is held by the grandmaster clock[4] and is distributed to the other clocks in the network. Precision-time protocol supports both L2 and UDP transport. It has an operation similar to network time protocol (NTP), where the master and slave exchange messages to determine the path delays and correct their clocks accordingly. In PTP, different profiles are defined, each having different configurations and requirements. Such profiles are tailored for specific applications and are, for example, used in time-sensitive networking (TSN) [10]. Different clocks are defined based on their capabilities. For instance, transparent clocks are network devices that alter the timestamps in the packets to remove the time spent in these devices, effectively making them transparent to the PTP protocol. A boundary clock, on the other hand, serves as both master and slave. It synchronizes its internal clock to a master on a specific port and serves as a master on other ports.

While the IEEE 1588v2 method ensures a scalable and low-cost solution, several other synchronization technologies exist. Recently, another Ethernet-based protocol, White Rabbit (WR) [11], is being integrated in the IEEE 1588 protocol (IEEE 1588-2019) [9]. It extends the conventional IEEE 1588v2 protocol by including clock syntonization (frequency synchronization), phase detection (increasing the timestamp accuracy) and link asymmetry detection. These features result in sub-nanosecond accuracy time synchronization.

Dedicated cabling To provide a reference for time and frequency synchronization, clock distribution modules[5] can be used in the infrastructure. It provides a frequency and time reference via a 10 MHz and 1 pulse per second (1PPS) source, respectively. The clock distribution modules are synchronized via the grand master clock, which is in turn synchronized via GNSS. While this approach is not scalable, it serves as a baseline to compare the deterioration of the system services when the synchronization accuracy decreases or when high-accurate synchronization is required.

[4] Microchip TimeProvider 4100.
[5] National Instruments OctoClock CDA-2990

Over-the-air Next to distributing synchronization information, the radio elements need to be calibrated to remove the hardware impairments and in particular mismatches between the many RX and TX chains. RadioWeaves increases the technological difficulty to do this because of the high number of antenna elements and the ad-hoc nature of the infrastructure, i.e., there is no imposed design of connections and distribution of resources in space. Proposed over-the-air algorithms in literature, such as in, can be evaluated in this testbed **D3.2**, [12], [13].

2.2.1.5 Data Acquisition System

The DAQ[6] can provide the tiles with a synchronized analog data acquisition channel, able to sample at $1.25\,\mathrm{MS/s}$, with a resolution of 16 bit, and for a total number of 192 differential channels or 384 single-ended channels. This sample frequency and number of bits is sufficient to sample almost every potential sensor, which allows us to move the state-of-the-art in sensor fusion and underpins our aim for multi-modal sensing and positioning research. The high number of synchronized channels allows for setting up truly dispersed architectures, obtaining large and distributed sensor arrays.

The DAQ also has 48 synchronized 16-bit digital-to-analog converter (DAC) output channels, with a sample rate of $3.3\,\mathrm{MS/s}$, able to steer a variety of actuators. For example, an array of speakers, ranging into the deep ultrasonic range, *e.g.*, $45\,\mathrm{kHz}$, can be implemented in research on transmit beamforming and backscattering for fully passive mobile devices. While not imperative for the RW infrastructure, it can enhance positioning accuracy and robustness through other sensor and actuator deployments. Additional features with actuators such as speakers, lamps, vibration elements, etc. could be added to build a test infrastructure for the virtual reality (VR), augmented reality (AR) and smart home automation use cases. The DAQ could be used to complement RF.

The combination of the DAQ and the edge devices form the infrastructure that is required to validate new concepts based on distributed nodes. By using the fully synchronized sensing capabilities of the central data acquisition (including a $24\,\mathrm{TB}$ RAID storage) and processing infrastructure, a baseline performance can be established. This baseline performance can act as a reference for more realistic scenarios where distributed edge devices collaborate in a loosely coupled (and varying) configuration.

2.2.1.6 On-Tile – Sensors, Radios, Processing

Each tile can accommodate a diversity of sensors and actuators, transmitters and receivers and host processing resources. In this section, we elaborate on the PoE board, the RF communication through SDRs, edge processing and automated 3D sampling via a rover.

PoE Board. The standard PoE hardware attached on top (HAT) for the Raspberry Pi, or other derivative forms, are readily available, but do not support the latest PoE version. For this reason, an 802.3bt supported PD circuit was developed. The system is based on the ON Semiconductor NCP1096 which is the PoE-PD interface controller. On top of the 802.3bt support, the board features several connectors and voltages to power different devices and sensors.

Wireless RF Communication through software-defined radio. The testbed hosts a fabric of

[6]The DAQ consists of an 18-Slot PXI Express chassis (NI PXIe-1095), peripheral modules and a controller (NI PXIe-8080 controller). The chassis holds 12 multifunction I/O modules (NI PXIe-6358), a timing and synchronization module (NI PXIe-6672) and a PXI bus extension module (NI PXIe-8394).

distributed SDRs. Each tile is equipped with one USRP NI B210 [14], featuring four RF channels (if in full-duplex mode) with a maximum transmit power of $20\,\mathrm{dBm}$ over a frequency range of $70\,\mathrm{MHz}$ to 6 The supported frequency range covers most licensed and unlicensed bands. The B210 hosts an open and reprogrammable Spartan6 XC6SLX150 field-programmable gate array (FPGA). The B210 is connected to the RPi 4 over USB 3.0. The internal reference clock is used to generate all data clocks, sample clocks and local oscillators. An external $10\,\mathrm{MHz}$ clock and 1PPS can be used for time synchronization between the SDRs. In this manner, it is possible to coherently transmit and receive on all SDRs, when properly calibrated [15] and/or trained [16].

In Techtile, the baseband signals are processed by the RPi 4. Depending on the application, the distributed signals are aggregated to the central server for further processing. By having both local processing and central processing, different techniques regarding local, edge, and cloud processing can be studied.

2.2.1.7 Interface and Data Exchange

The user can interact with the testbed in different ways depending on the complexity of the test scenario. The full stack can be designed with open-source tools provided by the host. A general work-flow is provided by default. The collected channel measurements, *i.e.*, the frequency response at a given carrier frequency, is collected at the central server in the DSS format. These samples can be processed offline to study different RW topologies, federation orchestration, WPT, etc.

In case dynamic real-time channel measurements are required, the in-phase and quadrature (IQ) samples can be processed at one of the three levels, *i.e.*, locally at the FPGA of each CSP, at the local host (RPi 4) or at the central server.

A guideline to access the testbed and to retrieve the data will be documented at `github.com/techtile-by-dramco` together with all the open-source designed tools. In this manner, both the consortium and external partners have a well-structured document to control the testbed. The produced data and results will also be included for reproducibility, adoption, and verification.

2.2.2 LuLIS: Lund University LIS Testbed

Lund University is currently building on the RW concept to develop a real-time, modular, and scalable testbed for fully-digital large intelligent surface (LIS) operations. The LuLIS testbed consists of panels of 16-antenna arrays with localized processing (both analogue and digital) and data storage capabilities. The panels are distributed in the environment, as illustrated in Figure 2.4a, and together operate coherently with a bandwidth of $100\,\mathrm{MHz}$ in the $3.8\,\mathrm{GHz}$ band. Sixteen of such panels will be deployed at the first phase, with the potential to upscale to 64 panels in the second phase. Each panel is supported by an AMD/Xilinx Radio Frequency System-on-Chip (RFSoC) board ZCU-216, with direct RF sampling of 16 channels. An overview of the internal architecture of the panels is shown in Figure 2.4b. The goal of this design is to further explore what can be achieved by exploiting spatial domain even further than what is done in traditional mMIMO. Based on RFSoC technology, the testbed is software controlled and supports real-time distributed processing, making it capable of both exploring many different service types, such as communication and positioning and also performing propagation measurements for the purpose of channel modelling, sensing, and creation of datasets for machine-learning experiments.

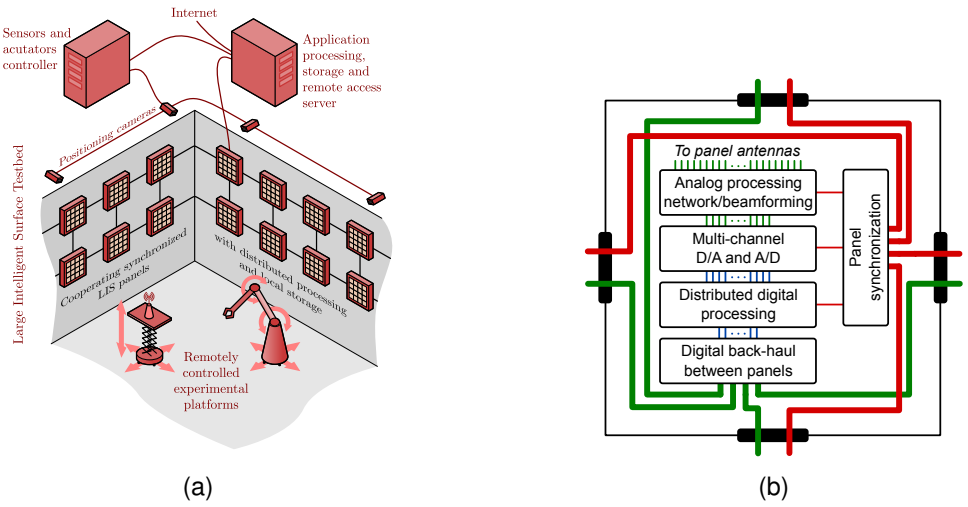


Figure 2.4: (a) Deployment example of the LuLIS testbed under development at Lund University. Cooperating synchronized panels, based on AMD/Xilinx RFSoC technology. (b) Overview of the internal architecture of the LIS testbed panels.

2.2.2.1 Mini D-MIMO using RFSoC 2×2

Before implementing the large-scale testbed, we started with a small-scale system using the Xilinx RFSoC 2×2 development board [17] to gain experience. The state of charge (SoC) features 2 RF 12-bit Analog-to-digital converters (ADCs) with a maximum sampling rate of 4.096GS/s and 2 RF 14-bit DACs with a maximum sampling rate of 6.554GS/s. The SoC integrates Zynq UltraScale+ FPGA including 64-bit quad-core Arm Cortex-A53 and dual-core Arm Cortex-R5F based processing system, allowing the use of PNYQ framework [18] for easier usage of real-time processing accelerators on FPGA, faster system verification, and low-latency system control and reconfiguration.

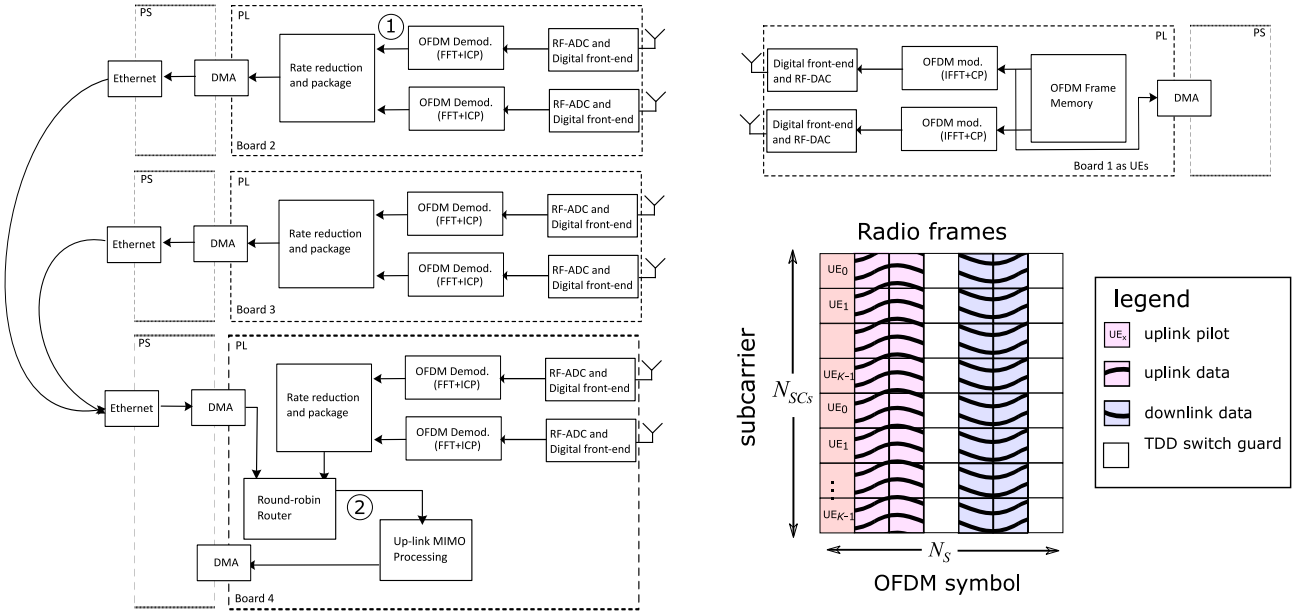


Figure 2.5: High-level block diagram of the mini-D-MIMO system.

