WORTECS

H2020-ICT-2016-2

RIA

Project-ID: 761329

WORTECS

Networking research beyond 5G

Deliverable 4.2

Radio communication prototype

Contractual Date of Delivery:	**2019, August**
Actual Date of Delivery:	**2019, October**
Editor(s):	**Vladica Sark (IHP)**
Author(s):	**Rodolphe Legouable (BCOM), Guillaume Vercasson (BCOM), Marcin Brzozowski (IHP), Lukasz Lopacinski (IHP),Vladica Sark (IHP)**
Work package:	**WP4**
Security:	**PU**
Nature:	**Deliverable**
Version:	**version 0.9**
Total number of pages:	37

Abstract

This deliverable presents the parts needed for demonstrating the radio system functionality, envisioned in the WORTECS project. Two radio demonstrators are planned. The first one should use 60 GHz analog frontends, due to unavailability of the 240 GHz frontends. The second should use 240 GHz analog frontends and reuse the baseband and MAC processor functionalities from the first demonstrator. In this deliverable the simulation results as well as the hardware in the loop results for the developed multi-gigabit baseband processor are shown. Further the implementation details of the developed real-time radio communication system are given. This deliverable also describes the HetNet system intended to be used in conjunction with the radio system and finally the Arctic Sea virtual reality platform is described.

Keyword list

Virtual Reality, Multi-gigabit, Terahertz, Baseband processor, Wireless, HetNet, radio transmission, MAC Layer, HW implementation

Executive Summary

This task is mainly focused on demonstrating the high speed wireless radio transmission. This document presents the results obtained from the performed simulations and shows that the proposed approach is capable of supporting multi-gigabit data rates. This was also confirmed with the hardware in the loop system, where the same system model, used for simulation, was used to transfer data over radio interface, in order to confirm that the gigabit speeds are achievable using a real wireless channel and real wireless transceivers.

Theoretically, these multi-gigabit throughputs in the 60 GHz band are possible. The main problem faced here is the processing speed and capacity of the digital platform used for processing of the digital baseband signal and implementing the data link layer. In this demonstrator, a field programmable gate array (FPGA) is used. It has limited resources and implementing multi-gigabit speeds is challenging. In simulation we have shown that we can easily achieve almost 8 Gbps data throughput using a 2 GHz channel bandwidth. This is also easily achievable in the hardware in the loop system. Anyway, due to limited resources of the FPGA, at this moment a working system with 1 Gbps real-time throughput can be demonstrated. The baseband supporting 4 Gbps was already implemented, but the current MAC is still being extended to support the 4 Gbps throughputs.

In order to be able to use the radio system with the optical, envisioned in the WORTECS project, a HetNet system was developed and its basic properties are shown here.

Finally, the Arctic Sea virtual reality test platform is described. This is the VR platform which should be used with the radio, as well as with the optical system to demonstrate the functionality of the radio and the optical data transmission system as well as the functionality of the HetNet system.

Impact on the other Work-packages

This deliverable results would mainly impact WP4 since these results would be used for development of the final demonstrator.

List of Authors

First name	Last name	Beneficiary	Email address
Vladica	Sark	IHP	sark@ihp-microelectronics.com
Rodolphe	Legouable	b<>com	rodolphe.legouable@b-com.com
Guillaume	Vercasson	b<>com	guillaume.vercasson@b-com.com
Marcin	Brzozowski	IHP	brzozowski@ihp-microelectronics.com
Lukasz	Lopacinski	IHP	lopacinski@ihp-microelectronics.com

Document History

First name	Last name	Version	Comments
Vladica	Sark	0.1 – August 20th	First draft
Lukasz	Lopacinski	0.2 – September 18th	First draft of the 60 GHz real-time system
Rodolphe	Legouable	0.3 – September 24th	Arctic Sea platform contribution
Guillaume	Vercasson	0.3 – September 24th	Arctic Sea platform contribution
Lukasz	Lopacinski	0.4 – September 25th	Final version of the 60 GHz real-time system
Marcin	Brozozowski	0.5 – October 3rd	HetNet final version
Vladica	Sark	0.6 – October 4th	Simulation and hardware in the loop results
Vladica	Sark	0.7 – October 8th	Initial deliverable version
Vladica	Sark	1.0 – October 9th	Final deliverable version

List of Acronyms

Acronym	Meaning
AD	Analog to digital
AGC	Automatic gain control
ARQ	Automatic repeat request
AWG	Arbitrary waveform generator
BPSK	Binary phase shift keying
CRC	Cyclic redundancy check
DA	Digital to analog
DSP	Digital signal processor
EVM	Error vector magnitude
FEC	Forward error correction
FF	Flip flop
FFT	Fast Fourier transform
FPGA	Field programmable gate array
FSM	Finite State Machine
HMD	Head mounted display
IFFT	Inverse fast Fourier Transofrm
IQ	In-phase-quadrature (usually meant for quadrature signals)
ISM	Industrial-Scientific-Medical
MAC	Medium Access Control
OFDM	Orthogonal frequency division multiplexing
PCB	Printed circuit board
QAM	Quadrature amplitude modulation
QPSK	Quadrature phase shift keying
SDR	Software defined radio
SNR	Signal-to-noise-ratio
TDM	Time division multiplex
VR	Virtual reality
WiFi	Wireless Fidelity
WORTECS	Wireless Optical/Radio TErabit CommunicationS

Table of contents

List of Tables

List of Figures

1. Introduction

With the current development of semiconductor technology, more applications requiring higher data rates, are emerging. Huge number of these applications are requiring wireless connections in order to perform optimally. The industrial-scientific-medical (ISM) bands are especially interesting for many applications since no licensing is required. They are free for use by the users which comply with the maximum allowed transmit power as well as with the occupied bandwidth and maximal power spectral density. On the other hand this is also a disadvantage, since the band becomes congested and achieving higher throughputs is often challenging.

In the WORTECS project, the main use case is Virtual Reality (VR) scenario. In this project the link between the head mounted display (HMD) and the central computer used for processing of the 3-dimensional (3D) virtual reality (VR) scene should be wireless. The available ISM bands are a perfect candidate since mature wireless data transmission technologies for use in these bands already exists. WiFi in the 2.4 GHz and 5 GHz ISM bands can fit perfectly for this use case. Unfortunately, the video resolution available in the VR helmet would be extremely limited, probably to the lowest one, due to the limited data rate in these bands. Additionally, these bands are congested from other neighboring users, mainly WiFi and Bluetooth devices, which additionally limits the usability of these bands for real-time VR applications. The 60 GHz ISM band offers more bandwidth. The new IEEE 802.11ad compliant devices enable data rates up to 7 Gbps. These data rates would enable significant increase of the VR HMD video resolution and enable better user experience. Some VR HMDs already use this technology [CUE18]. The 60 GHz technology is also becoming cheaper and, therefore, widely commercially available. Nevertheless, as specified in D2.3 [WD23], the high end VR helmets should support much higher video resolutions, which leads to multi-gigabit data rates, not supported even by the 60 GHz systems. The main limitation of the 60 GHz ISM band as well the 2.4 and 5 GHz ISM bands is the limited link budget. Given the maximum allowed transmit power as well as the available channel bandwidth and the common distances in a typical VR scenario, the required data rates, estimated in D2.3 [WD23] cannot be achieved. In order to achieve these data rates, it is necessary to either increase the transmit power (usually regulated and limited by regulation bodies) or to use larger channel bandwidths. Larger channel bandwidths are only available in the terahertz (THz) spectrum. Therefore, this spectrum is especially interesting for high data rate applications like VR use case in the WORTECS project.

The main advantages of the THz band is the large available bandwidth, which for frequencies of 275 GHz and above is not currently regulated. The frequencies envisioned in the WORTECS project are around 240 GHz. The main reason is that at these frequencies a data communication band is envisioned and the second reason is that the current IHP's technology [IHP] is not able to produce enough output power for higher frequencies. With the future IHP SiGe technology (still in experimental phase at the moment), higher frequencies can be targeted. The available output power is important since the free space path loss at these frequencies is significant. Having low output power would significantly limit the link budget,

Two prototypes are envisioned in the WORTECS project. The first one should be lower data rate prototype, working in the 60 GHz ISM band, mainly because the terahertz analog front-ends are still not available at this point. The second demonstrator should be high data rate and should use the developed and manufactured terahertz analog front-ends.

The main idea is to develop the main parts of the prototype and to reuse them in the second demonstrator. Probably not all of the developed parts would be reused, but most of them would be designed with reusability in mind. Additionally, tests would be performed with the first demonstrator in order to find the possible weaknesses which can be corrected in the second version of the demonstrator.

2. Multi-gigabit radio transmission prototype model

As mentioned previously the first demonstrator would work in the 60 GHz band. The available bandwidth per channel in this band is 2 GHz per channel with a total of 4 channels (depending on the county). The baseband processor was modelled in MATLAB using floating point arithmetic. The later implementation on FPGA must be performed in fixed point arithmetic and, therefore, a special care should be taken.

2.1. Baseband processor

The baseband processor was designed to support multi-gigabit data rates. It supports only orthogonal frequency division multiplex (OFDM) modulation. Each subcarrier can support multiple modulation schemes. Also a convolutional code supporting different coding ratios with a Viterbi code [VIT05] is implemented as an inner code for forward error correction (FEC). As an outer FEC coder, a Reed-Solomon code is used. This coder can be used or simply bypassed since the convolutional coder already corrects all of the errors in most of the cases.

2.1.1. Supported modulation coding schemes

In Table 1 the implemented OFDM modulation coding schemes are shown. The baseband processor can support only a single modulation coding scheme on all of the subcarriers. The modulation coding scheme is fixed and cannot be changed during runtime. Dynamic change of the modulation coding scheme is not supported in this baseband processor.

Table 1 - Implemented modulation coding schemes and the achievable data rates in the matlab model

Modulation	Coding ratio	Gross data rate [Gbps]	Net data rate [Gbps]
BPSK	1/2	1.30	0.65
BPSK	2/3	1.30	0.87
QPSK	1/2	2.59	1.30
QPSK	2/3	2.59	1.73
16QAM	1/2	5.18	2.59
16QAM	2/3	5.18	3.45
16QAM	3/4	5.18	3.88
64QAM	2/3	7,77	5.18
64QAM	3/4	7,77	5.82

As can be noticed the maximal gross data rate is 7.77 Gbps to which corresponds a net data rate of 5.82 Gbps.

2.1.2. Simulation results

The created MATLAB model includes both the transmitter and the receiver. It also models the impairments in the analog frontend as well as the channel and the noise at the receiver. In the current simulation only AWGN channel was used and 300 frames with length of 2164 bytes each were transmitted. The constellation for the 64QAM modulation is shown in Figure 1.

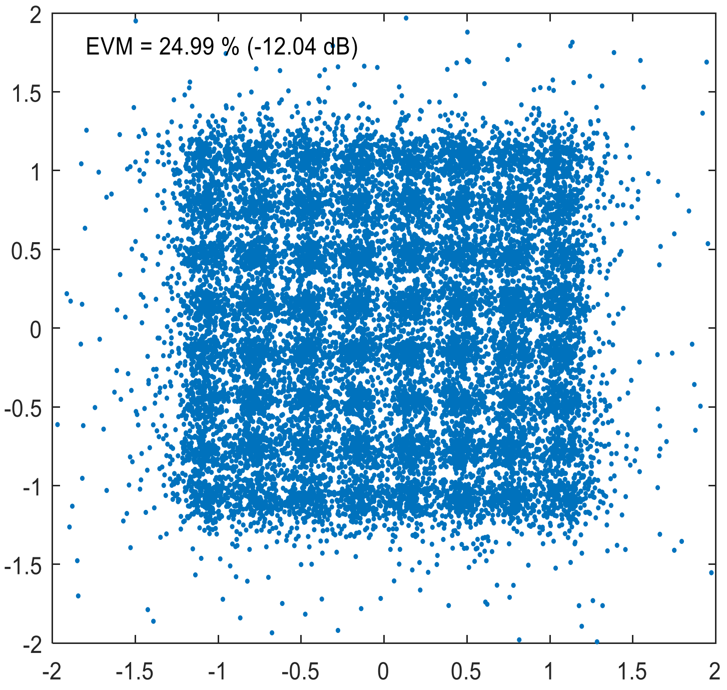

Figure 1 - Simulation of the baseband - 64QAM with 3/4 convolutional coding - net data rate of 5.82 Gbps

The signal to noise ratio (SNR) at the receiver is 20 dB, which makes the constellation plot quite noisy. Nevertheless, the Viterbi decoder succeeds to correct all the bit errors at the receiver.