System setup and parameters

Figure 2.5 shows the block diagram of the mini 6×2 D-MIMO system using 4 RFSoC 2×2 boards.

One of the RFSoC boards (board 1) is used to implement 2 single-antenna UEs, which are transmitting the orthogonal frequency-division multiplexing (OFDM) signals at the same time-frequency resource according to the frame structure depicted in the figure. A radio frame consists of 7 OFDM symbols with the first one being the uplink pilots for channel estimation. The 2 UEs transmit pilots that are orthogonal in the frequency domain (e.g., for a particular sub-carrier, only one UE transmits pilot whiles others remain silent). The uplink pilot OFDM symbol is followed by 2 uplink data, 1 TDD-switch guard symbol, 2 downlink data, and another TDD-switch guard symbol at the end of the radio frame. The other 3 RFSoC 2×2 boards implement a 6-antenna BS. Each antenna at the BS is followed by RF-ADCs, digital front-end (digital mixer, down-sampler, and filters) and OFDM processing. The OFDM outputs (① in the figure) are transferred via the interconnection system (using the 1G Ethernet on board) to board 4. The aggregated data (② in the figure) are then sent to the MIMO processing block for multi-user MIMO detection. The detection results (constellation points) are then sent via Direct Memory Access (DMA) to the Processing System (PS) to be visualised in PYNQ. With the main focus on the implementation of signal processing and data movement, cables are used to perform multi-board frequency synchronization and UE-BS timing synchronisation.

Current implementation status

The subsystems in Figure 2.5, i.e., OFDM and digital-front-end subsystem, interconnection subsystem, and multiple-input multiple-output (MIMO) processing subsystem, have been implemented and verified. Now the development of the mini-D-MIMO system is entering the system integration phase.

OFDM subsystem: The OFDM subsystem implements OFDM modulation and demodulation, up/down sampling, filtering, and the configuration of RF data converters. The functionality (mainly the data-path) of the OFDM subsystem is verified by loop-back test as shown in Figure 2.6.

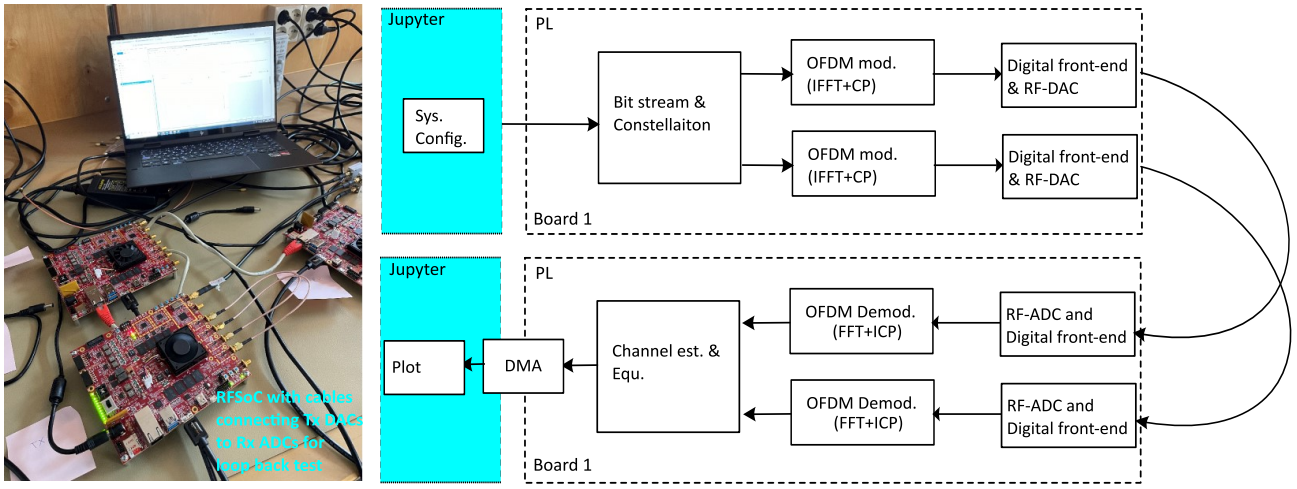


Figure 2.6: Setup of the loop-back test to verify the OFDM system

Interconnection subsystem: The board-to-board interconnection is made via the Ethernet ports connected directly to the PS on the RFSoC board which employs the *socket* python class for the data transmission. Figure 2.7 shows the detailed block diagram. The **Rate Reduction** block is implemented since the PS-based (software) Ethernet pipeline can't support real-time transmission. Thereby, interconnection subsystems (e.g., 100G Ethernet) with dedicated hardware implementations are more suitable for the large-scale real-time D-MIMO testbed. The interconnection subsystem has been tested by using a counter as the source data to verify the correctness of the data movement between RFSoC boards.

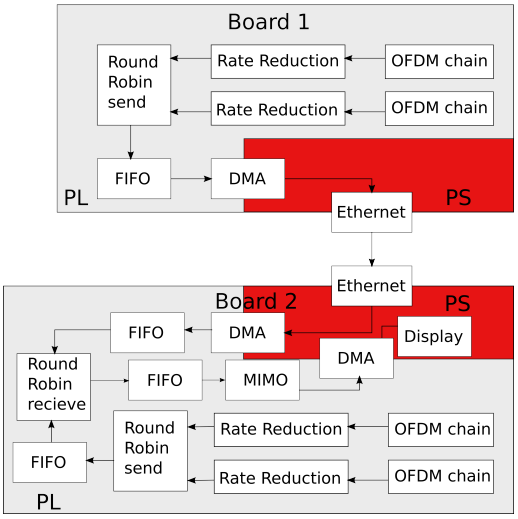


Figure 2.7: Detailed block diagram of the interconnection subsystem via 1G Ethernet.

MIMO processing subsystem: includes channel estimation, channel matrix pre-processing, and multi-user detection. In this implementation, a ZF algorithm is implemented for inter-user interference cancellation. For fast prototyping, easy debugging, and parameterized design, high-level synthesis is used to implement the MIMO processing subsystem, which is verified and tested using the PYNQ framework.

2.2.3 KULMaMi

In this section, a brief overview of the KU Leuven testbed, *i.e.*, KULMaMi, is presented. The content of this section is partially adopted[7] from [19].

The testbed hardware is provided by National Instruments and is centred on the NI USRP-2942R SDR. The BS has 64 antennas achieved by 32 USRPs, each with 2 RF chains, controlled by a CPU in the main chassis. The USRPs are connected to the CPU through switches and optical cables. A timing module provides synchronisation using a 10 MHz REF signal and a PPS signal, distributed through OctoClocks. Three FPGAs help the CPU process UL pilots, estimate CSI, combine signals, precode transmitted signals, and perform bit processing of transmitted data. All devices are mounted on two server racks, making the testbed semi-mobile. See Table 2.2 for an overview and Figure 2.8 for a picture of the assembled BS.

Table 2.2: An overview of the used hardware in the BS of the testbed

Device	Function	Amount
NI USRP-2942R	Generating the wireless signals	32
NI PXIe-1085	Chassis: contains all modules of the CPU	1
NI PXIe-8135	Host controller chassis module: CPU	1
NI PXIe-7976R	FPGA chassis module: MIMO processor	2
NI PXIe-7976R	FPGA chassis module: Bit processor	1
NI PXIe-6674T	Timing chassis module: Generates REF and PPS	1
NI PXIe-8384	PXIe bus extension chassis module	4
NI CPS-8910	PXIe switch	4
NI OctoClock	Clock distribution	5

[7] With the approval of the authors.

Figure 2.8: The KU Leuven massive MIMO testbed. The base station is fitted with 64 patch antennas, which can be configured into multiple types of antenna arrays. The figure shows two rectangular arrays of 32 antennas. Note, that other antenna topologies can be constructed.

Table 2.3: Key parameters of the mMIMO testbed during the measurements.

Parameter	**Value**
FFT size	2048
CP length	144/160
Subcarriers used	1200
Subcarrier spacing	15 kHz
Sample clock	30.72 MHz
Effective Bandwidth	18 MHz
Frame length	10 ms
Slot length	0.5 ms
Symbols per slot	7

The KULMaMi mMIMO testbed has a unique feature in its flexible base station antenna topology. The antenna deployment can be easily altered, thanks to the use of long coax cables between the USRPs and the antennas, together with flexible mounting brackets for the antennas. This feature allows for distributed antenna configurations and enables measurements with various antenna topologies.

The antennas used in the testbed are connected to the base station using $8\,\mathrm{m}$ long coaxial cables. Currently, two sets of in-house designed patch antennas are available for use, optimised for the $868\,\mathrm{MHz}$ and $2.4\,\mathrm{GHz}$ to $3.5\,\mathrm{GHz}$ bands, respectively.

The hardware provided is driven by the NI MIMO application framework, a LabView project that implements communication between the BS and UE. Researchers can customise the code in the LabView project. The framework uses LTE-based TDD and OFDM modulation with 1200 subcarriers and a subcarrier spacing of $15\,\mathrm{kHz}$, resulting in a bandwidth of $18\,\mathrm{MHz}$. The frame structure repeats every $10\,\mathrm{ms}$ and is divided into two subframes with ten slots each, where each slot consists of seven OFDM symbols. Five types of OFDM symbols are used for different purposes: PSS, DL pilot, DL data, UL pilot, and UL data. The symbols are scheduled in time, and the first slot sets up the communication link, transmitting a PSS symbol, two DL pilots, and an UL pilot before the remaining slots transmit DL pilots and data, followed by UL pilots and data.

Automated measurement system developed using CNC-enabled positioners to mount antennas, allowing for automatic user placement. Four OpenBuilds ACRO 1515 positioners were used, providing a reach of $1250\,\text{mm}$ by $1250\,\text{mm}$ with an accuracy of better than $0.1\,\text{mm}$. The movements of the positioners are driven by stepper motors controlled by an Arduino-compatible microcontroller running GRBL. Turrets were also mounted to change the orientation of the antennas, accurately controlled using high-quality servo motors and a microcontroller. To trigger the BS to capture a channel state information (CSI) sample, a Transmission Control Protocol (TCP) packet is sent to the BS on port 5022. Finally, a central controller in Python coordinates the movements of the positioners and turrets and triggers the BS to capture a CSI sample.

2.2.4 LUMaMi

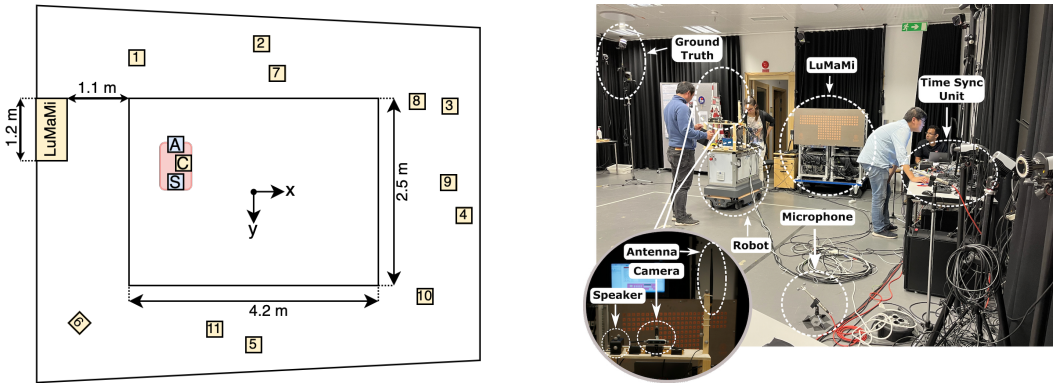
The LUMaMi testbed [2] is a mMIMO testbed with a centralized 100-antenna array. It performs real-time OFDM processing with 50 SDRs from NI, each with two TX and RX chains and Kintex-7 FPGAs and centralized and mMIMO with four additional Kintex-7 FPGAs. The frequency range of the SDRs is $1.2\,\text{GHz}$ to $6\,\text{GHz}$, and the current setup is tailored for $3.7\,\text{GHz}$ with a $20\,\text{MHz}$ bandwidth. With the 54 FPGAs and the PCIe high-speed interface, LuMaMi is capable of real-time operation of a 30.72 MS/s LTE-like air interface in TDD mode. Due to the TDD-operation and reliance on reciprocity for channel knowledge and beamforming, the testbed also contains the necessary functionality for reciprocity calibration [20].

LUMaMi has been used extensively in several different roles, testing efficiency of spatial multiplexing [21], testing real-time processing in high mobility environments [22] (shown in Fig. 2.9), and exploring mMIMO technology for new services in the millimeter wave (mmWave) $28\,\text{GHz}$ band [23]. It has also been used as a robust low-latency wireless link in cloud control of robots [24], a channel sounder in industrial environments [25], and when collecting audio/video/radio propagation data-sets for machine-learning based positioning services (as illustrated in Fig. 2.10) [26].

2.2.5 RUSK

(a) (b)

Figure 2.9: (a) LUMaMi on the roof top facing the parking lot for mobility test. (b) UE antennas mounted on top of the car roof.



(a) (b)

Figure 2.10: (a) Top-down view of the measurement setup: red rectangle is the MiR200 robot used for the measurement campaign, which is equipped with antenna (“A”), camera (“C”), and speaker (“S”) respectively. The placement of the microphones is illustrated as numbered squares and LUMaMi is located on the left-top corner of the room (b) A snapshot of the measurement campaign.

The RUSK LUND channel sounder (Figure 2.11) was designed by Medav GmbH, Germany and provides exceptional capabilities for a complete characterization of MIMO propagation channels. It comprises a high-speed antenna switch in both the TX and RX units and uses the switched array principle to sequentially rotate through all pairs of TX and RX antennas. The RF-switches support up to 32 antennas on both units, resulting in up to 32×32 link combinations

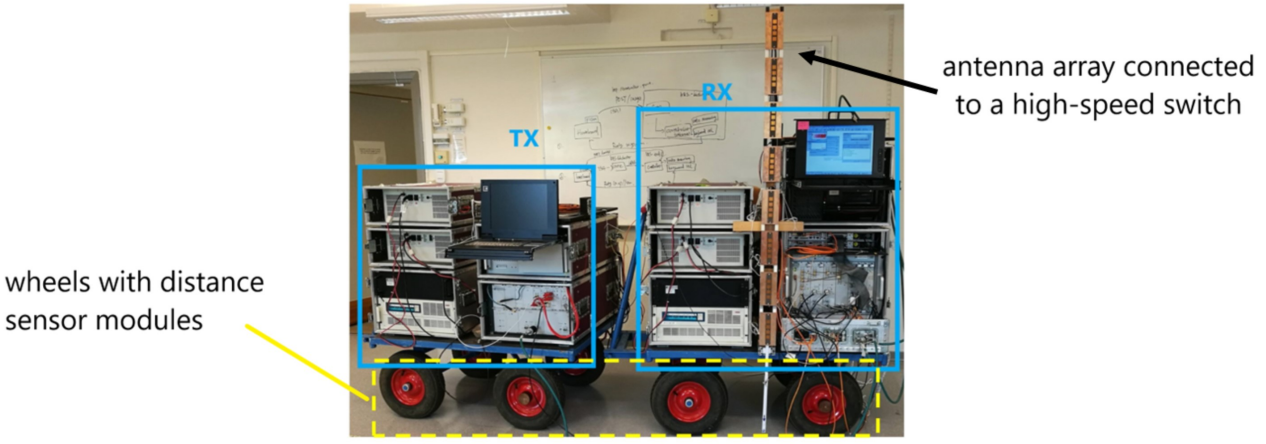


Figure 2.11: The RUSK LUND wideband switched array channel sounder. The TX and RX units are displayed on the left and right sides, respectively.

The RUSK channel sounder is based on periodic multicarrier spread spectrum test signals defined in frequency domain and real-time sampling at the receiver. The test signal is very similar to an OFDM signal, except that the phases of individual subcarrier are fixed and optimized to ensure a constant power spectral density over the whole bandwidth and to minimize the peak-to-average power ratio (PAPR) (crest factor) of the resulting time domain test signal. The low PAPR leads to low distortion in the amplifiers and modulator circuits when channel sounding is done simultaneously at different frequencies. The RUSK sounder supports measurement in 300 MHz, 2 GHz, and 5 GHz frequency bands with a bandwidth up to 240 MHz.

The RUSK sounder is capable of investigating the time-variant characteristics of dynamic MIMO channels. Specifically, the high repetition rate of the measurements ensures that the time-variant

channel is sampled fast enough to capture the fast fading statistics.

The frequency synchronization of the TX and the RX units is realized by using a high precision 10 MHZ Rubidium (Rb) frequency reference at both sides. Before each measurement campaign, the two Rb references need to be connected via an optical fibre cable or a BNC cable. After being synchronized for a significant amount of time, the synchronization can maintain for a few hours after cable disconnection. If the TX and RX separation distance is small, e.g., less than 10 m, a more reliable option is to maintain cable connection during the measurement.

For data evaluation, it is often needed to record the ground truth information of the measurement setup, e.g., distance and coordinate. which means that the simultaneous recording the external sensor data together with the frequency response measurements is of great importance. For this purpose, the RUSK sounder supports various external data sources e.g., distance, compass, etc. In the current setup, a distance sensor module comprising two individual wheel sensors is installed at both the TX and RX units. The 1PPS signal from the wheel sensor can be used to simultaneously trigger the recording of the distance measurement and the channel measurements.

The sounder has been used for several measurements in the past, e.g. [27], and in the REINDEER project [28], [29].

2.2.6 LUDyna - LUnd DYNAmic multi-link channel sounder

The description of LUDyna (LUnd DYNmic multi-link channel sounder) in this section is adopted from [28], [29] where the results of a measurement campaign are presented.

To investigate the temporal non-stationarity occurring in dynamic scenarios, a new distributed multi-link channel sounder to capture the dynamics of the channel has been implemented. The new channel measurement system is built around the NI USRP hardware. More specifically, the NI-USRP 2953r platform. The USRP consists of two RF boards with an analogue bandwidth of 40 MHz and a tuneable carrier frequency ranging from 1.2 GHz to 6 GHz. A *radio* refers to *one* of these boards – i.e., one USRP consists of two radios. To achieve synchronisation between the SDRs four Rb-clocks are used [30]. Each USRP connects to an external Rb oscillator, which provides a 10 MHz reference frequency and a 1PPS output signal to align the snapshots in the distributed system. Before each measurement campaign, one of the Rb-oscillators is selected as the primary and is used to synchronize the clocks on the other Rb-units.

The NI LabVIEW software framework is used on the host computers to configure the radios and to acquire and store the data. The system is based on a time-division multiple-access (TDMA) scheme that assigns all participant radios a unique time slot to transmit. During that slot, all other radios in the setup receive and store the received complex-valued samples. The hardware for the multi-link system is summarized in Table 2.4.

Table 2.4: Hardware for the multi-link measurement system.

Hardware	Description
7 NI-USRP 2953r 40 MHz	The radios with RF heads
3 SRS FS725	10 MHz frequency standard
1 SRS FS740	10 MHz frequency standard with GNSS
7 Host computers	Radio control and logging data

The system can easily scale; there is nothing in the implementation limiting (within reason) the number of SDRs used in a given measurement setup. As long as the SDRs can be synchronised

and share the same notion of time the only limiting factors are the channel characteristics one wishes to measure. This is due to the TDMA scheme where the measurement time will increase linearly with the amount of radios.

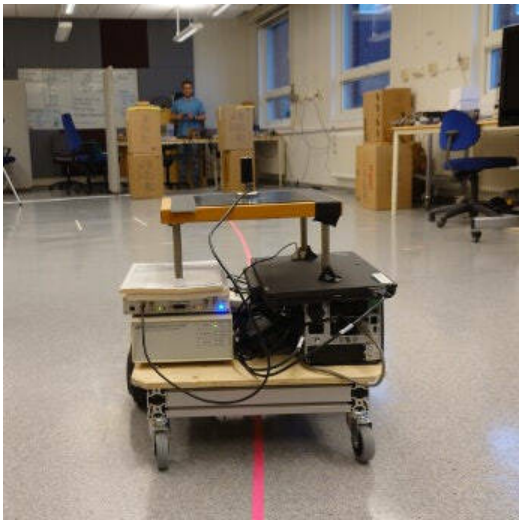


Figure 2.12: Example of a radio setup of the Lund Dynamic Multi-link Channel Sounder. The mobile radio from the measurement presented in [28].

A distributed MIMO system in a dynamic scenario requires a rapid change of the receiver gain due to the spatial distribution of radios. Since this change of gain is in the sub-microsecond range, an automatic gain controller (AGC) was implemented on the FPGA to try to use the full range of the ADC. The system can transmit any arbitrary waveform that can be loaded from a file and then transferred to the memory of the FPGA for transmission replay.

2.2.6.1 Interface and data exchange

The system measures the in-phase (I) and quadrature (Q) components and streams them to a NI proprietary file format. The files are later processed in Python to conform with the DSS format.

2.2.7 LARVA - LARge Virtual Array

The measurement system for large virtual arrays, termed LARVA, was developed at TU Graz to enable measurement of synthetic arrays with a very large spatial aperture, termed XL-MIMO, large synthetic arrays (LSAs) or physically large arrays (PLAs) in literature, as well as LIS or reflective intelligent surface (RIS) in the wider sense. It consists of a mechanical positioner, with the channel measurements performed either using a correlative channel sounder (CCS) performing time-domain measurements or using a vector network analyzer (VNA) and performing measurements in the frequency domain. While RW is expected to cover only a relatively small bandwidth with an envisioned system bandwidth of up to 100 MHz (available at different centre frequencies), ultrawideband (UWB) measurements give valuable insight due to the high time resolution, especially when paired with the flexibility and large spatial resolution of the LARVA measurement system. The system is not real-time capable in the conventional sense, but pseudo-real-time capable in post-processing, i.e., all RF measurements are fully coherent. An outline of the included hardware is given in Table 2.5, the description of the measurement system can also be found in D1.2 [28].

Table 2.5: Hardware components of the LARVA measurement system developed at TU Graz, consisting of two mechanical positioners.

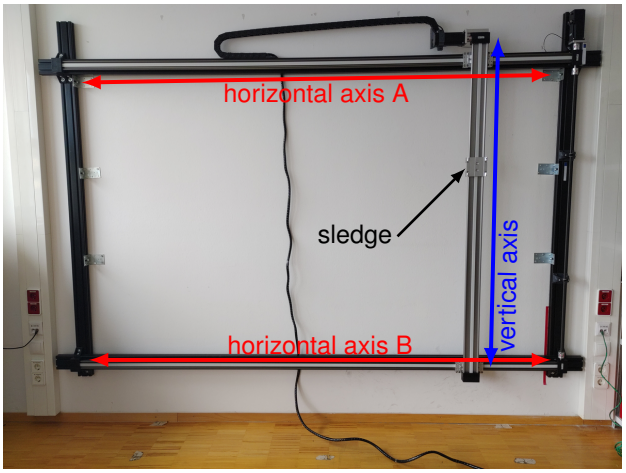
qty.	Hardware	Description
1	R&S® ZVA24	radio measurement device, DC to $24\,\text{GHz}$
	Ilmsens Channel Sounder	radio measurement device, approx. $0-7\,\text{GHz}$ or $3-10\,\text{GHz}$
4	dryve® D1	motor controller
4	drylin® NEMA24	high resolution stepper motors with holding break
1	host computer/laptop with Matlab	Device control data logging via USB or Ethernet

2.2.7.1 Mechanical positioning devices

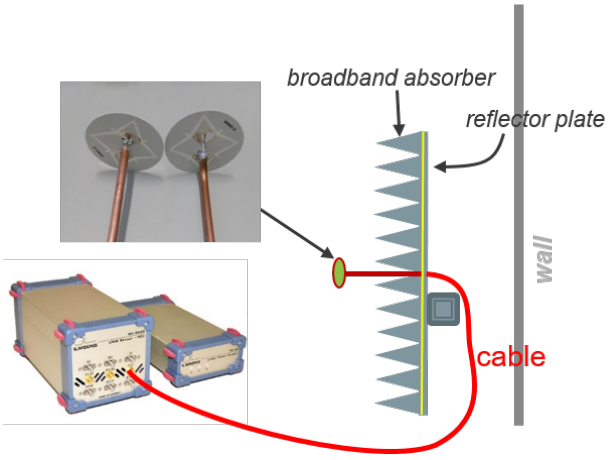
The use of mechanical positioners enables flexible measurements of different array geometries in an automated manner, with accurate relative positions between measurement points. This allows to form virtual arrays covering a large area, e.g., exhibiting a large aperture.