In Figure 2 a simulation of 16 QAM OFDM modulation with 3/4 convolutional coding is shown.

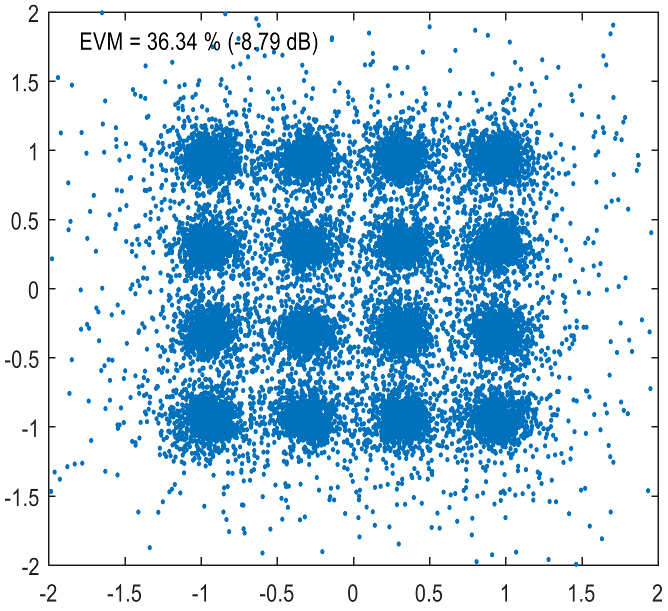

Figure 2 - 16QAM simulation

In the both cases, 64 QAM and 16 QAM, not a single frame error was detected by sending 300 frames.

2.2. MAC layer processor

The medium access control (MAC) processor complexity is significantly lower compared to the baseband processor. Therefore, the simulation of the MAC processor can be performed directly in VHDL (a hardware description language). This is more natural, since the MAC processor does not perform any signal processing but works directly on bits and implements multiple finite state machines (FSM) for controlling the MAC behavior.

2.2.1. MAC functionalities

The MAC processor performs a few important tasks. At the transmitter it receives the frames from the higher layer, buffers them, concatenates them in order to make a larger frame, encapsulates it and adds additional data in the encapsulating frame. This large frame is later sent to the baseband processor for transmission. Further, the MAC processor waits for the frame to be acknowledged. The frames incoming from the higher layers are meanwhile buffered. After receiving acknowledge for the previous frame, a new frame is transmitted. In case no acknowledge for a frame is received, or a negative acknowledge is received, the same frame is retransmitted. The maximal number of retransmissions can be set in the MAC processor.

At the receiver side, the MAC processor is receiving the frames and checks the CRC of the frame. If the frame CRC is correct the smaller frames, previously grouped in a large frame, are unpacked and sent to the higher layers. In case the CRC is not correct a negative acknowledge is sent to the transmitter.

At this moment, the MAC has only a minimal support for point to multipoint links and should be additionally improved.

3. Multi-gigabit radio transmission prototype hardware in the loop system

In order to test the developed baseband before creating the hardware real-time system, a so called hardware in the loop system was built and tested. The MATLAB model is used to generate the samples of the transmit frame, which are then sent using a 60 GHz analog frontend. After receiving the frames with another 60 GHz analog frontend, the frame are decoded with the same baseband processor model in MATLAB.

The signals accepted by the analog frontends are analog and, therefore, AD and DA converters are needed. As a DA converter an arbitrary waveform generator (AWG) can be used. Samples are sent from MATLAB to the AWG and a command for transmitting the samples is issued. As an AD converter at the receiver, an oscilloscope can be deployed. The scope samples the incoming frame and later the samples are read back from the MATLAB code. The two instruments are connected with a triggering cable, but also the oscilloscope can be triggered by the incoming frame itself. Another approach is to use the IHP's digiBackBoards [DIGI19], which can be used in so called software defined radio (SDR) mode. In this mode the samples are sent to the digiBackBoard and stored into its own memory. After sending a start signal, the samples are sent to the DA converter and the resulting analog signal is brought to the analog frontend. On the receiver side, the signal is received with the analog frontend and brought to the analog inputs of the digiBackBoard. The board is triggered from the transmitter boars and starts sampling the analog signal. The samples are stored in the RAM of the board and, when the sampling finishes, they are transferred in MATLAB for further processing. These samples are fed into the receiver MATLAB model and the received data is decoded.

3.1. System description

The test system is shown in Figure 3 and in Figure 4. In Figure 3 the test system uses the analog frontends from IHP while the test system from Figure 4 uses the analog system from Sivers IMA [SIVERS].

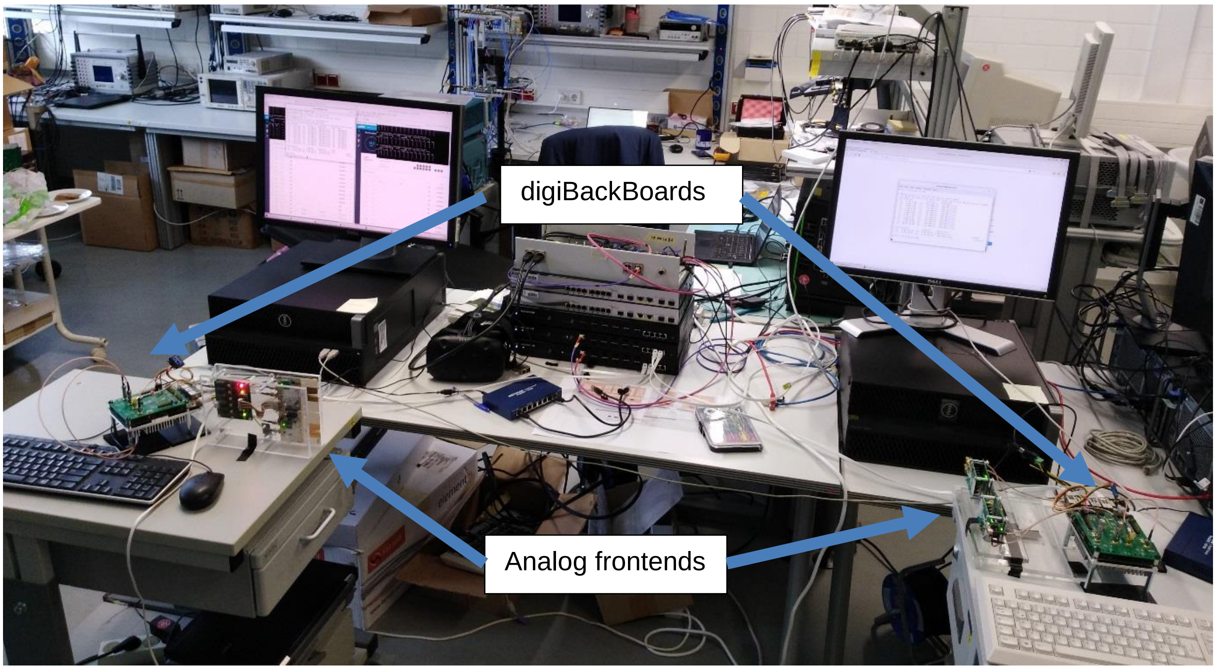

Figure 3 - 60 GHz measurement setup with IHP's frontends

In order to test the system, frames are constantly transmitted and received. The received frames are decoded and their content is inspected for errors.

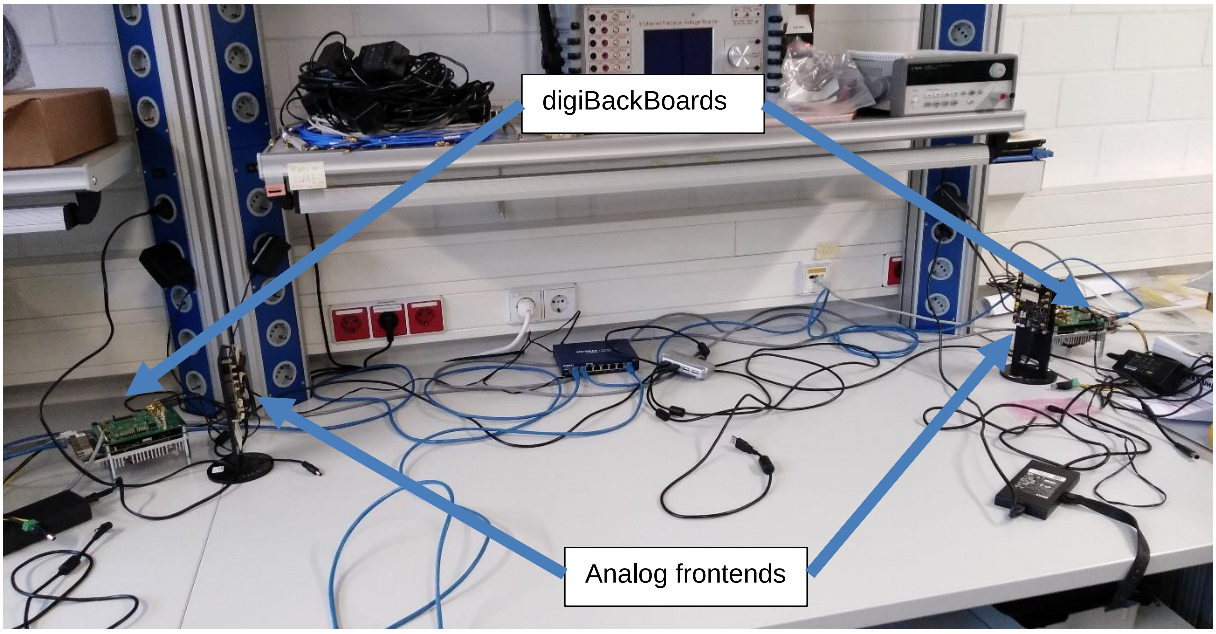

Figure 4 - 60 GHz measurement setup with Sivers IMA beam steering frontends

3.2. Performed tests and obtained results

The described system was tested for distances of up to 2 meters due to the space limitations. Nevertheless, the analog frontends used have significant output power and can be used up to 10 (or even more) meters without any issues.

Two tests were performed. One with 16 QAM transmission and 3/4 convolutional code leading to approximately 3.9 Gbps net data rate and the other with 64 QAM and 3/4 convolutional code leading to approximately 5.8 Gbps net data rate. In the both cases, the SNR was sufficient to avoid any frame errors. Hundreds of frames were exchanged for the test. The used Viterbi decodes was always able to correct the bit errors and no frame errors were detected. In order to estimate the “over the air” frame error rate much longer tests need to be performed.

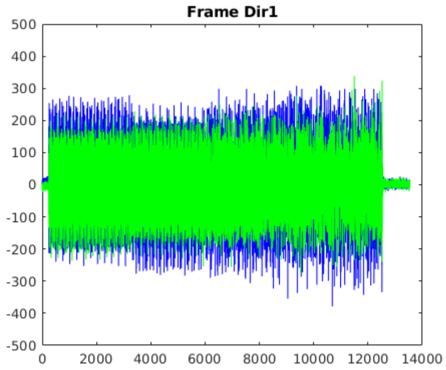

Figure 5 - In-phase and quadrature signals at the receiver

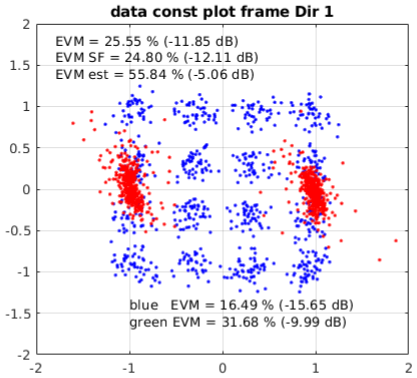

Figure 6 - 16 QAM constellation of the received signal

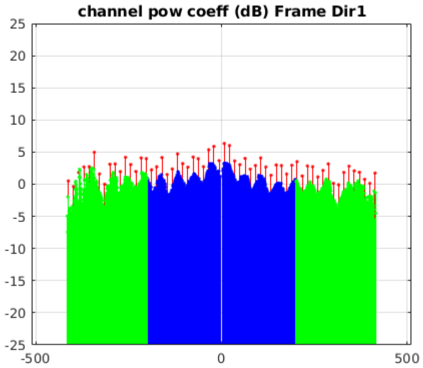

Figure 7 - Spectrum of the received signal. Red subcarriers are the pilots.