Two structurally identical mechanical positioners were constructed, with the dimensions of the employed motion axis allowing to measure roughly 2.5 m $\times$ 1.5 m, shown in Figure 2.13a. The vertical motion axis and the horizontal axis A are equipped with *dryve® D1* motor controller driving *drylin® NEMA24* stepper motors, allowing accurate positioning in the sub-millimeter range with high repeatability. The horizontal axis B is connected to axis A via a spindle shaft that reduces the load on the motor and allows smooth motion of the vertical axis. The stepper motors feature a holding break to keep the position during stop intervals, and allow motion with variable acceleration, deceleration, and velocity. The vertical is equipped with the sledge on which the antennas (or other measurement devices) are mounted.

To emulate a RW CSP mounted directly on a surface, each mechanical positioner is equipped with a broadband absorber and a reflector plate on the sledge behind the antenna, effectively removing the surface reflection on which the positioner is mounted on. The suppression of the back reflection results in realistic measurements of a CSP equipped with an array mounted directly on an environment surface as described [31]. Due to this construction, the distance from the antenna phase centre to the back wall is roughly 43 cm (including the antenna length of 10 cm), i.e., the corresponding RW CSP is not mounted directly on the wall but as close as possible.



(a) wall mounted mechanical positioner



(b) antenna, absorber and measurement device

(c) antenna and absorber

(d) slot antennas

(e) dipole antennas

Figure 2.13: Measurement setup showing one mechanical positioner mounted on the wall (a) and the corresponding setup consisting of the measurement device, antenna, cables, and absorber on reflection plate in (b) and (c). The patch antennas are shown in (d), the dipole antennas in (e).

Depending on how the mechanical positioners are mounted in the environment, the effectively usable measurement area can be smaller due to requirements for safety margins accounting for the size of the absorber and reflector plate.

2.2.7.2 Radio measurement devices

Two different devices are available to perform the radio measurements: a VNA *ZVA24* and an Ilmsens correlative channel sounder (ICCS), where the former performs measurements in the frequency-domain and the latter in the time-domain. The VNA covers a frequency range of 24 GHz (from DC) and can be calibrated with a corresponding calibration kit provided with the device. The calibration is a standard through-open-short-match (TOSM) calibration performed for all required measured ports. The ICCS covers a frequency band of $f = 3.8 - 10.2$ GHz, centered at a carrier frequency of $f_c = 6.95$ GHz, and allows fully coherent measurements of two receive channels with respect to one transmit channel. To remove the effects of the system response, e.g., cables and internal hardware and cross talk between the two receive channels, a calibration procedure comparable to a TOSM calibration is performed before each measurement. Independent of the used measurement device, all measurements are fully coherent. The signal-to-noise

ratio (SNR) can be improved by additional averaging of channel responses, which will decrease the measurement speed.

2.2.7.3 UWB antennas

The antennas that were used in the measurements are dipole antennas and dipole-slot antennas, see Figure 2.13d and Figure 2.13e, with the latter being termed slot antenna for better differentiation. The dipole-slot antennas were manufactured according to the cross exponentially tapered slot (XETS)-antenna design from [32], which perform well in the frequency band of interest that selected as $3 - 10$ GHz. While the VNA covers a much wider frequency region, measurements are only performed in the frequency band for which the antennas are designed. As the measurement aperture will allow reception from azimuth and elevation, the slot antennas' design were chosen as it exhibits an approximately omnidirectional pattern. When mounted on the absorber and reflector plate construction the antennas only receive from the half plane facing away from the absorber, i.e., only in the half plane facing away from the surface the CSP would be mounted on.

The dipole antennas are made from 2-cent coins and exhibit an omnidirectional pattern in the horizontal plane with increasing attenuation towards nadir and zenith, being circularly polarized. The cent-antennas were already used in measurements [33], [34] and are known to work well.

The slot antennas were manufactured at TU Graz based on an antenna design proposed in literature [32], termed XETS antennas. Measurements of the gain patterns of the XETS antenna mounted on the reflection plate with the absorber were performed by TU Vienna in an antenna measurement chamber. The measurements show that the resulting attenuation towards the back, e.g., towards the floor or the wall the mechanical positioner will be mounted on, should be sufficient to remove unwanted close reflections due to mounting in the environment. The slot antennas are vertically polarized, which needs to be considered manually during the assembly.

2.2.7.4 Interface and data exchange

All devices can be controlled from Matlab, with control commands and data transferred via USB when using the ICCS, or via Ethernet when using the VNA. The mechanical positioners are natively controlled via Python but are also interfaced from Matlab. Positions for the positioner to move to are given in local Cartesian coordinates, with conversion to the global environment coordinate system necessary to be taken care of manually by measuring the orientation of the measurement area of each mechanical positioner in the global coordinate system.

The VNA directly measures frequency domain transfer functions with variable bandwidth, frequency spacing, frequency region, and transmit power (to name the parameters of main interest). These parameters dictate the measurement rate as well as the number of complex-valued samples of the channel transfer function available for storage.

A selection of representative measurements will be provided for public access, collected in various environments and channel conditions, e.g., line-of-sight (LoS), non-line-of-sight (NLoS). For the measurement environments, all antenna positions will be provided, in addition to an environment floor plan to be plotted using Matlab.

2.3 RadioWeaves Devices

In REINDEER, two EN device platforms are developed. One is designed by NXP and operates close to the 2.4 GHz frequency. The main scope of this board is to study the received power attained at a certain position. In addition, it supports on–off keying (OOK) modulation at a maximum rate of 6 MHz. The other board is designed by KU Leuven, extending the previous design by allowing other frequencies to be used and including a maximum power point tracking (MPPT) to increase the power harvesting efficiency. Furthermore, the device is fully powered through RF, in contrast to the NXP board, which is powered externally, thereby better mimicking a practical energy-neutral device. To be complete, several conventional methods to emulate devices will also be used, such as SDRs or signal generators.

2.3.1 NXP END - Energy harvester evaluation board

In contrary to Section 2.3.2, where a commercial harvesting IC is used (AEM40940), NXP is working on a custom-specific energy harvesting IC utilizing their in-house intellectual properties (IPs) for design and manufacturing. Engineering efforts are accounted to WP4, however, in terms of completeness, capabilities in regard to demonstrations are outlined in this document. The IC development is in an early stage; however, key components like the harvesting unit (the charge pump) and connected backscatter modulator are already fabricated on a multi-project wafer (MPW) and available for internal testing. Currently, the IC comes with a limitation of powering the internal level shifters and input/output multiplexers by an external source. However, this is only a limitation for the development of a standalone class 1 EN device, but not for the characterisation and performance evaluation of the harvesting unit subject to frequency diversity (for example 900 MHz versus 2.4 GHz) and harvesting conditions (mMIMO versus conventional point-to-point radio links).

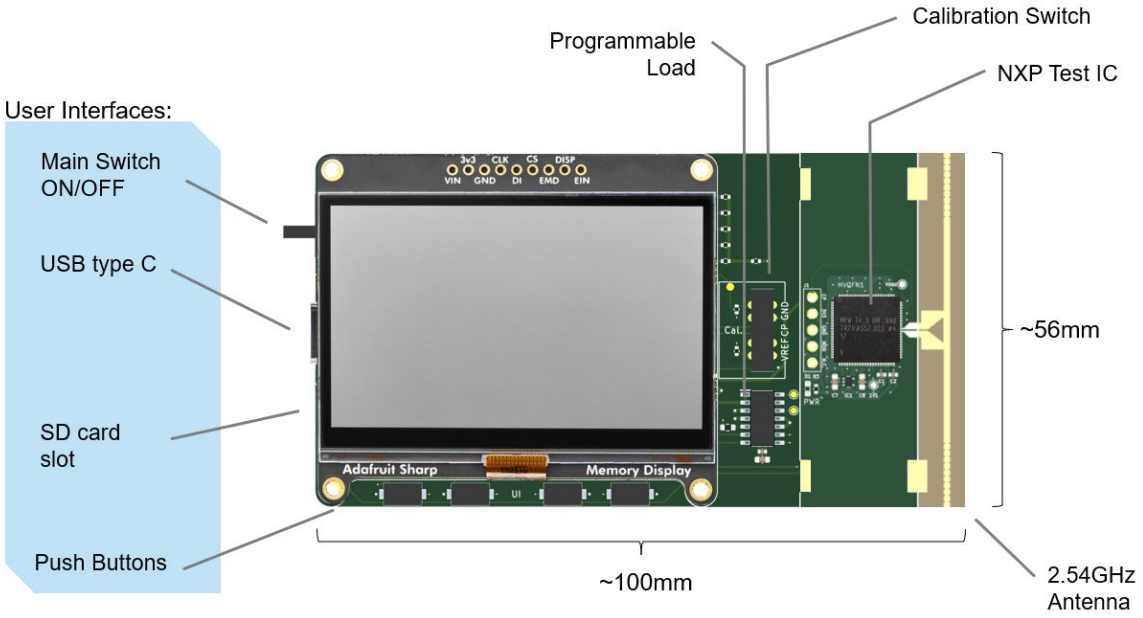


Figure 2.14: Energy harvesting demonstrator design

Figure 2.14 shows the demonstrator design. The core element is the NXP harvesting test IC, which is directly matched to a 2.54 GHz antenna. The antenna features some tuning stubs capabilities, optionally available to tune the resonance frequency for optimum harvesting performance.

Furthermore, the device is equipped with a calibration circuit, necessary to calibrate the measurement amplifier connected to the charge pump output. The programmable load is used to load the charge pump depending on the environmental conditions and thereby limits the actual output voltage the charge pump can handle. Finally, a display is used to form a hand-held device showing the actual status information and provides user interactions without the need to be connected to a PC. However, the device features USB and SD-Card support for transferring and storing of measurement data. We plan to mount the device as a storage device, holding the measurements. Both manual and threshold-based (RF power) triggering is planned to be implemented. A manual will be provided in order to conduct experiments.

Backscatter modulation is performed by means of the main board MCU (which also controls the user interfaces and the display). The NXP test IC just covers the modulation transistor, located internally in parallel to the antenna terminals. It allows short-circuiting the antenna terminals to produce an on/off keying type of modulation only. The sequence of how the modulator is controlled is very flexible and can reach up to 6 MHz data rate. As such, the user is intended to write the accompanying software to perform the modulation.

2.3.2 KUL END - Energy harvester with back-scatter functionality

An initial integration of an EN device is proposed by KU Leuven and must meet the minimum requirements to support class-1 use cases from D1.1. The EN device architecture is presented in Figure 2.15. The electromagnetic waves are collected by the receive antenna, and a rectifier together with a matching network performs RF to DC conversion. An optional MPPT efficiently fills the energy buffer. The energy from the buffer can be used to power the low-power microcontroller unit (MCU). The MCU will, in turn, be connected to the demodulator and modulator to enable the downlink and backscatter functionality, respectively. The connected antenna is terminated with a varying impedance to accomplish backscatter uplink communication. A specific use case can be supported by the connection of additional sensors and actuators to the MCU.