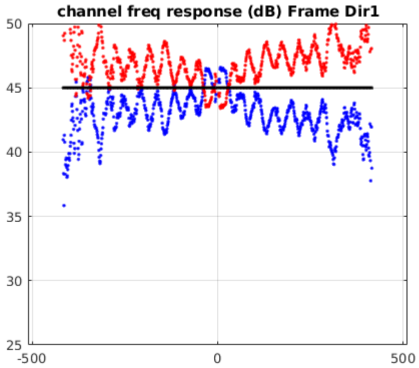

Figure 8 - The channel spectrum response (blue) and the correction coefficients (red) for zero forcing equalization

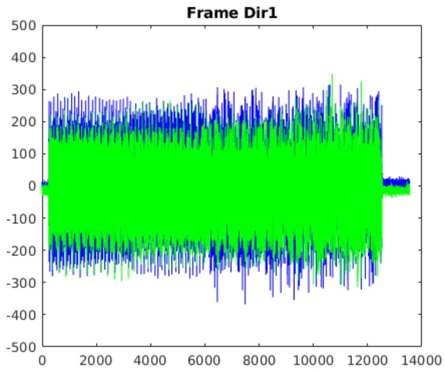

Figure 9 – In-phase and quadrature signals at the receiver

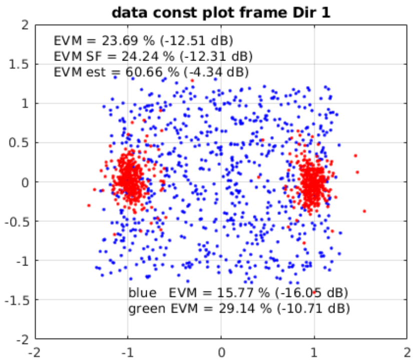

Figure 10 - 64 QAM constellation of the received signal

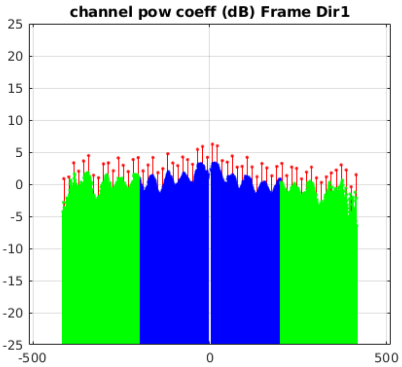

Figure 11 - Spectrum of the received signal. Red subcarriers are the pilots.

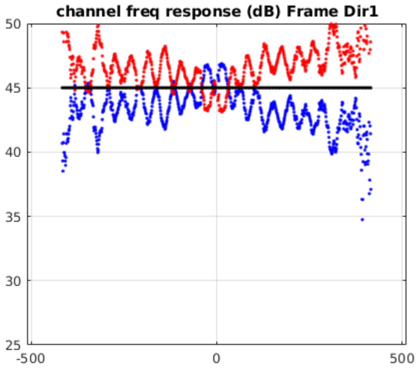

Figure 12 - The channal spectrum response (blue) and the correction coefficients (red) for zero forcing equalization

3.3. User Manual

The whole MATLAB model of the system comes in a bundle of multiple directories. These directories must be included in the MATLAB path recursively. After including them in the MATLAB path, the scripts for starting the simulations or for starting the hardware in the loop system are found in the directory “main”. In order to start the simulations, the script “AFETestFER.m” is started. The parameters of the system can be changed inside this file. The SNR can be changed, the frame length, channel model etc. All the parameters are in this file and are commented.

n order to test the hardware in the loop setup with the digiBackBoards, from the same directory, the script “test_setup_demod_loop.m” is started. All of the necessary settings can be accesed in the script. They are all commented and described. Before starting this test the both of the digiBackBoards must be connected to an Ethernet switch and also the computer running MATLAB must be connected to the same Ethernet switch. The computer must have an IP address 192.168.0.1.

4. Real-time multi-gigabit radio transmission prototype

The design has been prototyped in Xilinx Zynq XC7z100ffg900-2 FPGA. The platform used for the baseband/data link layer processing has also high speed dual AD and DA converters. They are used for generating and acquiring the in-phase/quadrature (IQ) baseband signal. This platform is called digiBackBoard. The platform also consist of a 4x 1Gb Ethernet ports and 2x SFP+ cages, compatible with 10 Gb Ethernet adapters. These Ethernet ports are used for transferring the data that needs to be sent/received over the 60 GHz wireless interface.

For transmission of data in the 60 GHz ISM band a few different solutions are available. At first the IHP’s 60 GHz analog frontend can be used. This frontend has PCB integrated antennas with an antenna gain of 9-10 dBi. Another option is to use commercial analog front-ends from Analog devices, Infineon or Sievers IMA. The first two analog frontends are used with external horn antennas having antenna gain of approx. 22 dBi. The third analog frontend has a phased array beam steering antenna. For the communication distances envisioned in the WORTECS project, all of the front-ends, the IHP's as well as the commercial are suitable.

4.1. Architecture of the FPGA design for 4 Gbps throughput – transmitter

In Figure 13 the block diagram of the transmitter is shown. The architecture is simpler compared to the receiver described later. Moreover, four of eleven blocks are shared with receiver and the complete transceiver has only a single instance of an ARQ engine, CRC Add and Check, Media Access Controller, and IFFT block. Sharing of the data link layer entities is needed due to the functionality purposes. All data link layer modules need to have access to the transmitter and receiver, in order to keep the ARQ state machines synchronized by exchanging the beacon frames and acknowledges. Sharing the FFT/IFFT is necessary due to limited DSP resources in the chosen FPGA device. This implies that the complete transceiver can process only a frame that is scheduled for transmission or a frame that has been received, but not the both at the same time. In this demonstrator, full-duplex operation is not possible, since the transmission and reception operate in the same wireless channel. In order to enable communication in the both directions, over the same channel a time division multiplexing (TDM) is used. Therefore, sharing the same FFT/IFFT core between the transmitter and the receiver does not limit in any way the functionality of the transceiver.

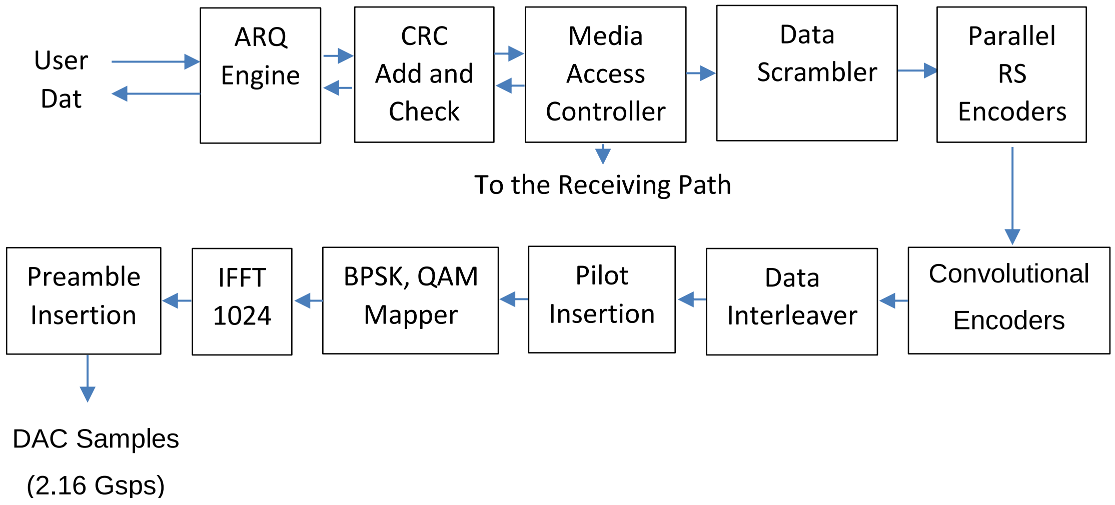

Figure 13 – Architecture of the implemented transmitter. The design includes a complete processing path with fully supported baseband and data link layer functionality

The functionality of data link layer modules (ARQ Engine, CRC Add and Check, Media Access Controller) is similar as in the case of the receiver. Their description is given in the next subsection. Baseband processing starts with data scrambling in order to avoid series 0's or 1's. Long sequences of 0's or 1's might lead to high power peaks in the OFDM waveform and should be avoided. The scrambler initial value is chosen randomly in order to generate a different pseudo-random sequence for each successive frame to be transmitted. This also applies for frames repeated by the ARQ module, meaning that all retransmissions of the same frames would show different waveforms. If a very high power peak has been generated at the initial transmission, there is still a chance that the retransmitted data will have more flat power distributions.

After the scrambler, the resulting data bits are processed by forward error correction (FEC) encoders. The use of outer Reed-Solomon codes is optional, but the inner convolutional coding is always applied. The operation and implementation of the interleaver are similar to the deinterleaver and are described in the next section. The amount of RAM required for interleaving is same as for deinterleaving. Therefore, it is important to design these modules with lowest possible overhead. In the other case, the wasted resources are doubled.

The interleaved data goes to the mapper, where the binary representation is translated into a QAM-16 constellation. Further, QAM symbols are placed at the data subcarriers. For the pilot tones, a pseudo-random sequence with BPSK encoding is used. The frame transmission starts with the preamble, followed by the IFFT-transformed signal field and user data. The preamble sequence is stored in read only memory (ROM). The transmitter reads it after activating the transmission mode and forwards it directly to the digital to analog converters (DACs).

4.2. Architecture of the FPGA design for 4 Gbps throughput - receiver

In general, the implemented receiver is an OFDM based processor that includes all digital blocks required for typical baseband and data link layer processing. It consists of 17 main processing blocks, as shown in Figure 14.

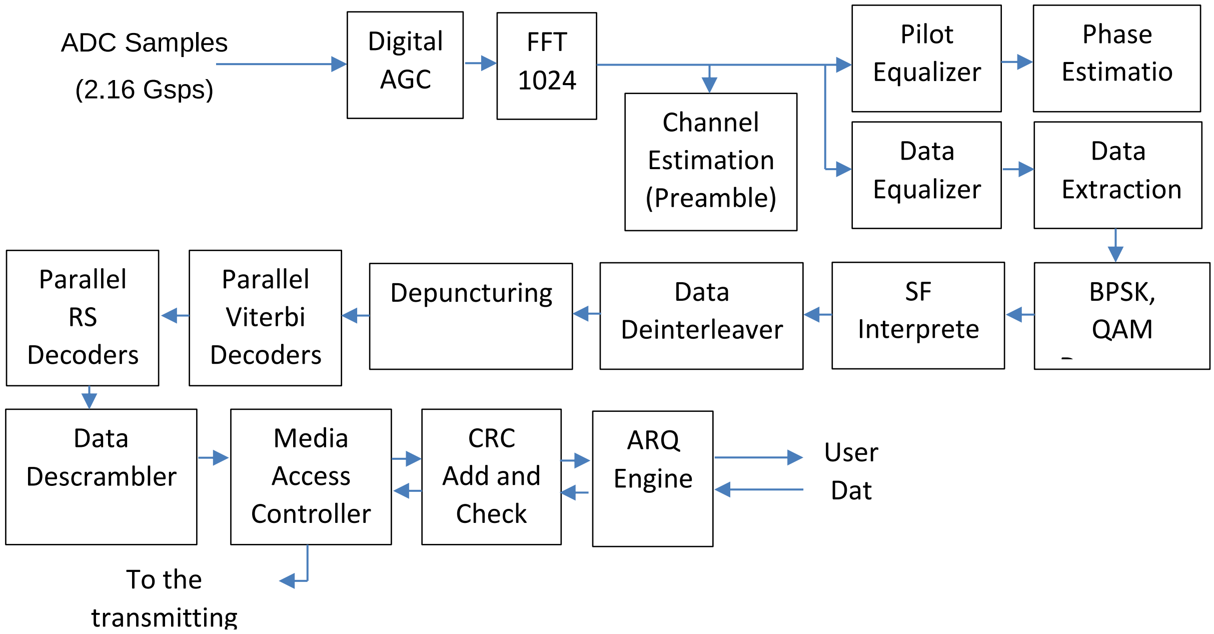

Figure 14 - Architecture of the implemented receiver. The design includes a complete processing path with fully supported baseband and data link layer functionality

The used hardware platform samples the incoming baseband signal with 2.16 Gsps dual-ADC and forwards the acquired IQ samples directly to the XC7z100ffg900-2 FPGA. The first processing block, automating gain controller (AGC), adjusts the amplitude in order to prevent signal clipping and to avoid excessive quantization noise in later processing. Briefly, it adjusts the amplitude in the supported dynamic range. It consists of a digital power detector, low-pass filter, and variable gain amplifier. For our implementation, a feed-forward topology has been chosen as shown in Figure 14.

The AGC sets the gain factor at the beginning of the preamble reception. As soon as the signal is sufficiently amplified, the AGC fixes the amplification factor to allow further channel estimation.