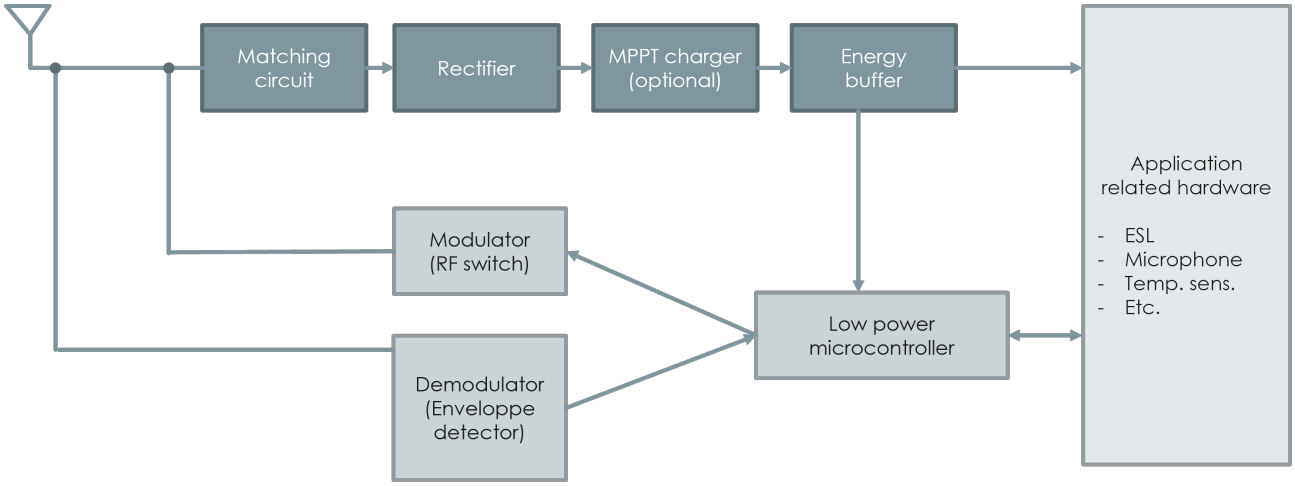


Figure 2.15: Architecture EN device.

A complete implementation of our EN device is represented in Figure 2.16 involving all required building blocks. To validate experiments from Chapter 4, the EN device should support several features. KU Leuven proposes a first version of a Class 1 EN devices that is fully powered by RF energy and is able to provide energy to the onboard electronic components. Therefore, at least, an energy harvester circuit combined with a processing unit and a bidirectional communication

system is crucial to perform the experiments. Back-scatter communication is enabled by two separated circuits acting as a modulator and an amplitude-shift keying (ASK) demodulator.

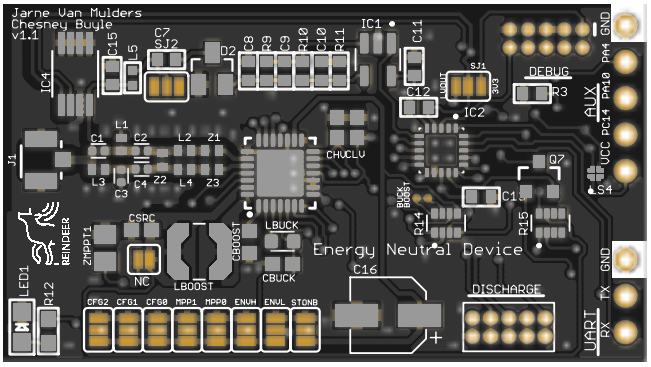


Figure 2.16: Complete implementation of an EN device.

The documentation, concerning the specifications, specific hardware components, schematics and firmware examples are available in the github repository [35].

The communication functionality can support both uplink and downlink communication, making this platform generally deployable to be extended with sensors or actuators, to support specific use cases. High communication rates and low latency levels cannot be achieved with this hardware.

On both infrastructure and EN device side, a debug connector and interfaces are available. The EN device has an universal asynchronous receiver-transmitter (UART) interface and debug connector to follow up the running code in the device itself. At the infrastructure side, the uplink communication happens via backscattered ASK signals and can be demodulated by an SDR that converts the backscatterd RF signals again into baseband signals. Similarly, ASK signals can be sent out via the SDR and demodulated at the EN device.

2.4 Software

2.4.1 Localisation Engine

The positioning engine is based on a particle filter to compute the most likely location of a mobile node considering current and past information received. A particle filter provides a flexible framework in which sensor models and a movement model can be plugged in. For each type of measurement (e.g., received signal strength indicator (RSSI), time-difference-of-arrival (TDOA), and camera image), a sensor model describes the likelihood of observing a measurement given an assumed position (and typically also assumed other properties that may influence the likelihood functions, e.g. antenna attenuation. The movement model describes the likely movements of a mobile node, possibly also directly informed by measurements from the inertial sensor. A particle filter iteratively processes sensor data to update its position estimate. A number of particles represent a probability distribution as a weighted sample. The movement model is used to update the particle state to new possible positions. Subsequently, the sensor models are used to update the weight of each particle using the likelihood contributions of each sensor.

The cloud infrastructure provides an interactive floor plan editor which can be used to define the scene information used by the motion model, but also the location of fixed sensors or other

features that provide the geometric information to interpret sensor data. The floor plan is further processed and optimised for use by the positioning engine efficiently.

Chapter 3

RadioWeaves Simulators

3.1 Introduction

In addition to the testbeds, two simulators are under development, *i.e.*, TugSim and LuSim. They will be used to extend the experiments conducted in the testbeds to more complex and more varying scenarios. In addition, both the testbeds and simulators will be jointly utilized to perform *hardware in the loop* simulations, where part of the experiments are done in hardware, while others are emulated in simulations.

Both simulators are complementary to each other. The TugSim has two main purposes: i) generation of synthetic received signals for positioning and ii) geometry-based prediction of channel vectors to evaluate beamforming methods and compute the WPT efficiency. In contrast, LuSim focuses on 'real-time' interactive participation and use-case validation. It aims to provide an interactive end-to-end simulator with a user-friendly interface through Unity while separating Unity from logic for improved usability and flexibility. It incorporates cross-layer capabilities to analyse the effects of physical layer changes on the application layer.

The DSS format will be used in the simulators to exchange data, as well as to use it as input to perform *hardware in the loop* simulations.

3.2 Geometric channel simulator

The REINDEER consortium has designed the TugSim simulator that computes wireless propagation channels based on a geometric environment model. The implemented channel model corresponds closely to the model introduced in the REINDEER deliverable D1.2 [28, Chapter 2]. The computation of channels based on geometric environment information can be used for multiple analyses.

3.2.1 Synthetic received signal generation

The initially intended purpose of the simulator was to compute synthetic received signals $\boldsymbol{y}$ (cf. [28]) for a specific system model, i.e., RW panels with various layouts and positions, and site-specific environment model, i.e., a three-dimensional (3D) floorplan of an indoor scenario. Fig. 3.1 shows the schematic building blocks of the TugSim simulator. For the performance eval-

uation of positioning algorithms (cf. [36]), this synthetic received signal generation is of particular interest.

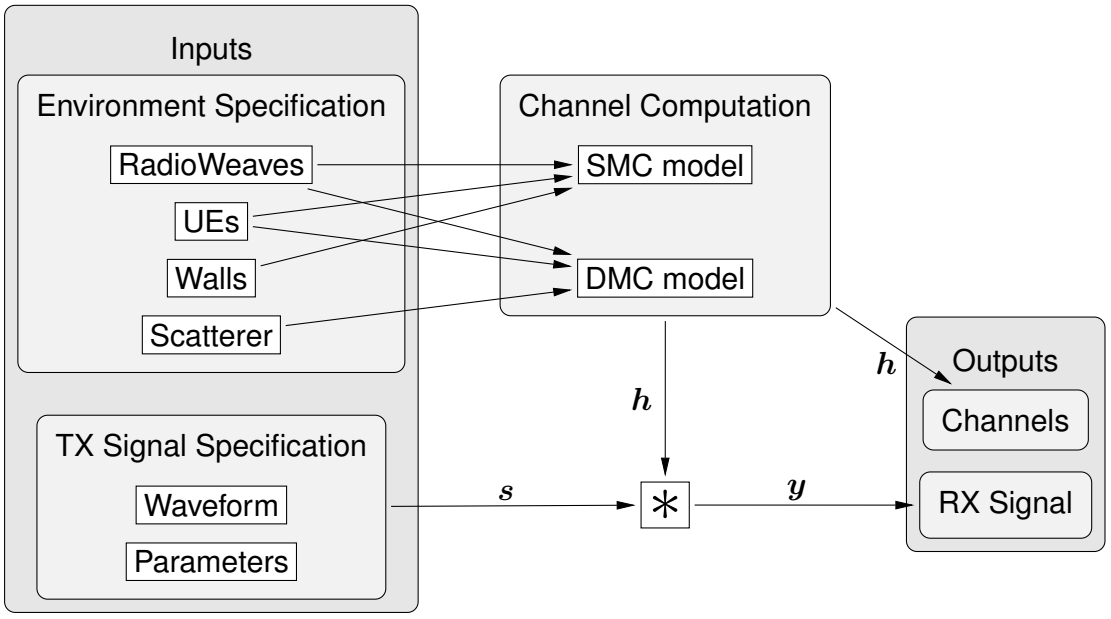


Figure 3.1: Schematic block diagram of the geometric channel simulator TugSim.

Having the ground truth of the parameters underlying the data, i.e., the received signals, produced with a generative model allows evaluating the accuracy of a certain estimator in inferring the parameters. Furthermore, exact knowledge of the parameters can be used to evaluate lower bounds on the estimates against which the estimator performance can be compared to assess its efficiency.

3.2.2 Geometry-based beamforming

We found that simulating channel vectors $\boldsymbol{h}(\boldsymbol{p}, f)$ for multiple-input single-output (MISO) systems is also of interest for geometry-based beamforming. In the context of RadioWeaves, relying on geometry-based channel predictions rather than channels estimated based on a "measured" pilot proved valuable for the initial access to energy neutral (EN) devices. Using the synthetic aperture measurement testbed described in Section 2.2.7, we were able to accurately model channel measurements acquired with a VNA. Both for beam sweeping [37], as well as for environment sensing [38], we have successfully used geometry-based beamformers on synthetic aperture data.

Figure 3.2 shows an example of a geometry-based beamforming evaluation computed with the simulator. We simulated the free-space path gain, linearly proportional to the power density, of a $(2.5\,\mathrm{m} \times 1.5\,\mathrm{m})$ uniform rectangular array (URA) with (40×25) antennas, and MRT chosen as a precoding method. A wall is added which causes a strong specular multipath reflection. The resulting power budget in Figure 3.2 demonstrates how the transmitted power of a *physically large* antenna array is spatially distributed. In this simulation, we assumed isotropic transmit and receive antennas and no polarization losses or parasitic coupling. This simulation demonstrates how multipath propagation improves the power budget compared to a LoS beamformer only. Compare, for instance, the simplified back-of-the-envelope calculations made in the REINDEER

deliverable[1] D2.1 [39]. The computed MISO path gain PG at the EN device location $\boldsymbol{p}_{\mathrm{EN}}$ yields $-23.6\,\mathrm{dB}$.

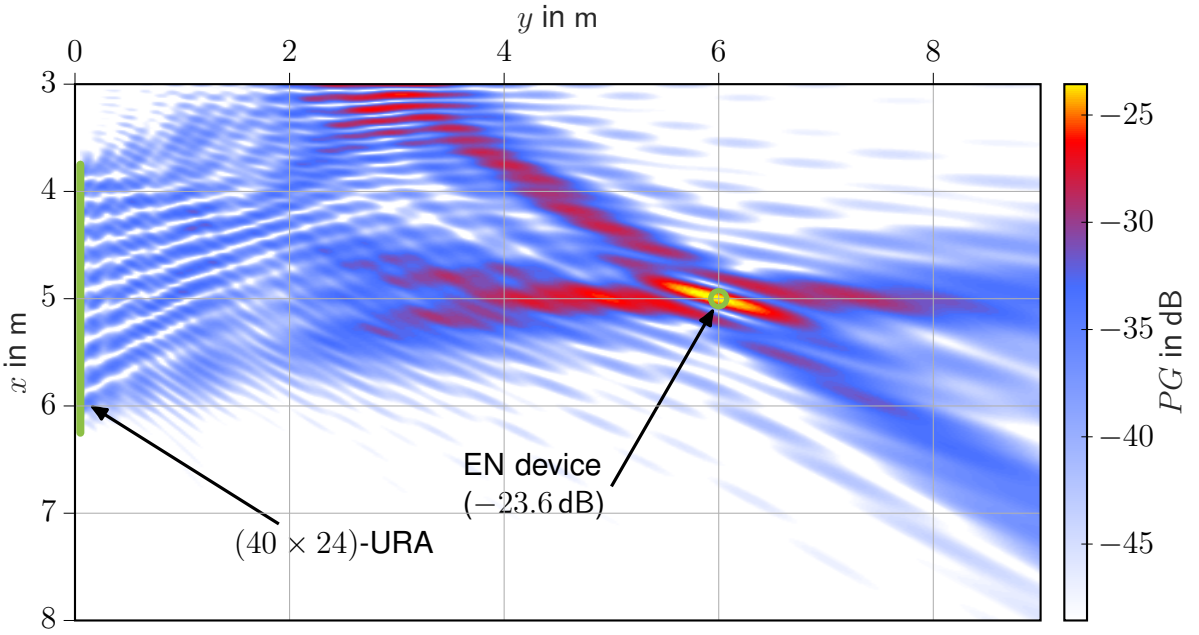


Figure 3.2: MRT applied on our geometric specular multipath component (SMC) channel model from [28]. Depicted are the LoS beam and one SMC beam.