The next processing unit, FFT with 1024 points, converts the time domain signals to frequency domain. It is the main block of the OFDM architecture. Further, signal subcarriers are divided and processed fully separately and concurrently. In our case, 1024 subcarriers (tones) are used, but only 768 are allocated for data transmission. The remaining tones are reserved for pilots or are used as band guards and zero-carriers. This gives a maximal data throughput of 5.18 Gbps at the physical layer, assuming QAM-16 modulation, 474 ns FFT period, and cyclic prefix of 119 ns.

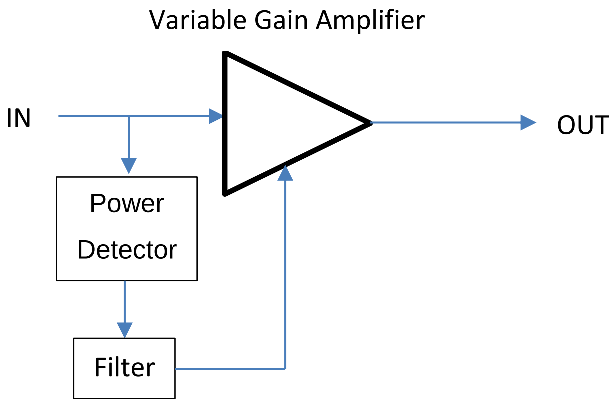

Figure 15 – Architecture of the implemented automatic gain controller

The channel estimation, equalization, and fine frame synchronization use four symbols of the second part of the preamble to tune the samples. The inverse channel coefficients are stored for later processing. Additionally, subcarrier power coefficients are estimated and forwarded to the QAM-16 demapper. The equalizer uses zero-forcing algorithm and compensates each subcarrier by the corresponding channel inversion. Pilot equalizer and phase estimator compensate phase and timing errors. An example of channel estimation performed in the real-time FPGA implementation is shown in Figure 16, while Figure 17 depicts the same estimation performed in Matlab, i.e. offline processing.

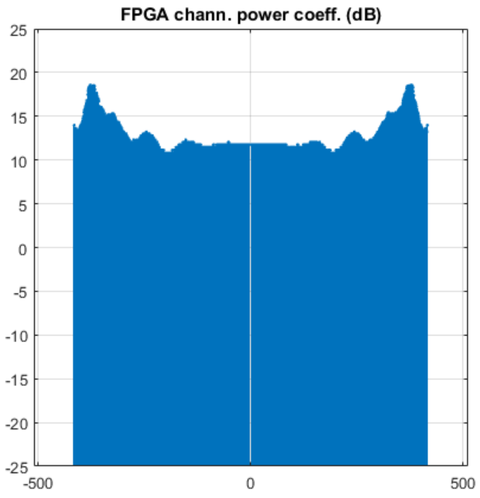

Figure 16 - An example of channel estimation performed in the real-time (FPGA)

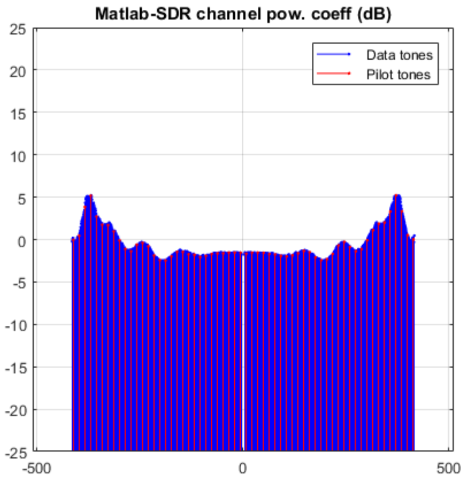

Figure 17 - An example of channel estimation performed in Matlab as off-line processing. The figure depicts positions of the pilot subcarriers additionally

After the fine synchronization and channel equalization take place, the tone samples are processed by the demapper and the resulting 5-bit soft values are computed. The processor estimates the bit values by log-likelihood ratio probabilities. The signal is demodulated and all further processing is performed on the obtained bit values. The tones and subcarriers are not needed any longer. Therefore, the demapper brings the processing in a new computation domain, where only bit representations are required.

Figure 18 and Figure 19 depict constellation diagrams recorded in FPGA (real-time processing) and MATLAB (offline processing), respectively. As compared in Figure 19, the FPGA implementation based on fixed-point arithmetic gives almost the same results as the MATLAB software model. Although the constellation can be easily decoded, we see some extensive noise. This noise comes from reflections occurring on cables between a transmitter and receiver. The PCB impedance matching has to be improved in this aspect in next release of the baseband signal processing board.

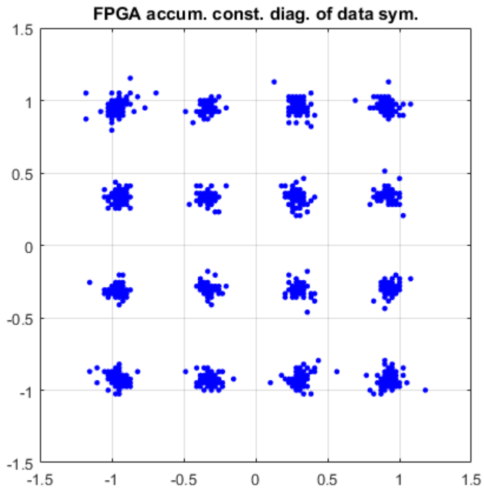

Figure 18 - Example of QAM-16 constellation recorded in real-time using the dedicated FPGA platform

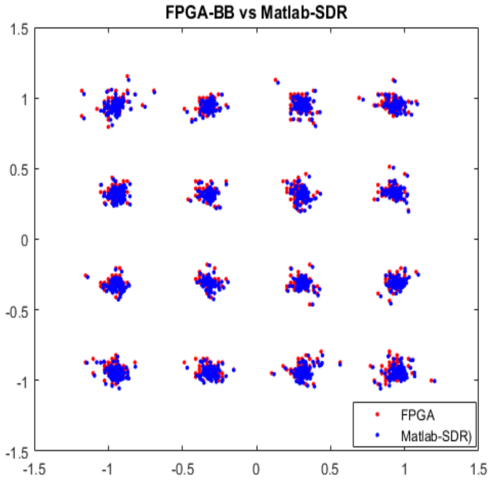

Figure 19 – Comparison of computation accuracy between FPGA VHDL implementation and Matlab software model

The signal field interpreter (SF-interpreter as shown in Figure 14) extracts frame length, data modulation, and data encoding information from the preamble. These settings are required to process the data according to the predefined modulation and coding schemes. Although we target to use QAM-16 modulation with FEC at code rate 3/4, the processor supports simpler modulation schemes (BPSK, QPSK) and lower code rate as well (1/2, 2/3). Thus, the block needs to set the data processing path before the frame is fetched into the FEC entities.

Data deinterleaver is required to spread long burst errors to single errors. It is necessary to avoid burst errors due to employed convolutional codes. The codes show very weak error correction performance when they are applied against errors longer than 1-bit. Thus, the Viterbi decoder is usually coupled with an interleaver. In our implementation, a memory-based approach has been selected. The interleaver alone uses 39% of the memory reserved for the whole baseband processor receiver. Thus, it is critical to design the interleaving and deinterleaving according to the employed hardware and targeted channel characteristic. We compared two commonly used interleaving architectures according to error correction performance and occupied FPGA area. The first architecture, convolutional interleavers (Figure 20), use flip-flops (FFs) and FPGA routing resources to perform bit mixing. It is also one of the most common architectures described in literature. Although the hardware is very simple, the structure leads to significant resource utilization and power consumption. As depicted in Figure 21, the number of flip-flops grows quickly with the interleaver size. For our demonstrator, we need 20 000 FFs to achieve the optimal results (Figure 22), and at least 10 000 FFs to achieve result comparable to the optimal (Figure 23). Although it is possible to fit such an interleaver to the targeted FPGA, we choose a RAM-memory based solution that uses Block RAMs and predefined read and write addressing patterns. In this way, we reduce switching power significantly, due to the reason that a single RAM cell is smaller than a single FF.

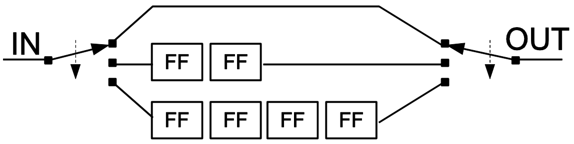

Figure 20 – Example of a convolutional interleaver with three rows and slope equal to two

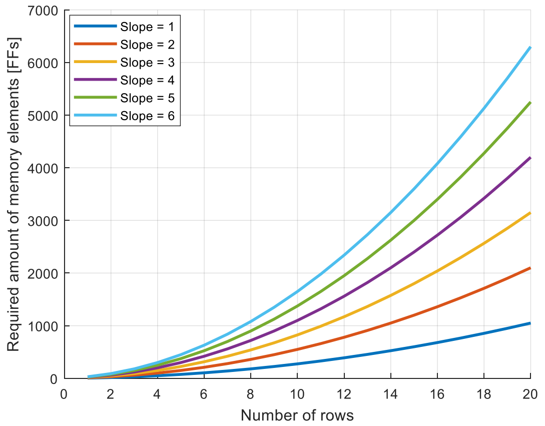

Figure 21 - The required amount of flip-flops (FF) needed for different types of convolutional interleavers

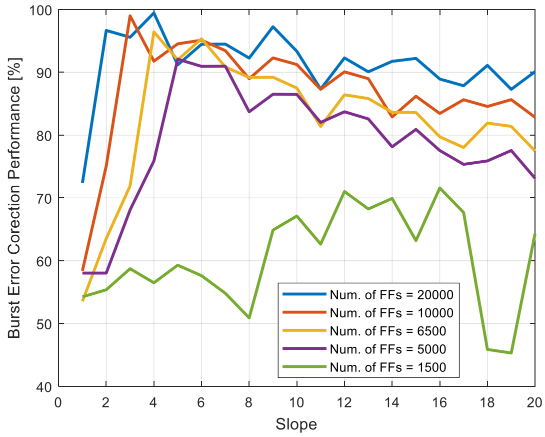

Figure 22 - Performance of different convolutional interleavers according to interleaver size and architecture

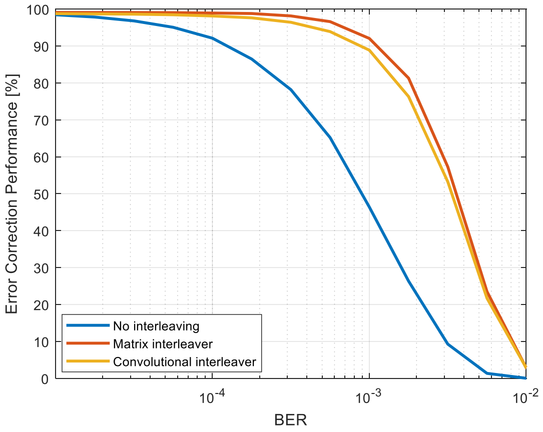

Figure 23 - Performance of convolutional and RAM-based interleavers for the implemented FPGA demonstrator

Puncturing and depuncturing entities are responsible for adapting the code rate and error correction performance according to channel quality. By default, the encoder produces data stream with code rate equal to 1/2. To achieve higher user data throughput, we remove predefined bits from the encoded stream before RF-transmission. On the receiver side, in the place of removed bits, zero elements are inserted before decoding. These zero values are natural for error correction processes and are omitted.

For FEC decoding, an array of 24 Viterbi decoders at 200 MHz is used. Thus, the processor delivers continuously up to 4.8 Gbps of user data. After Viterbi decoder, an optional outer FEC code is placed. In our case, we use Reed-Solomon at a code rate of 223/255. The outer decoder wipes all errors that were not corrected by the Viterbi entities. This operation is optional and can be enabled on demand by setting a dedicated bit in the signal filed.

The last step of baseband processing is data scrambling. The scrambler is an LFSR that produces a pseudo-random sequence that is XORed together with user data. This operation is needed in order to avoid long sequences of zeroes in the stream going to convolutional encoders. Figure 24 summarizes this section and shows the real timings of the baseband processor in the receive mode.