3.2.3 Application examples

The simulator has been used in several different works for different purposes. These were mostly focused on geometry-based beamforming so far. In an early development stage, the simulator has been used to compute power budgets for a simplified indoor scenario and to evaluate the effect of multipath propagation on the efficiency of power transfer [39]. In [40], the simulator has been used to evaluate a geometry-based initial-access scheme for WPT. In the REINDEER deliverable D4.1 [41], simulation results for beamforming have been compared with measured data acquired with a VNA to verify the validity of our implemented channel model. Furthermore, the simulator has been used to compute range-dependent gain patterns in the array near field. In [42], the simulator has been used to compute spatial power distributions when using a "random beamformer" for initial access.

Future applications will focus more on the context of positioning, i.e., for received signal generation as described in Section 3.2.1. These results may be found in the REINDEER deliverable D3.3 [36].

[1] See Table 6.3 on page 47 of REINDEER deliverable D2.1 [39].

3.3 LISim - Real-time system and PHY-level Simulator

3.3.1 Objectives

The objective of the LIS simulator being implemented using Unity as a foundation is to provide an interactive end-to-end simulation environment that leverages physical layer properties while disentangling Unity from logic. The simulator aims to offer a user-friendly interface through Unity, which will primarily focus on ray-casting and GPU acceleration, leveraging Unity's strengths. Scenarios definitions, data presentation, data analysis, dynamic resource orchestration, and scenario dynamics will be externalized to external scripts, such as Python, increasing usability and flexibility.

The simulator extends the first version by including indoor and cell-free scenarios with a massive amount of antennas, facilitating realistic testing of use cases defined in the Reindeer project. It enables the study of dynamic resource allocation based on the environment and application requirements, and facilitates the orchestration of federations. Additionally, the simulator incorporates cross-layer capabilities to model the effects of changes in the physical layer on the application layer. The simulator allows for user interaction and can be used for synthesizing data for machine learning experiments, serving as a channel emulator. Furthermore, the simulator has the potential to allocate and track energy consumption of blocks, allowing for energy-aware simulations and analysis of LIS systems.

The LIS simulator is designed to provide a versatile and powerful platform for researchers and practitioners to investigate various aspects of LIS systems in real time. It leverages physical layer properties to create an interactive end-to-end simulation environment, enabling users to study the behaviour and performance of agents within a LIS system. The simulator offers a user-friendly interface through Unity, primarily focused on ray-casting and GPU acceleration while externalizing logic to external scripts for increased usability and flexibility. It supports batch mode for efficient simulations and allows for user interaction. Future work includes evaluating the framework based on real measurements, continuous improvement of its capabilities to support research, development, and validation of LIS systems, and exploring energy-aware simulations by allocating and tracking energy consumption of blocks within the simulator.

3.3.2 Foundation

The LIS simulator is based on the Sivert Vehicle-to-Vehicle (V2V) Simulator, originally designed for outdoor environments, particularly urban canyons. However, the LIS simulator extends the capabilities of the V2V simulator to focus on indoor environments with complex shapes, where the spawning of Multipath Component (MPC) points is adapted using spawning volumes instead of rules. This allows for more realistic modeling of indoor LIS scenarios with a large number of transmitters and receivers. Additionally, the LIS simulator incorporates more efficient handling of MPC points to support a larger number of Tx and Rx elements, and includes declaration of scenarios to minimize interaction with Unity, ensuring efficient simulations. The channel modeling in the LIS simulator is based on geometry-based stochastic channel modeling, where MPC points are distributed in the environment and used for channel modeling, instead of exhaustive ray tracing. The LIS simulator also includes the ability to connect to NS-3 for more advanced simulations and analysis of channel characteristics.

By leveraging the capabilities of the existing Sivert V2V simulator and building upon its foundation, the LIS simulator is able to provide a robust and flexible simulation environment for studying

LIS systems in indoor environments. The adaptations and enhancements made to the original simulator enable realistic modeling of indoor scenarios with a large number of antennas and efficient handling of MPC points. The geometry-based stochastic channel modeling approach balances accuracy and computational efficiency, making the LIS simulator a powerful tool for researchers and practitioners in the field of LIS systems. The ability to connect with NS-3 further extends the simulator's capabilities, allowing for advanced simulations and analysis of channel characteristics in a comprehensive manner.

3.3.3 Current Status and way forward

The current status of the LIS simulator is that it is being developed and built upon the Sivert V2V simulator, with adaptations for indoor environments, complex shapes, efficient MPC handling, and scenario declaration for minimizing interaction with Unity. The simulator allows for real-time interactive participation of agents in a LIS system, investigates infrastructure management, explores use-case validity, verifies and develops channel models, and serves as a platform for machine learning.

Work is ongoing on externalising scenario definitions, data presentation, and data analysis to Python scripts for easy customisation and the ability to run large batches of simulations in parallel without user interaction. Unity will continue to be used for computing the channel and mobility models, and providing the user interface when running in interactive mode.

Furthermore, the LIS simulator will be extended to include indoor and cell-free scenarios with a massive number of antennas, to better reflect real-world LIS deployments and enable realistic testing of use cases defined in the Reindeer project. The simulator will also enable testing of dynamic resource allocation and federation orchestration based on environmental and application requirements, facilitating the study of dynamic scenarios in a LIS system. Cross-layer capabilities will be incorporated to capture the effects of changes in the physical layer on the application layer, allowing for more comprehensive simulations and analysis of cross-layer interactions.

Additionally, the LIS simulator will provide user interaction capabilities for greater flexibility and customization of simulations, allowing users to interact with and control the simulation in real-time. The simulator will also be leveraged for synthesizing data for machine learning applications, such as channel modeling, performance evaluation, and training machine learning algorithms for LIS systems.

Future work on the LIS simulator will focus on these objectives to further improve its capabilities and expand its utility in simulating LIS systems in various scenarios and use cases. This includes increasing usability through disentangling Unity from logic, providing a user-friendly interface, and allowing for easy customization through external Python scripts. The adaptation of the simulator with new MPC spacing methods and more efficient look-up tables for multi-path components will also enhance the simulation efficiency, enabling support for the massive number of antennas and UEs expected in LIS deployments.

Chapter 4

Experiments

In WP5, experiments will be carried out to directly evaluate he use cases presented in D1.1 [31]. Furthermore, the feasibility of practical implementations will be assessed by conducting these experiments. This chapter highlights the experiments planned to evaluate KPIs. It describes which KPIs could be potentially investigated by which testbed, energy-neutral device and/or simulators. This is summarized in Table 4.1.

Five experiments we presented to assess some key features of RW, with respect to other 5G and beyond 5G systems. To provide sufficient KPI coverage, each experiment evaluates different KPIs. An overview of these experiments can be found in Table 4.2. It indicates which testbed, EN devices and simulators are used to address the aforementioned KPIs.

First, the grouping of infrastructure resources tailored to the application requirement will be studied. The robustness-versus-latency trade-off will be experimentally validated when scaling the number of CSPs. Next, in experiment 3, accurate positioning will be assessed in the LARVA testbed, complemented by the TugSim simulator. Experiment 4 will dive deeper into powering, communicating with and positioning of energy-neutral devices. Four beamforming techniques will be considered to charge EN devices with practical implementations in mind. Lastly, in experiment 5 multi-user capabilities of RW will be validated through measurements of the RUSK testbed, accompanied by potentially an early prototype in the LuLIS testbed.

These experimental results will be taken as input in T1.3 and generalised to a wider scope and more complex deployments following the use case presented in D1.1 [31]. Therefore, WP5 serves as a foundation for experimental investigation of KPIs, while the use case evaluation is performed in T1.3. This deliverable describes the scope and aim of the planned experiments.

Table 4.1 summarises the KPIs outlined in D1.1 and how each testbed, simulator, and EN device is able to assess these KPIs. To provide sufficient coverage, five main experiments are planned, each evaluating different KPIs. An overview of these experiments can be found in Table 4.2. It indicates which testbed, EN devices and simulators are used to address the aforementioned KPIs.

Table 4.1: Indicates which KPIs can be evaluated by which testbed, KPIs 1-12 are defined in [31]. The grey check marks (✓) indicate that a testbed can potentially include the KPI, but this is not planned in WP5. Note that next to EN devices, other conventional hardware will be used to emulate the UE, e.g., SDRs or signal generators.

		Testbeds							EN devices		Simulators	
Num.	Key performance indicator	Techtile	RUSK	LUMaMi	KULMaMi	LuLIS	LARVA	LUDyna	KUL END	NXP END	TugSim	LuSim
1	A massive number of devices/connections		✓	✓		✓	✓		✓	✓	✓	✓
2	High traffic volume/sum rate		✓	✓		✓						✓
3	High peak rate		✓			✓				✓		✓
4	Higher mobility speeds (up to 7-10 m/s)		✓	✓		✓		✓		✓	✓	✓
5	Deployment in open or complex environments		✓	✓	✓	✓		✓	✓	✓		✓
6	High reliability		✓	✓		✓	✓	✓		✓	✓	✓
7	Low latency		✓	✓		✓		✓		✓		✓
8	High positioning accuracy	✓	✓	✓		✓	✓			✓	✓	✓
9	Low-energy communication	✓					✓		✓	✓	✓	✓
10	Low-energy positioning	✓					✓		✓	✓	✓	✓
11	Energy-neutral devices and WPT	✓	✓				✓		✓	✓	✓	✓
12	Real-time processing, edge computing support	✓	✓	✓	✓	✓						
13	Efficient resource allocation	✓	✓			✓			✓	✓		✓

Table 4.2: Overview of the planned experiments.

Num.	Experiments	KPIs	Testbeds	EN devices	Simulators
1	Dynamic resource allocations via federations	5, 13	Techtile		LuSim
2	Robustness versus latency trade-off	6, 7	LuLIS and Techtile		LuSim
3	Accurate positioning	8	LARVA		TugSim
4	Powering, communicating with and positioning of EN devices				
4a	Powering EN devices	11	LARVA and Techtile	KUL EN and NXP EN	TugSim
4b	Communicating with EN devices	9	Techtile	KUL EN and NXP EN	TugSim
4c	Positioning of EN devices	10	LARVA and Techtile	KUL EN	TugSim
5	Multi-user capabilities	1, 2	Techtile and RUSK		LuSim

Experiment 1. Dynamic resource allocations via federations

Due to the rich set of diverse services, the set of resources used, *e.g.*, processing and radio elements, are tailored to the particular application. This means that the wireless access infrastructure must allocate resources to specific applications with different requirements. For example, for wireless power transfer, charging resources located near the intended device will yield the highest efficiency. In contrast, in XR applications the mobility of the user's body in space and the head movement require a high spatial diversity of the antenna resources to mitigate outages and peaks in latency. This demonstrates the need for grouping of resources in a cell-free context in a dynamic manner, *i.e.*, in both the temporal and spatial domain. In REINDEER, we introduced the term *federation(s)* to denote the group of resources that jointly serve a given application. In this experimental plan, the 'real-time' and 'real-space' operation for resilient interactive wireless applications is studied through this dynamic resource allocation. In this experiment, the federation orchestration algorithms, introduced in D3.2 and [7], are investigated. The experiment will demonstrate the switching of federations depending on the requirements and position of the UE. Based on the conducted experiments, the LuSim will be extended to run multiple scenarios, complementing the experiments. The diverse set of requirements, *e.g.*, positioning versus low-latency communication, can be assessed in Techtile in an interactive manner. It will also serve as the demonstrator of D5.2, illustrating the near-real-time operation of creating, moving, shrinking, and expanding federations based on physical (PHY) layer conditions.

Experiment 2. Robustness versus latency trade-off

Adding more CSPs with interference cancellation algorithms to provide more robust services could potentially reduce latency by avoiding retransmissions. On the other hand, more CSPs imply increased latency in coherence processing, *e.g.*, for ZF precoding and detection. As discussed in [8], [39], the processing latency depends on selected signal processing algorithms, the topology that connects CSPs, and also the deployment scenarios. For instance, the processing latency increases linearly with the number of CSPs when the recursive least squares (RLS) algorithm is mapped to a Daisy-Chain topology. Moreover, information exchange between CSPs with large front-haul distance (FD) may require multi-hop routing, which will result in much greater latency. An interesting aspect to be considered is that further increasing the number of CSPs may allow the use of local-processing-only algorithms, *e.g.*, maximum ratio combining (MRC), which will significantly reduce processing latency by parallel processing that minimizes the information

exchange between CSPs. In summary, there is a non-trivial trade-off between the robustness and latency when changing the number of CSPs in service. In this experiment, we will use LuSim as the main platform for design space exploration to find the number of CSPs that balances robustness and latency. Real-life hardware parameters from LuLIS and Techtile, *e.g.*,, processing and information exchange latency, will be imported to LuSim for calibrated simulation.

Experiment 3. Accurate positioning

A RW infrastructure holds the potential for accurate positioning. RW panels benefit from very large apertures, giving them high *angular resolutions*. The *range resolution* per panel suffers from a low bandwidth. However, multiple distributed RW panels can cooperatively recover a great "range resolution", and a low positioning error bound (PEB) in general. Particularly, if the RW panels are operated *coherently*, there are exceptional positioning accuracies achievable [43]. Eventually, this sets the stage for joint positioning and synchronization algorithms.

The REINDEER consortium develops positioning bounds and algorithms in the deliverable D3.3 [36]. Both are initially evaluated on models and synthetic data. As part of the experiment plan, we will conduct measurement-based validations of the achievable positioning and synchronization accuracies. A particular focus lies on the implementation of robust algorithms that will provide *resilient* communication and power transfer. Environment-awareness is leveraged to achieve this robustness against possible multipath fading, and obstructions (i.e., shadowing) while accommodating a high user mobility. Tracking of devices, paired with geometry-based channel prediction, helps to anticipate unfavourable channel conditions and use diversity to bypass the affected channels. These synergies of positioning and environment awareness with communication and power transfer ultimately serve the needs of coming generations of use cases in 6G and beyond.

We anticipate to evaluate the enhancement of robustness to the positioning through the Blooloc positioning engine. The software is ideally suited to model imperfections of raw positioning sensor data to achieve robust positioning. To that extent, one needs to model the possible uncertainty/unreliability of the underlying sensors in a likelihood model for that specific sensor, and such models can be readily plugged into the positioning engine.

Experiment 4. Powering, communicating with and positioning of energy neutral devices

4a. Powering EN devices

A wireless infrastructure operating fully passive EN devices faces the initial access problem: When it is put into operation for the first time, it aims to communicate with EN devices and supply them with power wirelessly. Different strategies will be experimentally investigated to perform WPT to the EN devices, *i.e.*, through:

1. reciprocity-based beamforming,
2. geometry-based beamforming,
3. codebook-based beamforming,
4. random-phased charging.

As reciprocity-based beamforming (strategy 1), obtained through sending uplink pilots, is unavailable before the initial wake-up of an EN device, geometry-based (strategy 2), *i.e.*,, *predicted* CSI can be used to overcome this problem by predicting channels from a known geometry, i.e., known positions of RWs panels and EN devices. Unfortunately, the position $\boldsymbol{p}$ of an EN device cannot always be assumed known, which adds to the complexity of the initial access problem. To compensate for a possibly unknown position $\boldsymbol{p}$, a *beam sweep* can be conducted, *i.e.*, a space of possible positions $\boldsymbol{p}$ can be searched iteratively until the initial wakeup of the EN device. It should be noted that geometry-based channel prediction demands a known geometry (including the array layout) and also a phase-calibrated array, *i.e.*, the phases at all antennas within one array need to be aligned. Other approaches like (exhaustive) codebook-based beamforming (strategy 3), or random-phased charging (strategy 4) do not demand geometric knowledge or phase-calibrated arrays. The former can achieve good performance if iterated through an exhaustive codebook, but the codebook search becomes prohibitive in mMIMO applications due to the large codebook sizes. The latter approach, *i.e.*, random-phased charging (strategy 4) can be utilised to slowly charge devices, only for initial access without knowing their location. After being charged, a pilot can be sent to enable positioning and reciprocity-based beamforming.

Geometry-based WPT.

In [40], the REINDEER consortium has used the TugSim simulator to show how *beam diversity* can be used to overcome impairments in the initial access introduced by an environment with strong multipath propagation. In [42], the LARVA testbed was used to demonstrate how geometry-based beamformers can be used to solve the initial access problem. A variety of geometry-based beamformers have been tested on synthetic aperture measurements acquired with the LARVA testbed. Furthermore, a random (opportunistic) beamformer has been explored using the TugSim simulator to solve the initial access problem and a closed-loop approach has been implemented, which iteratively performs positioning and geometry-based beamform.

The deliverable D4.2 [42] represents a foundation for exploiting geometry-based predicted CSI for the initial access procedure. So far, a complete exhaustive beam sweep has been performed only with a simple planar wavefront beamformer. Conducting exhaustive beam sweeps with spherical wavefront beamformers may be costly both in terms of time and energy. As part of the experiment plan, particularly with respect to [44], the REINDEER consortium will investigate different beam sweeping schemes that will reduce sweep times and overcome problems associated with exhaustive beam sweeps.

In realistic scenarios, we will demonstrate how close we can get to these upper limits for power budgets. Using synthetic aperture measurements, we demonstrated that a (40×25)-URA transmitting $1\,\mathrm{W}$ of power at an operating at $3.79\,\mathrm{GHz}$ is capable to supply more than $1\,\mathrm{mW}$ of power to an EN device located at a distance of $12.3\,\mathrm{m}$ [37]. Lifting the receivable powers from the microwatt to the milliwatt region is a feature that a RadioWeaves infrastructure achieves through physically large apertures with massive, distributed antennas, and operating at sub-10 GHz frequencies. These are distinguishing characteristics in an era where most 6G research seeks to explore very high frequency ranges from the mmWave up to THz bands.

Random-phased WPT.

In addition to geometry-based WPT, random phased WPT can be used to charge EN devices initially. This approach reduces the exploration space to search for devices, resulting in a more

scalable method. Furthermore, this method does not impose that the infrastructure needs to operate coherently. Hence, it can be used before or during phase calibration of the network.

Power budgets for EN devices

In deliverable D4.1 [41], the REINDEER consortium has examined the achievable power budgets of a RW deployment with respect to regulatory constraints. A major finding of the analysis was that WPT benefits from physically large or distributed radio architectures, both in terms of high *efficiencies* through large apertures and *regulatory compliance* through spatially low power densities. Given that the RW infrastructure is distributed reasonably well with respect to the EN device, it may be possible to generate a focal region attaining the maximum regulatory-compliant power density while exhibiting lower power densities everywhere else. Based on that assumption, the REINDEER consortium proposed power budgets achievable with isotropic antennas at different sub-10 GHz frequencies [41, Table 4.1]. We demonstrate that the extra-large MIMO (XL-MIMO) measurements in [37] suffice the mentioned requirements, and an EN device located at a distance of $12.3\,\mathrm{m}$ can be supplied with a power close to that stated in the deliverable D4.1.

4b. Communicating with energy-neutral devices

The theoretical foundations on the two types of RF communication with Class 1 and Class 2 energy-neutral devices is detailed in D4.1 [41, Section 4.2]: (i) downlink data transfer and (ii) uplink data transfer. As stated, to reduce the complexity, power and cost, the same, simple modulation technique should be used. Within the planned experiments, the main focus lies on the uplink communication as, on a hardware level, this is more difficult to realise. Additionally, the processing power available at the infrastructure can be used to demodulate potential noise-induced signals.

To enable uplink data transfer on energy-neutral devices, backscatter communication is considered. The concept is to map the energy consumption in function of the uplink data speed for different modulation techniques. The main contribution of this section lies in the potential achievable data rate gain thanks to the improved SNR when using a power spot. In a bistatic, dislocated backscattering setup, an additional SNR gain could be observed with this power spot. As backscattering is, in essence, a mixing process, the dislocated receiver would not receive the carrier frequency that contributes to the noise, as this carrier is focused on the EN device position. We plan to investigate whether this could potentially eliminate the energy-consuming local oscillator on the energy-neutral devices that is necessary to overcome in-band interference. The potential data rate gain as predicted in D4.1 [41, Section 4.2] can be found in Table 4.3.

We plan to build a bistatic backscatter setup in the Techtile testbed. For the first experiments, the on-tile SDRs would be used to (i) generate a carrier wave and (ii) to receive the backscattered uplink data. The received baseband signals can be processed locally by a RPi 4. We plan to extend these experiments by generating power spots around the EN device hardware. The latter is explained in Section 2.3.2 and can support bistatic backscatter experiments since it meets the required specifications.

4c. Positioning of EN devices

Knowing the actual position of an EN device holds large potential from an application point-of-view. Real-time location data can improve safety, performance, experience, and convenience in

Table 4.3: Overview of the maximum, best-case data rate. The term MISO relates to the fact that the array gain is only exploited in the downlink to the EN device and not in the (backscattered) uplink from the EN device, closely corresponding to the initial access problem described in D4.1 [41, Section 4.2].

SISO data rates	868 MHz	2.4 GHz	5.0 GHz
$R_{\max}$ (kbit s^{-1}), no local oscillator	0.15	0.29	0.29
$R_{\max}$ (kbit s^{-1}), with local oscillator	913	1315	360
MISO data rates	868 MHz	2.4 MHz	5.0 GHz
$R_{\max}$ (kbit s^{-1}), no local oscillator	138	277	277
$R_{\max}$ (Mbit s^{-1}), with local oscillator	9.79	18.35	14.13

different types of solutions. Positioning is susceptible to the initial access problem. Before positioning, EN devices need to be powered and have to communicate, *e.g.*, through uplink pilot transmission. In this experiment, the *closed loop approach* elaborated in [42] is experimentally evaluated.

Results from the first step in this closed loop approach can provide useful information on the uniformity of the power density throughout the experimental setup when random phase errors are introduced on purpose to each antenna element. In the next steps, the concept is to test different positioning algorithms based on pilot signals that can be generated with energy efficient backscattering communication. The positioning algorithms will be processed on the infrastructure side, to fully unburden the EN device. The Euclidean distances on multitude of locations gives a good measure of how well the tested algorithms perform. In addition, we plan to research the influence of the amount of closed-loop iterations to reduce this error between the actual and estimated position. With the envisioned communication setup in the Techtile infrastructure, the influence of the number of active antenna elements (receive and transmit) used for localization can be tested. The result of these tests can compare the received energy before and after the power spot generation. Power spot widening and EN device tracking could be implemented and tested at a later measurement stage.

Experiment 5. Multi-user capabilities

The multi-user capabilities of a dense RW infrastructure will be assessed through indoor channel measurements originating from the RUSK testbed. This allows experiments with very large numbers of antennas using measured channel data. The channel data are, however, measured in a static environment, creating the possibility to simulate a very large virtual array where different subarray configurations can be used to assess the multi-user capabilities of a RadioWeaves. The limitations of these experiments are the absence of a dynamic propagation environment to perform experiments on. However, user mobility can be simulated to some extent by switching between channel responses measured at different points during a simulation.

There is also an option to use an early prototype of the LuLIS testbed, currently under construction, for performing real-time multiuser experiments. To what extent such tests can be done and made available to the REINDEER project depends on the development status of the testbed in the last six months of the REINDEER project.

Chapter 5

Conclusion

This deliverable lays the foundation for the experimental work conducted in REINDEER. It provides insides into the feasibility of implementing RadioWeaves (RW) in practical scenarios and assesses the key performance indicators (KPIs) of REINDEER.

To do so, the following method is used. A standard is developed to enable storing data sets originating from measurements and simulations, *i.e.*, Dataset Storage Standard (DSS). It contains all the information regarding the testbed, environment and scenario. The resulting datasets are thus thereby self-describing. The testbeds are categorised based on their intended use and capabilities, resulting in four distinct testbed levels. Level 1 testbeds are designed to study one particular part of a system, *e.g.*, the antenna array topology. Level 2 testbeds, assess the link between two point, *i.e.*, they investigate the wireless propagation. Level 3 and 4 tesbeds are capable of evaluating end-to-end performance of the network. L4 testbeds extend L3 testbeds by providing real-time processing capabilities. In REINDEER, three L2 tesbeds, one L3 and four L4 testbeds are available or are being implemented. Next to the testbeds, two energy-neutral devices are under development to evaluate the proposed energy neutral (EN) operation of RW. To extend and complement the testbeds and energy-neutral devices (EN devices), two simulators are being designed. A joint approach will be utilized for some experiments, where testbed measurements or hardware models will be included in the simulations.