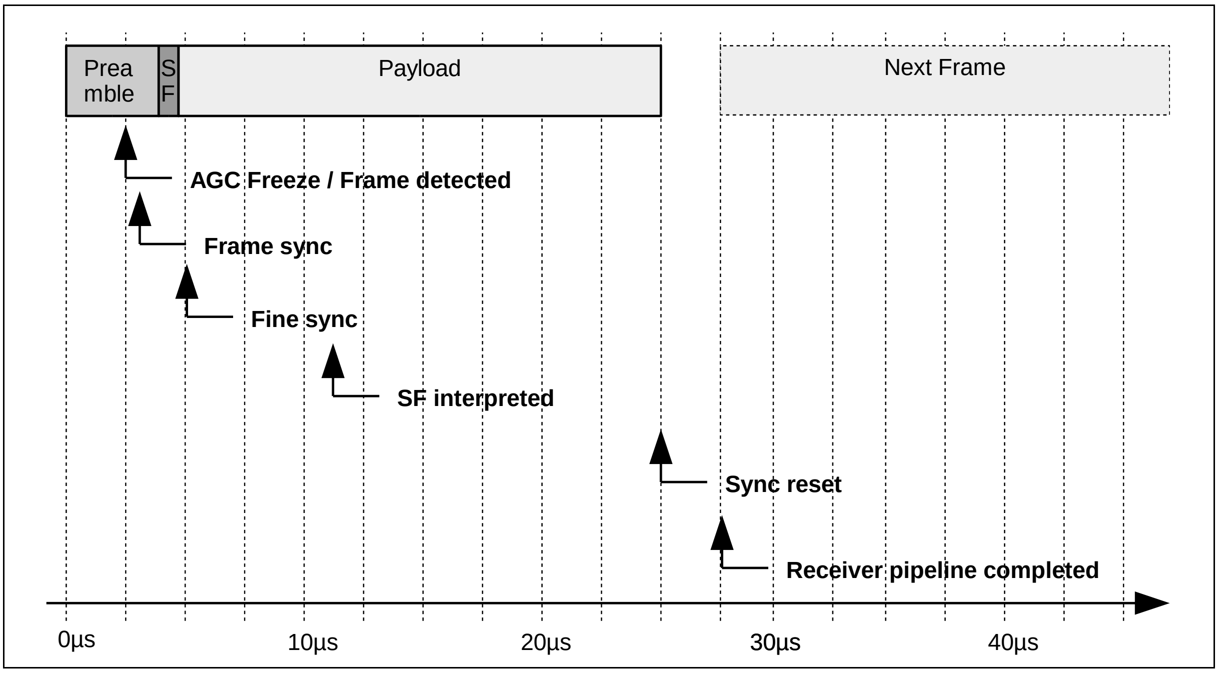

Figure 24 - Receiver scheduling with real timings

Finally, the baseband data is processed by data link layer modules. In our case, we distinguish three main data link layer entities: media access controller, CRC checker, and automatic repeat request module (ARQ). The access controller transmits periodical beacons and synchronizes the master with slaves. It is also responsible for collision avoidance. The next module, adds and recalculates CRC in frames. The ARQ module is relatively complex and performs three operations: data aggregation, lost frames repetition, and acknowledge generation. Frames aggregation has significant impact on the overall data rate of the demonstrator. Every frame sent by the transmitter is extended by a PHY-preamble. The preamble is an important part of a frame, but during preamble transmission, user data is not exchanged. Thus, preambles are reducing the effective good-put of communication systems. To mitigate this effect, frame aggregation technique is employed. The transmitter improves the overall throughput by merging user data and reducing the number of transmitted preambles. In our case, user data is aggregated up to 16 kB and transferred in a single PHY frame with one preamble only. The aggregation size defines the user data throughput as depicted in Figure 25.

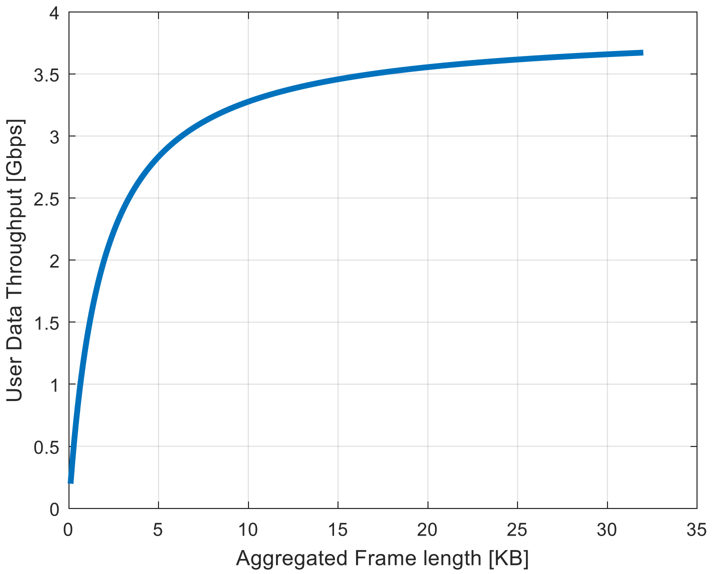

Figure 25 - User data throughput as a function of data link layer aggregation size for the implemented FPGA demonstrator at PER = 0

4.3. The most important parameters of the implemented transceiver

Table 2 summarizes the most important baseband parameters, Table 3 shows supported data rates, while Table 4 reports resource utilization of the selected XC7z100ffg900-2 FPGA.

Table 2 - PHY layer parameters

Channel bandwidth	2160 MHz
FFT bandwidth	2160 MHz
FFT size	1024 points
Subcarrier spacing	2.11 MHz
Guard interval	119 ns
FFT period	474 ns
OFDM symbol time	593 ns
Data sub-carriers	768
Pilot/zero carriers	60 / 5
Used bandwidth	1757.11 MHz

Table 3 - Supported data rates

Mode	User Data Rate [Mbps]	PHY Data Rate [Mbps]	FEC coding
BPSK-1/2	650 (CC) 568 (CC+RS)	1300	Convolutional 1/2 RS(255,223)
BPSK-2/3	860 (CC) 752 (CC+RS)	1300	Convolutional 2/3 RS(255,223)
QPSK-1/2	1300 (CC) 1136 (CC+RS)	2600	Convolutional 1/2 RS(255,223)
QPSK-2/3	1720 (CC) 1504 (CC+RS)	2600	Convolutional 2/3 RS(255,223)
16-QAM-1/2	2600 (CC) 2273 (CC+RS)	5200	Convolutional 1/2 RS(255,223)
16-QAM-2/3	3450 (CC) 3017 (CC+RS)	5200	Convolutional 2/3 RS(255,223)
16-QAM-3/4	3890 (CC) 3400 (CC+RS)	5200	Convolutional 3/4 RS(255,223)

Table 4 - XC7z100ffg900-2 resources utilization

Resource	Utilization	Available	Utilization [%]
LUT	191898	277400	69.18
LUTRAM	8682	108200	8.02
FF	171477	554800	30.91
BRAM	668	755	88.48

DSP	689	2020	34.11
IO	318	362	87.85
GT	1	16	6.25
BUFG	15	32	46.88
MMCM	1	8	12.5
PLL	1	8	12.5

4.4. User Manual

In order to test the boards they should be powered up as well as the analog frontends. The bords boot up and become ready for communication. The Ethernet packets arriving on the first Ethernet port of one board are wirelessly transferred on the others board first Ethernet port. No additional settings should be performed. The system is automatically configured and after the connection is established, a seamless link between the first Ethernet ports of the two systems is also established.

5. Heterogeneous networks (HetNet) prototype

In the first demonstrator we focused mainly on the data plane implementation. As already mentioned in the deliverable D3.3, our previous Layer2.5 software-based implementation in the OMEGA project, named the InterMAC, supported data rates up to 1.5 Gbps. This project, however, considers networks with data rates even 1 Tbps. Therefore, we need a new, ultra-fast Layer2.5 implementation based mainly on hardware.

In this project, we use the FPGA board VC709 of Xilinx, which includes four 10 Gbps SFP/SFP+ Ethernet ports. The data plane, presented in Figure 26, works like an intelligent switch in the following way: on receiving an Ethernet frame on any port, the data plane reads frame headers (Ethernet, IP and TCP/UDP headers). Then, it looks up the forwarding table to find the outgoing port for the received frame and passes the frame to this port.

By default, the data plane is configured to pass all incoming frames on port 0 to port 1, or to port2 in case port 1 is not working. Further, any frame received on ports 1 and 2 is passed to port 0. Therefore, in our demonstrator v1 the b-com server must be connected to port 0, and wireless transceivers to ports 1 and 2.

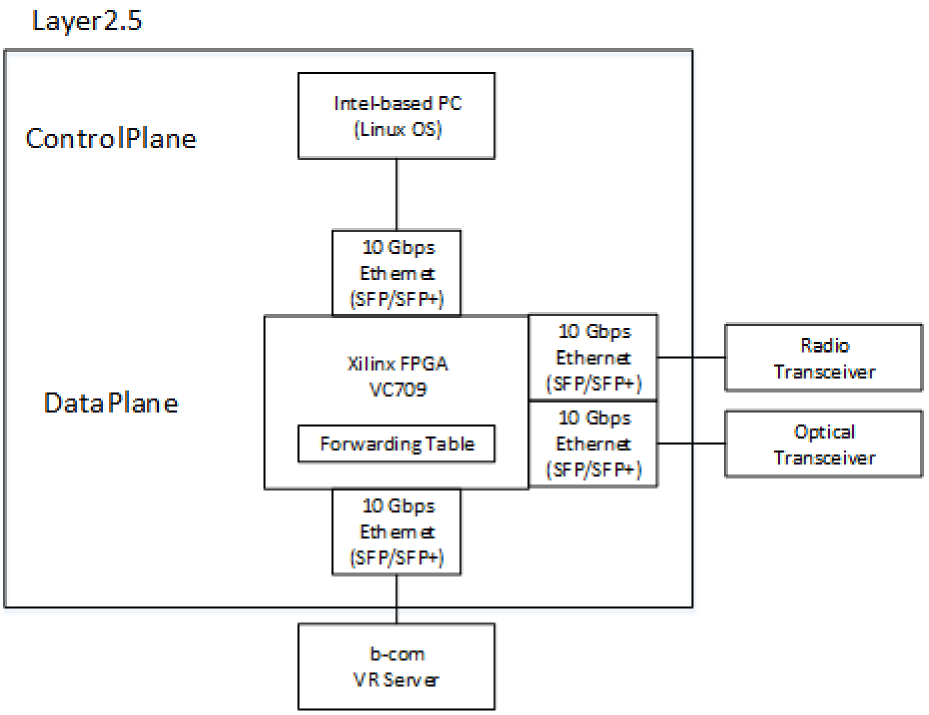

Figure 26 - We implemented the Layer2.5 data plane for the Xilinx VC709 FPGA platform

5.1. Graphical User Interface

For debug and presentation purposes we implemented a basic graphical user interface (GUI), depicted in Figure 27. At the top, there is a diagram showing the available network connections and their state, either working (a solid line) or not (red, dashed line). There are also two plots below the diagram. The first plot shows the current throughput for each connection, that is, the number of data transmitted. The bottom plot presents the delay for each connection estimated by sending and receiving probe frames.

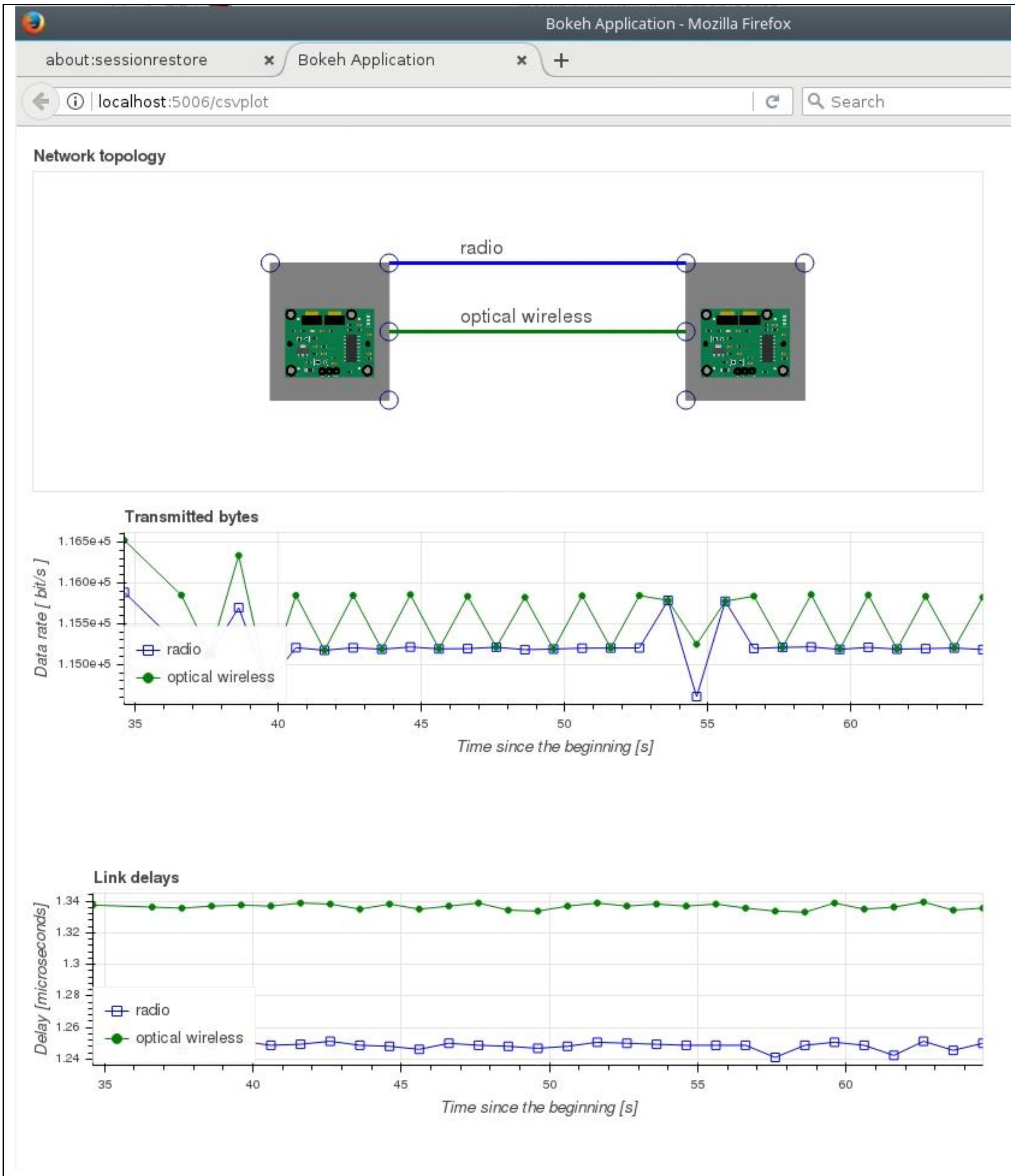

Figure 27 - Graphical User Interface shows links between HetNet switches, including current throughput and delays

We based GUI on webpages, which are displayed in a web browser on the end-user devices. The webpage and its plots are depicted in Figure 27.

To create the webpages to the end user, the web server gets periodically information about the network interfaces (network links) by sending query frames to the FPGA (see Figure 28). Then, the web server updates the web pages with plots. The complete web server is written in Python language with the bokeh framework. To start the web server and the program for getting data from the FPGA, two scripts from the directory ~/python/bokeh/csvplot/start_fpga_csv.sh must be executed (after changing the current directory to this one):

1. start_fpga_csv.sh
2. start_www.sh

After that, the end-user can connect to the web server and get the web page with plots.

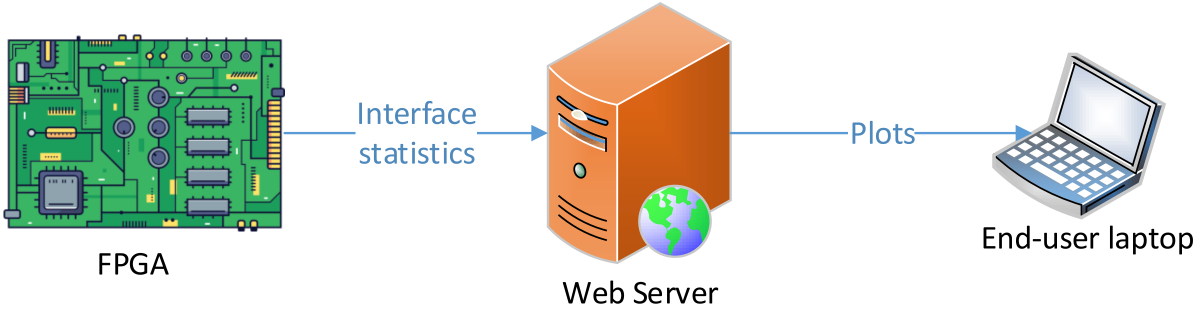

Figure 28 – Web server collects information about link statistics (throughput, data rate) from the FPGA and creates a webpage with plots to be displayed on end-user computers

5.2. Troubleshooting

After switching on the FPGA board, it needs a few seconds to start, mainly to get the image from the non-volatile memory. During this time all LEDs on the board are switched on. After the programming finishes, some LEDs start flashing. However, sometimes the FPGA has problems in getting the image and all LED are permanently powered on. In this case, the FPGA board must be started again (power down and up).

Each FPGA interfaces keeps sending probe frames, being Ethernet frames with the ethtype set to 0xff0 (hexadecimal notation). To check if the interface is working we can just start the Wireshark program on a laptop/computer connected to the interface of the FPGA and see if probe frames are received.

6. Artic Sea platform

6.1. System description

The content chosen for the evaluation of the WORTECS set-up is a multi-user interactive virtual environment named Virtual Arctic Expedition. The virtual experience allows up to 4 participants to explore the seabed of the artic regions in different epochs to observe the effects of the climate change on the arctic environment. Each participant is represented in the virtual environment via a simplified avatar.

The positions of the avatars, the movements of their hands as well as the rotation of their heads are obtained via the tracking information collected by the HTC vive LightHouses. Equipped with virtual reality headsets (HTC Vive) users will be able to interact naturally with the oceanographic environment thanks to the Vive controllers in order to obtain information on the different species, take underwater photos, or activate animations showing the evolution of marine currents over time (see).

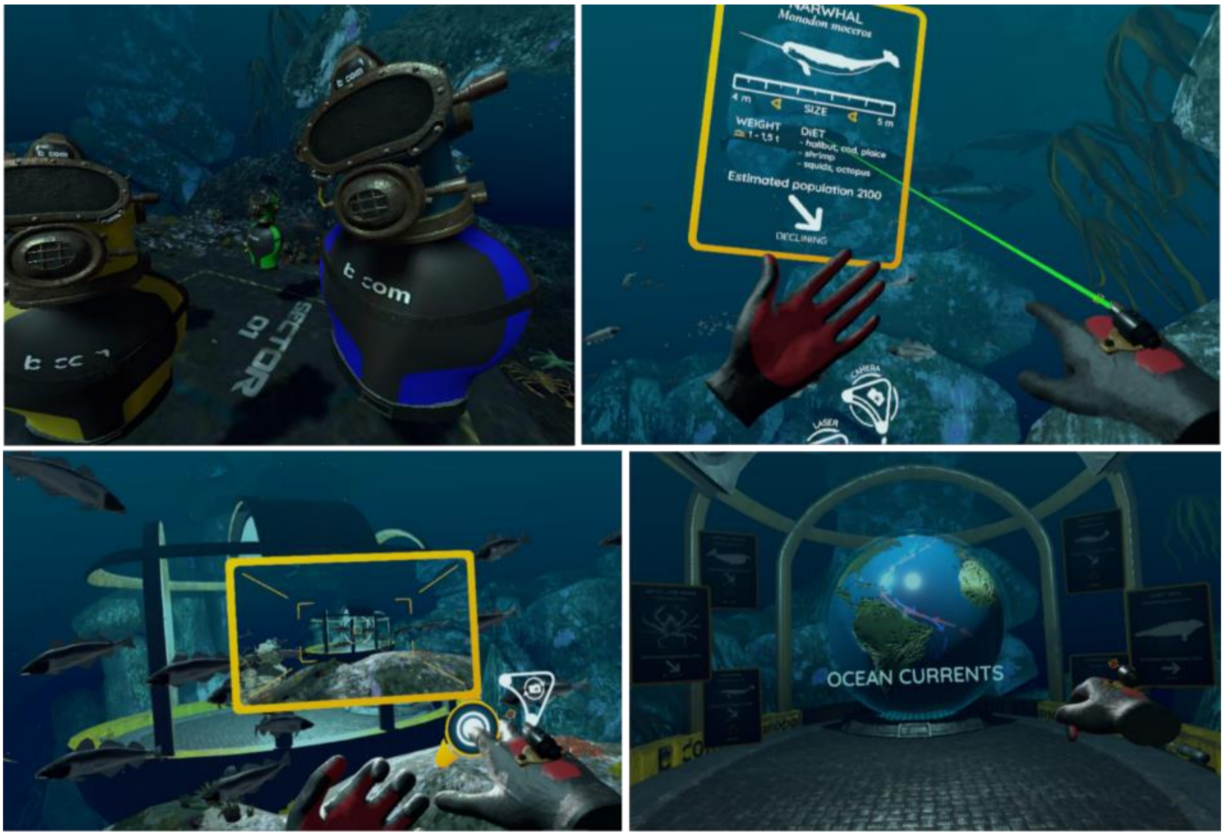

Figure 29: Multi-user immersive experience in Artic Ocean.

The hardware platform consists of a metal structure with a free internal area of 4m by 4m by 2,7m allowing to fix user's tracking systems as well as the different wireless hardware provided by the partners of the project (see Figure 30). 4 PCs equipped with graphics cards dedicated to virtual reality will run Virtual Artic clients and a server will synchronize each client to enable a collaborative immersive experience.

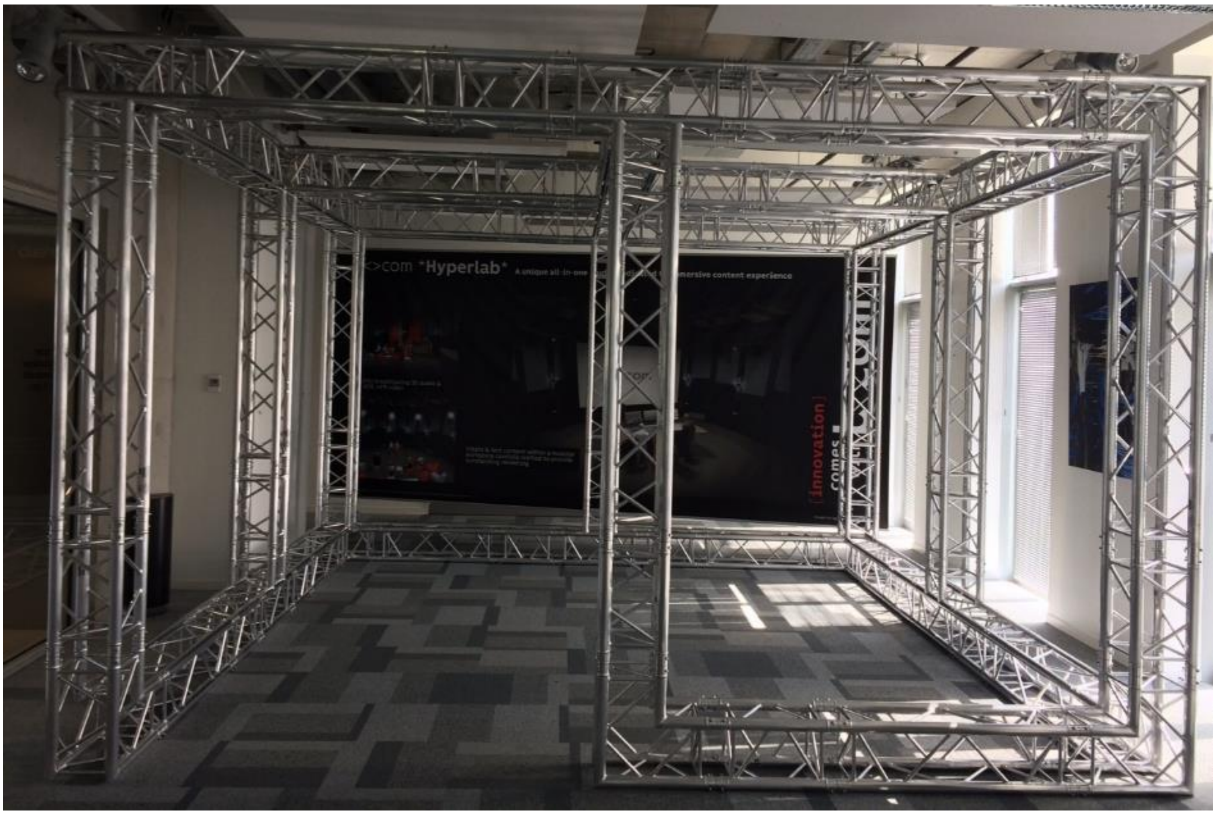

Figure 30: 4m x 4m x 2,7m metal structure for fixing VR and wireless hardware.

6.2. Test scenario description

The whole test scenario last between 15 and 20 minutes and is constituted of 4 different parts as described in the following subsections.

6.2.1. The underwater station

The first part is a short tutorial explaining the participants what they are going to experience and how they can interact with the environment. Participants can for instance point their laser to an animal to obtain further information about its specie or they can use a virtual camera to take pictures. This training takes place in a futuristic underwater station and ends with the opening of the metallic gate that allows the users to see the ocean surrounding the underwater station.