This deliverable contains the KPIs that can be evaluated by each test bed, EN device and simulator. A clear focus will be placed on planning experiments within REINDEER's scope. Five experiments were presented that assessed some key features of RW relative to other 5G systems and beyond 5G. A first experiment will analyse the grouping of infrastructure resources tailored to the application requirements. The robustness versus latency trade-off will be tested experimentally when scaling the number of contact service points (CSPs). The accuracy of the positioning and the robustness will be evaluated. In addition to this, energy-neutral devices powering, communication and positioning will be experimentally validated, considering four distinct beamforming techniques to charge EN devices with practical implementation in mind. Finally, the multi-user capabilities of RW will be validated.

Bibliography

[1] M. Truskaller, L. Fabrete, A. Stanek, D. Delabie, L. Van der Perre, E. G. Larsson, S. Rimalapudi, E. Fitzgerald, F. Tufvesson, O. Edfors, *et al.*, "Use case-driven specifications and technical requirements and initial channel model," 2021.

[2] S. Malkowsky, J. Vieira, L. Liu, P. Harris, K. Nieman, N. Kundargi, I. C. Wong, F. Tufvesson, V. Öwall, and O. Edfors, "The World's First Real-Time Testbed for Massive MIMO: Design, Implementation, and Validation," *IEEE Access*, vol. 5, pp. 9073–9088, 2017, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2705561.

[3] J. R. Sánchez, F. Rusek, O. Edfors, M. Sarajlić, and L. Liu, "Decentralized massive MIMO processing exploring daisy-chain architecture and recursive algorithms," *IEEE Transactions on Signal Processing*, vol. 68, pp. 687–700, 2020.

[4] J. R. Sánchez, F. Rusek, O. Edfors, and L. Liu, "Distributed and Scalable Uplink Processing for LIS: Algorithm, Architecture, and Design Trade-Offs," *IEEE Transactions on Signal Processing*, vol. 70, pp. 2639–2653, 2022.

[5] D. Wang, C. Zhang, Y. Du, J. Zhao, M. Jiang, and X. You, "Implementation of a Cloud-Based Cell-Free Distributed Massive MIMO System," *IEEE Communications Magazine*, vol. 58, no. 8, pp. 61–67, 2020. DOI: 10.1109/MCOM.001.2000106.

[6] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, *et al.*, "The FAIR Guiding Principles for scientific data management and stewardship," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[7] G. Callebaut, W. Tärneberg, L. Van der Perre, and E. Fitzgerald, "Dynamic Federations for 6G Cell-Free Networking: Concepts and Terminology," in *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC) (IEEE SPAWC 2022)*, Oulu, Finland, Jul. 2022.

[8] "Evaluation of the distribution of processing across infrastructure and associated requirements on back-haul and synchronization, Reindeer deliverable D2.2," Tech. Rep., version 1, 2023.

[9] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008)*, pp. 1–499, 2020. DOI: 10.1109/IEEESTD.2020.9120376.

[10] N. Finn, "Introduction to Time-Sensitive Networking," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 22–28, 2018. DOI: 10.1109/MCOMSTD.2018.1700076.

[11] J. Serrano, M. Lipinski, T. Wlostowski, E. Gousiou, E. van der Bij, *et al.*, "The white rabbit project," 2013.

[12] J. Vieira and E. G. Larsson, "Reciprocity calibration of Distributed Massive MIMO Access Points for Coherent Operation," in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2021, pp. 783–787. DOI: 10.1109/PIMRC50174.2021.9569495.

[13] U. K. Ganesan, R. Sarvendranath, and E. G. Larsson, *BeamSync: Over-The-Air Carrier Synchronization in Distributed RadioWeaves*, 2021. arXiv: 2112.00592 [cs.IT].

[14] *USRP Hardware Driver and USRP Manual*, USRP B2x0 Series, Ettus Research, 2021. [Online]. Available: https://files.ettus.com/manual/page%5C_usrp%5C_b200.html.

[15] S. Chaudhary and A. Samant, "Characterization and calibration techniques for multi-channel phase-coherent systems," in *2015 IEEE AUTOTESTCON*, 2015, pp. 334–338. DOI: 10.1109/AUTEST.2015.7356512.

[16] G. Callebaut, *Advances in Single and Multi-Antenna Technologies for Energy-Efficient IoT*, eng, PhD dissertation, 2021.

[17] "Xilinx RFSoC 2 Kit." (), [Online]. Available: https://www.xilinx.com/support/university/xup-boards/RFSoC2x2.html. (accessed: 01.28.2023).

[18] "PYNQ: PYTHON PRODUCTIVITY." (), [Online]. Available: http://www.pynq.io/. (accessed: 01.28.2023).

[19] S. De Bast and S. Pollin, "User Localisation in Massive MIMO Networks," eng, PhD Thesis, 2022.

[20] J. Vieira, F. Rusek, O. Edfors, S. Malkowsky, L. Liu, *et al.*, "Reciprocity Calibration for Massive MIMO: Proposal, Modeling, and Validation," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 3042–3056, 2017. DOI: 10.1109/TWC.2017.2674659.

[21] A. Nordrum. "5G Researchers Set New World Record For Spectrum Efficiency," IEEE Spectrum. (May 20, 2016), [Online]. Available: https://spectrum.ieee.org/5g-researchers-achieve-new-spectrum-efficiency-record.

[22] P. Harris, S. Malkowsky, J. Vieira, E. Bengtsson, F. Tufvesson, W. B. Hasan, L. Liu, M. Beach, S. Armour, and O. Edfors, "Performance Characterization of a Real-Time Massive MIMO System With LOS Mobile Channels," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1244–1253, Jun. 2017, ISSN: 1558-0008. DOI: 10.1109/JSAC.2017.2686678.

[23] M. Chung, L. Liu, A. Johansson, S. Willhammar, M. Nilsson, Z. Ying, O. Zander, K. Samanta, C. Clifton, T. Koimori, S. Morita, S. Taniguchi, F. Tufvesson, and O. Edfors, "LuMaMi28: Real-Time Millimeter-Wave Multi-User MIMO Systems with Antenna Selection," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023. DOI: 10.1109/TWC.2023.3257195.

[24] P. Skarin, W. Tärneberg, K.-E. Årzen, and M. Kihl, "Towards mission-critical control at the edge and over 5G," in *2018 IEEE international conference on edge computing (EDGE)*, IEEE, 2018, pp. 50–57.

[25] S. Gunnarsson, S. Malkowsky, H. Tataria, F. Tufvesson, J. Schwarz, U. Glass, H. Raidt, and A. Rostami, "REPORT ON 5G RADIO DEPLOYABILITY IN THE FACTORY," EU project: 5G for Smart Manufacturing, Deliverable D4.2, version 2.0, May 31, 2022. [Online]. Available: https://5gsmart.eu/wp-content/uploads/5G-SMART-D4.2-v2.0.pdf.

[26] I. Yaman, G. Tian, M. Larsson, P. Persson, M. Sandra, A. Dürr, E. Tegler, N. Challa, H. Garde, F. Tufvesson, K. Åström, O. Edfors, S. Malkowsky, and L. Liu, *The LuViRA Dataset: Measurement Description*, 2023. DOI: 10.48550/ARXIV.2302.05309. [Online]. Available: https://arxiv.org/abs/2302.05309.

[27] Abbas, Taimoor, "Measurement Based Channel Characterization and Modeling for Vehicle-to-Vehicle Communications," eng, Ph.D. dissertation, Lund University, 2014, ISBN: 978-91-7473-853-7 (pdf). [Online]. Available: %7Bhttps://lup.lub.lu.se/search/files/3228298/4251868.pdf%7D.

[28] REINDEER Project, "Propagation characteristics and channel models for RadioWeaves including reflectarrays," Deliverable ICT-52-2020 / D1.2, 2023 (unpublished).

[29] C. Nelson, X. Li, T. Wilding, B. Deutschmann, K. Witrisal, and F. Tufvesson, "Large intelligent surface measurements for joint communication and sensing," *arXiv preprint arXiv:2304.12457*, 2023.

[30] *SRS time and frequency standards*, SRS F725/F740, Stanford Research Systems, 2023. [Online]. Available: https://www.thinksrs.com/products/time.html.

[31] J. F. Esteban and M. Truskaller, "Use case-driven specifications and technical requirements and initial channel model," REINDEER project, Deliverable ICT-52-2020 / D1.1, Sep. 2021. DOI: 10.5281/zenodo.5561844.

[32] J. R. Costa, C. R. Medeiros, and C. A. Fernandes, "Performance of a crossed exponentially tapered slot antenna for UWB systems," vol. 57, no. 5, pp. 1345–1352, May 2009, ISSN: 0018-926X. DOI: 10.1109/TAP.2009.2016727.

[33] C. Krall, "Signal processing for ultra wideband transceivers," Ph.D. dissertation, Graz University of Technology, 2008.

[34] P. Meissner, E. Leitinger, and K. Witrisal, "UWB for robust indoor tracking: Weighting of multipath components for efficient estimation," *IEEE Wireless Commun. Lett.*, vol. 3, no. 5, pp. 501–504, 2014.

[35] KU Leuven, *Energy Neutral Device from KU Leuven*, https://github.com/DRAMCO/EN-device-backscatter, 2023.

[36] R. project, "Position Estimation and Environment Learning," REINDEER project, Deliverable (unpublished) ICT-52-2020 / D3.3, 2023.

[37] B. J. B. Deutschmann, T. Wilding, M. Graber, and K. Witrisal, *XL-MIMO Channel Modeling and Prediction for Wireless Power Transfer*, 2023. DOI: 10.48550/ARXIV.2302.11969.

[38] B. J. B. Deutschmann, M. Graber, T. Wilding, and K. Witrisal, *Bistatic MIMO radar sensing of specularly reflecting surfaces for wireless power transfer*, 2023. DOI: 10.48550/ARXIV.2305.05002.

[39] O. Edfors, R. Brazil, H. Petautschnig, G. Callebaut, T. Feys, L. V. der Perre, E. G. Larsson, O. Edfors, E. Fitzgerald, L. Liu, J. R. Sanchez, W. Tärneberg, P. Frenger, B. Deutschmann, T. Wilding, and K. Witrisal, "Initial assessment of architectures and hardware resources for a RadioWeaves infrastructure, Reindeer deliverable D2.1," Tech. Rep., version 1, Jan. 2022. DOI: 10.5281/zenodo.5938909. [Online]. Available: https://doi.org/10.5281/zenodo.5938909.

[40] B. J. B. Deutschmann, T. Wilding, E. G. Larsson, and K. Witrisal, "Location-based Initial Access for Wireless Power Transfer with Physically Large Arrays," in *WS08 IEEE ICC 2022 Workshop on Synergies of communication, localization, and sensing towards 6G (WS08 ICC'22 Workshop - ComLS-6G)*, Seoul, Korea (South), May 2022.

[41] C. Buyle, B. Cox, D. Delabie, B. Deutschmann, T. Wilding, M. Graber, K. Witrisal, and U. Mühlmann, "System design study for energy-neutral devices interacting with the RadioWeaves infrastructure," REINDEER project, Deliverable ICT-52-2020 / D4.1, 2022.

[42] R. project, “Signal processing for WPT and uplink signalling, exploiting environment awareness,” REINDEER project, Deliverable (unpublished) ICT-52-2020 / D4.2, 2023.

[43] A. Fascista, B. J. B. Deutschmann, M. F. Keskin, T. Wilding, A. Coluccia, K. Witrisal, E. Leitinger, G. Seco-Granados, and H. Wymeersch, *Uplink Joint Positioning and Synchronization in Cell-Free Deployments with Radio Stripes*, 2023. DOI: 10.48550/ARXIV.2302.03387. arXiv: 2302.03387 [eess.SP].

[44] R. project, “Validation of concepts and experimental assessment of key technologies,” REINDEER project, Deliverable (unpublished) ICT-52-2020 / D5.3, 2024.