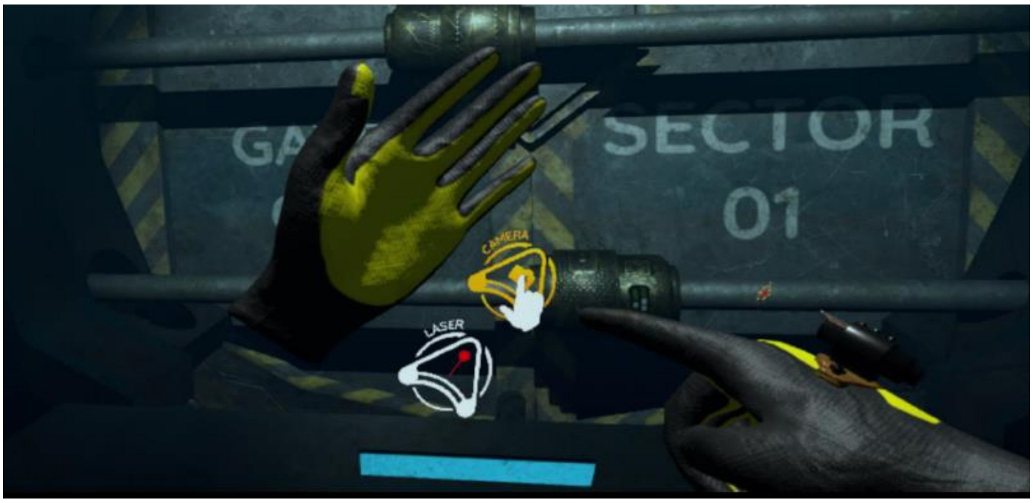

Figure 31: Participant testing the functionalities inside the ocean station

6.2.2. The arctic sea-bed in 1950

In the second part of the experience the participants are "teleported" outside the station and found themselves in the seabed 20 meters under the Arctic Ocean in the year 1950. A voice explains them where they are and gives information about the surrounding environment. In this part of the experience the participants watch and can partially interact with different spices like seals, narwhals and whales. The participants are invited to interact with the animals and point them with the laser to obtain complementary information about the species. This second part ends with the apparition of the "temporal navigation capsule" in the background.

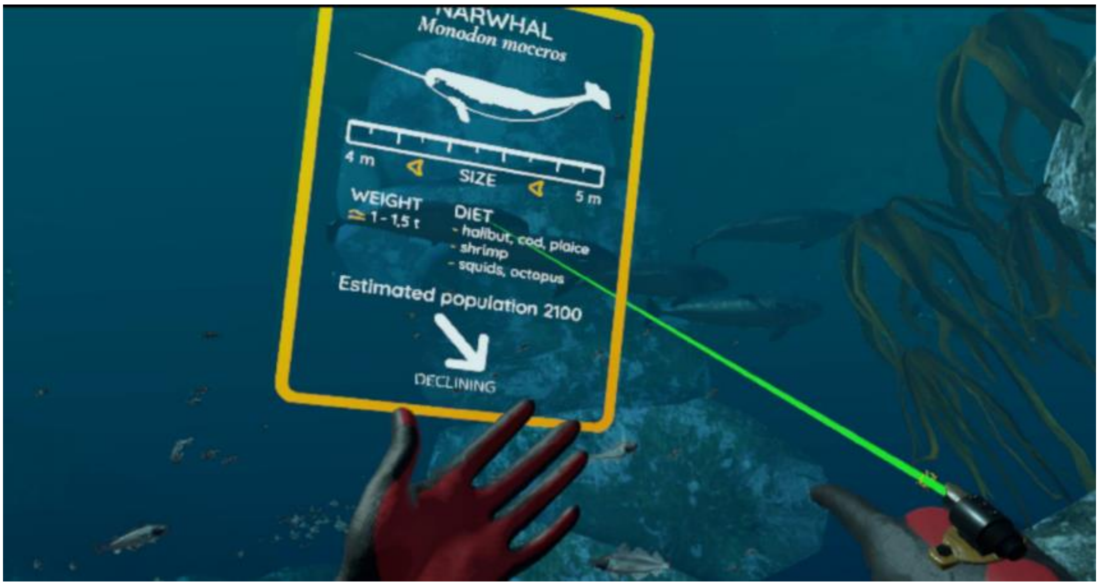

Figure 32: Participant in the 1950 pointing the laser at a narwhal to obtain more information

6.2.3. The temporal navigation capsule

In the third part the participants are teleported inside the "temporal navigation capsule". Here they assist to an animation explaining the evolution of the clime in the last years as well as a forecast of the consequences of the global warming in the artic region. Participants can obtain further information pointing the laser to the graphical representation of the species presents inside the "temporal navigation capsule".

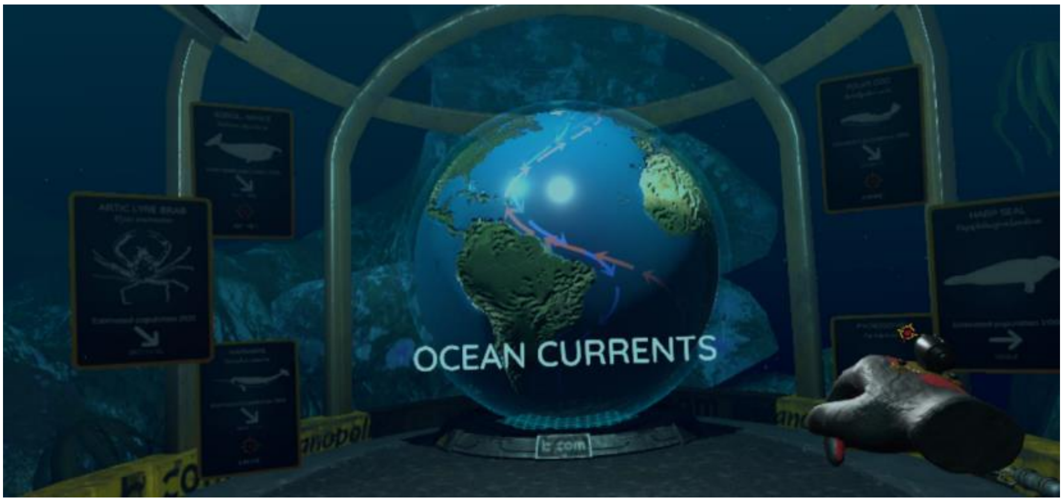

Figure 33: Presentation of the climate changes inside the temporal navigation capsule

6.2.4. The arctic sea-bed in 2100

In the fourth and last part the participants are teleported in the same underwater location they previously explored but this time in the year 2100. Like previously the participants are free to move around and explore the environment that now, due to the climate change, is populated by different species. During the experience the participants listen to a verbal explanation of the effects of the climate change on the arctic ecosystem.

Figure 34: Virtual screenshot of the sea-life in the arctic sea in 2100

The experience ends with the teleportation of the participants inside the underwater station where their virtual experience started. The participants are then informed that the experience has ended and they are invited to take off their HMD.

6.3. Assessment of the quality of experience for virtual reality contents

To evaluate the performance of the wireless tools developed in the WORTECS project, a virtual reality use case based on a dive into the Arctic seabed was chosen. Virtual Reality have become more and more popular at a consumer level in the last years mostly as a consequence of two factors: 1) the increasing computational power of the graphic cards needed to render stereoscopic content in real time and 2) the arrival in the market of various models of affordable Virtual Reality headset.

The adoption of a virtual reality use case as a testbed for the WORTECS technology is coherent with the market need. On one hand, resolution (around 4K) and frame rate (around 90Hz) of last displays embedded in virtual reality headsets continue to grow to provide user with a better experience. On the other hand, most companies are looking for a reliable wireless technology able to offer greater freedom of movement to users. Unfortunately, these two headset evolutions are contradictory since the current capacities of wireless transmission required to apply more complex video compression scheme offering a higher compression rate but generating a greater latency that does not guarantee an acceptable user comfort. Thus, the adoption of such displays request data transfer that go beyond the delivery capabilities of today’s commercial wireless solutions.

6.3.1. The evaluation of conventional videos vs the evaluation of virtual reality

The choice of using interactive virtual reality content instead of conventional video to assess the streaming capacity of the WORTECS solutions has various consequences on the evaluation methodology.

First, conventional videos have been used in the last years as a content to evaluate the efficiency of signal transfer. As a consequence the methodology to assess the quality of service as well as the quality of experience for this videos is well documented and some standardization organization like the ITU-T [ITU08] has proposed standard methods to perform such tests.

Part of the methods and tools preconized by the ITU-T will be included in the evaluation protocol adopted for the WORTECS project but, the majority of the ITU-T specifications are not directly applicable to virtual reality content. For instance the most of the proposed objective algorithms are specific to 2D non immersive videos and consequently they are not adapted to assess 3D stereoscopic interactive content.

Secondly, compared to conventional video content, virtual reality has some specificities that strongly influence the quality of experience. One of such peculiarity is known as immersion and can be defined as the level of control that the adopted interface has over the user perception [B07]).

While a conventional video is usually projected on a plane in front of the user, virtual reality is projected (via CAVE or HMD) all around the user. This fact have various implications, the first is that the human visual system is strongly impacted by the presentation device that covers a wider part of the user's field of view compared to the conventional screen (TV or cinema). This higher immersion could increase the user engagement but could as well be a source of discomfort inducing a phenomenon called Virtual Reality Sickness.

6.3.2. The impact of Virtual Reality Sickness in the quality of experience

While there is no complete agreement on the causes of the Virtual Reality Sickness, the most of the authors agree that the main cause is due to the detection of motion clues by the human visual system that are not consistent with equivalent stimulation of the vestibular system [MBS14].

The symptoms of Virtual reality sickness are mostly headaches, nausea and visual fatigue. The discomfort induced by the virtual reality sickness is often so intense that the participant decides to interrupt the virtual experience.

One of the recognized causes of virtual reality sickness is the reactivity of the system, as a consequence in the frame of the WORTECS user evaluation, a standardized questionnaire to assess the Virtual Reality Sickness like the SSQ [KLB93] will be included in the evaluation protocol. Such evaluation will allow to assess if the reactivity of the wireless transmissions will affect the user experience. The reactivity is directly linked to the system latency.

6.3.3. The impact of Presence in the quality of experience

Another peculiarity of the virtual reality that is often associated to the immersion is the feeling of presence that could be defined as the feeling of being there [S96]. The feeling of presence is often considered as an indicator of the quality of the virtual environment due to his multidimensional nature that covers various qualitative aspects of a virtual experience.

Some of such factors are:

- The ability of the system to react in a rapid and realistic ways to the actions performed by the participant;
- The coherence and synchronicity of the information (visual and auditory) that are presented to the user;
- The visual realism of the perceived media.

All these elements could be affected by a non-efficient transfer of information in the system. A failure in the data transfer would in fact induce artefact (like lags or drastic variations of the image frame rate) that could drastically impact the user experience.

For this reason a questionnaire to evaluate the perceived presence like the Presence Questionnaire [WS94] or the IPQ [SFR01] will be included in the evaluation protocol.

6.4. User Manual

To allow the user location, the setup of Lighthouse modules is required. Two of them would be set on two opposite corners of the ceiling in the smart cave. At 2.7m distance high, they would have a long range and reach all users whatever their position. They are connected to the power supply using the DC adapter provided by the manufacturer. To avoid interference the first one is configured on channel “a”, and the second one on channel “b”.

The Virtual Arctic Expedition application has been compiled in order to deliver a self-contain executable. Once the application is copied on the HDD of a Virtual Reality ready computer, (64GB RAM, Core i7 processor, GTX 980ti GPU), simply double click on the desktop short cut named “VirtualArcticExpedition”. The application would automatically start the StemVR environment, then detect the HMD and launch the starting point expedition pictures.

When the user is ready, simply click on the “a” keyboard touch and follow the audio instruction.

7. Video converter platform

The video converter has been developed to create a bridge between a video environment, and an internet one.

In one hand, signal from a video source respects HDMI or Display Port standards. These signals are based on a continuous transmission, from one point (source) to another one (destination).

In the other hand, signal from an internet object respects IP (Internet protocol) standard. These signals are transmitted in burst, with additional information that enrich the signal packet with data like: from where they are coming from, where they are going to, their size and so on.

These two standards are not compatible at all, and a conversion from one world to the other one is required to interface the video source with the IP switch or access point.

7.1. System description

The converter has to work in both side, from HDMI to IP on the access point side, and from IP to HDMI on the user equipment side.

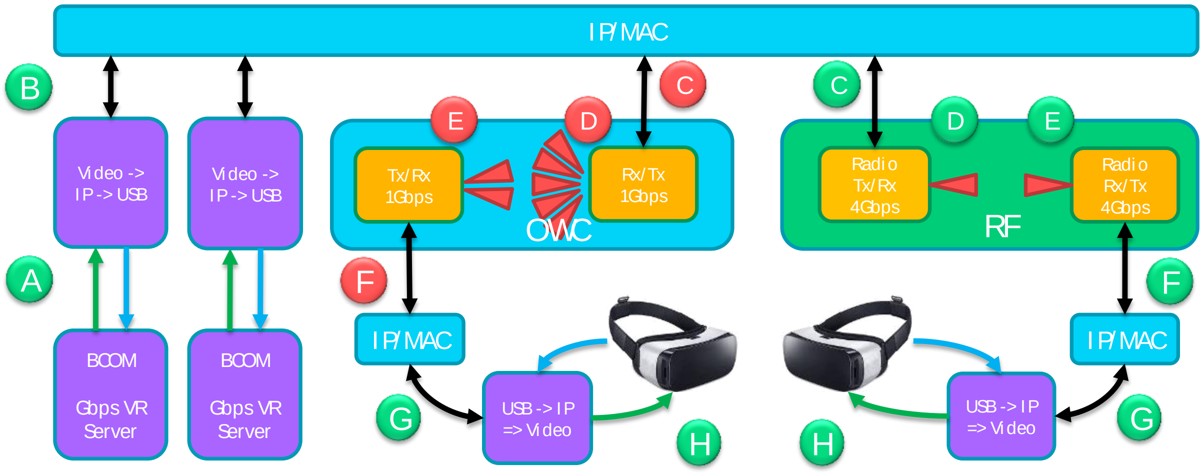

Figure 35: Context that present the video converter location

As described in Figure 35, the conversion operation should be realized on both sides:

- From video to IP : from the video server to the IP/MAC layer;
- From IP to Video : from the IP/MAC layer to the Head Mounted Display.

Furthermore, all these operation should be realized with a short latency keeping the overall latency smaller than 3ms.

As illustrated on Figure 36, the implemented function are on one side:

1. Extract video data from HDMI standard;.
2. Compressing the video if required, depending on the available access point bandwidth;.
3. Build packet following 10G eth IP standard..

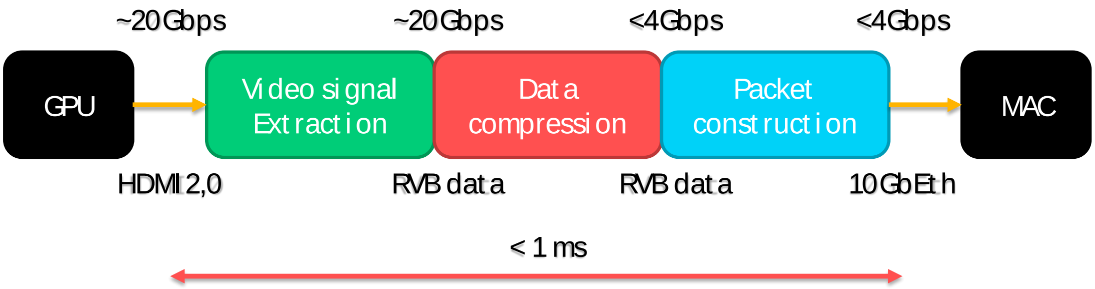

Figure 36 – Access point side main function presentation

And on the other side, the dual functionalities are implemented (Figure 37):

1. Accept packet following 10G eth IP standard;

2. Uncompressing the video if required, depending on compression applied onto the video;.
3. Build video signal following the HDMI standard..

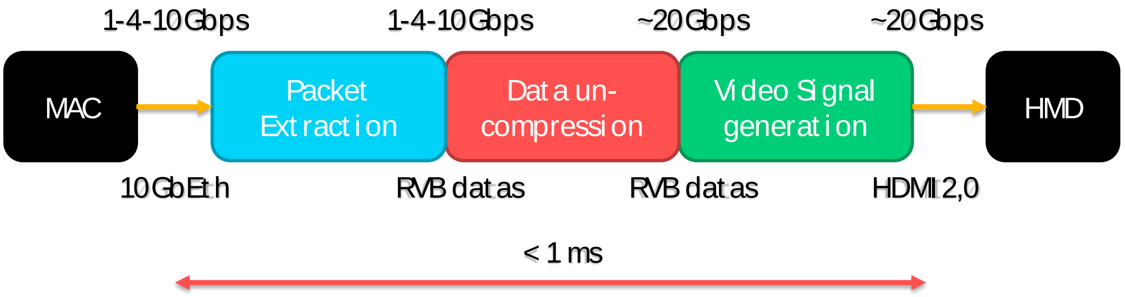

Figure 37 – User side video converter main function presentation

Due to the huge amount of data to process and the short allowed latency, the video conversion functions are hosted in an FPGA board. The following Figure 38 describes the signal round trip through each components:

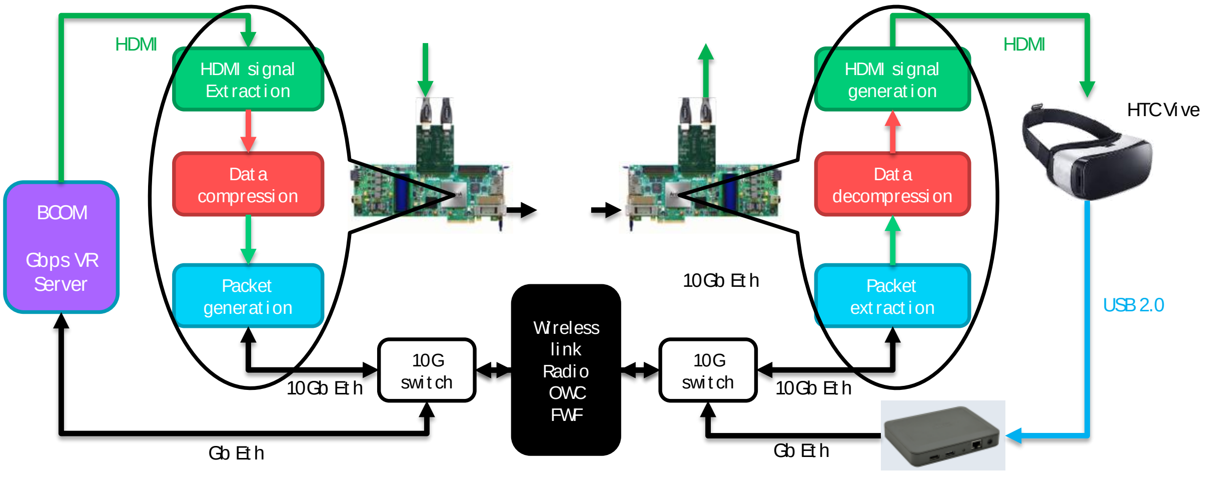

Figure 38 – FPGA implementation of main function on AP and UE side

The HDMI function comes from the FPGA vendor. It supports HDMI rev2.0 standard and then is able to transmit up to 18Gbps. In our case, the special VR resolution of the HMD HTC Vive (2160x1200@90Hz) drive to a throughput of 5.57Gbps.

The compression / decompression function comes from IntoPix, with a Tico encoder. A custom IP has been generated to deals the VR resolution. It allows a compression ratio from 6.8 (3.5bpp/24bpp) to 2.4 (24bpp/10bpp), and it provides compressed and un-compressed video content keeping latency smaller than 300us.

Video data are packetized respecting the RDD35 standard before feeding the 10G MAC.

The 10G function comes from the FPGA vendor and is configured to support optical transmission.

All these elements will be integrated in a 2U Rack for easier utilization.

7.2. Test scenario description

To check the functionality of the video converter, we need a VR video server, a video converter on the access point side, an LC/LC optical duplex fibre, a video converter on the UE side and a HMD.

The test consists in connecting the two video converters together with the LC optical fibre, then feed the AP side converter with the video generated by the VR server, and drive the HMD with the other converter.

The VR server should detect the HMD, get its position and generate the associated video. The HMD should receive the video and displays it smoothly.

The user should not suffer from seasickness thanks to the very short latency added in the loop HMD => VR Server => HMD.

Additional measure could be realized to quantify the added latency, the resulting video throughput and the amount of packets loss on the time going.

7.2.1. Uplink Latency

The uplink latency is the time lost from HMD out to VR server input. It will be realized with oscilloscope and electric signal routed onto dedicated pin on the FPGA board.

7.2.1. Downlink latency

The downlink latency is the time lost from HDMI Rx in (on the AP side) to HDMI Tx out (on the UE side). As previously, it will be realized with oscilloscope and electric signal routed onto dedicated pin on the FPGA board.

7.2.1. Compression ratio

The compression ratio is the video throughput reduction that is applied onto the incoming video signal. This value can be measured as the ratio between video bitrate that enters the encoder over the video bit rate that going out.

7.2.1. Packet loss or error

Packet loss or packet errors creation can append along the transmission chain. These loss/error measure should be zero if we use a perfect transmission link (optical fibre), and this whatever the video content or the user location. Dedicated counter inside the FPGA could manage the amount of packets loss and or error.

7.3. User Manual

The video converters are packaged in standard 2U rack. They are working by pairs. The first one is suited on the access point side. It converts HDMI standard into 10Geth one. The second one is suited on the UE side, on the backback bag and converts 10Geth standard into HDMI one.

Both equipment contains all the hardware and software required to do the conversion, and are ready to use just after the booting phase.

The connections:

Access Point video converter:

- Connect eth to PC
- Connect HDMI to PC
- Connect LC fiber to the RF modem
- Connect main power
- Power on PC and start the VR application

User Equipment video converter:

- Connect HDMI to HMD
- Connect power to HMD
- Connect USB to HMD
- Connect LC fiber to the RF modem
- Connect main power

Conclusion

In this deliverable we discussed the first radio demonstrator. We showed that data rate of almost 8 Gbps are possible in the 60 GHz band using a channel bandwidth of about 2 GHz. Additionally, the HetNets for supporting the radio demonstrator were discussed. At the end, the Arctic Sea VR platform is described. The radio system would be deployed in this platform and should be at the end consisted of one or more access points supporting multiple users.

References

[WD23] WORTECS Deliverable 2.3 – Focus on Virtual Reality

[IHP] www.ihp-microelectronics.com

[VIT05] Francis, Michael. "Viterbi Decoder Block Decoding-Trellis Termination and Tail Biting." Xilinx XAPP551 v2. 0, DD (2005): 1-21.

[DIGI19] M. Petri and M. Ehrig, "A SoC-based SDR platform for ultra-high data rate broadband communication, radar and localization systems," 2019 Wireless Days (WD), Manchester, United Kingdom, 2019, pp. 1-4.

[CUE18] Cuervo, Eduardo, Krishna Chintalapudi, and Manikanta Kotaru. "Creating the perfect illusion: What will it take to create life-like virtual reality headsets?." Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications. ACM, 2018.

[SIVERS] www.siversima.com

[B07]Burkhardt, J.-M. (2007). Immersion, représentation et coopération: Discussion et perspectives de recherches empiriques pour l'ergonomie cognitive de la réalité virtuelle. Intellectica, 45(1), 59–87. Retrieved from http://csaweb105v.csa.com.proxy.lib.wayne.edu/ids70/view_record.php?id=3&recnum=195&log=from_res&SID=nh6ff5fhjt3jpddet4pl1a7pv3

[ITU08] ITU-T. (2008). Subjective video quality assessment methods for multimedia applications. Retrieved from https://www.itu.int/rec/T-REC-P.910-200804-I/en

[KLB93] Kennedy, R. S., Lane, N. E., Berbaum, K. S., & Lilienthal, M. G. (1993). Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness. The International Journal of Aviation Psychology. https://doi.org/10.1207/s15327108ijap0303_3

[MBS14] Malik SH. Blake H. Suggs LS. (2014). A systematic review of Cybersickness. British Journal of Health Psychology, 19, 149–180. https://doi.org/10.1145/2677758.2677780

[S93] Sheridan, T. B. (1996). Further Musings on the Psychophysics of Presence. Presence: Teleoperators & Virtual Environments, 5(2), 241–246. https://doi.org/10.1109/ICSMC.1994.399986

[SFR01] Schubert, T., Friedmann, F., & Regenbrecht, H. (2001). The Experience of Presence: Factor Analytic Insights. Presence: Teleoperators and Virtual Environments, 10(3), 266–281. https://doi.org/10.1162/105474601300343603

[WS94]Witmer, B. G., & Singer, M. J. (1994). Presence Questionnaire. https://doi.org/10.4324/9780203871539