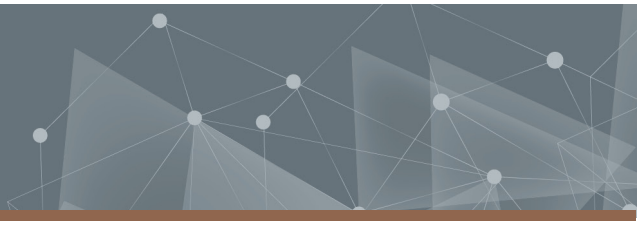
CHALMERS
UNIVERSITY OF TECHNOLOGY



Design and Evaluation of a WiFi-based Communication System for Industrial Automation Applications

Master's thesis in MPCOM

Pan Hu

DEPARTMENT OF Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Design and Evaluation of a WiFi-based Communication System on SMGs based on SDR

Keywords: Wi-Fi based, SMG(Slotted Microwave Guide), SDR(Software-define radio),

Pan Hu

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2023

Design and Evaluation of a WiFi-based Communication System on SMGs based on Software-defined Radio
Pan Hu

Advisor: Jafar Banar, PHD student in Communication Systems Research Group, Electrical Engineering Department, Chalmers University of Technology
Supervisor: Peter Kohlschmidt, Manager in Strategic and Operational Management Department, VAHLE Group
Supervisor: James Gross, Professor of Electrical Engineering & Computer Science Department, Royal Institute of Technology (KTH)
Examiner: Fredrik Brännström, Professor and Head of Communication Systems Group, Chalmers University of Technology

Master's Thesis 2023
Department of Electrical Engineering
Division of Communication Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Two USRP E320 SDR devices will be used to establish a stable 5GHz connection based on the 802.11n (WiFi4) amendment. With customized industrial 5GHz antennas on, They will be further utilized to design and evaluate a wireless communication system on slotted microwave guides for industrial purposes.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Design and Evaluation of a WiFi-based Communication System on SMGs
Pan Hu
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Over the last decade, Industry 4.0 has become a leading paradigm in industrial automation. However, the increasing mobility of automatic robots and devices, as well as flexibility in installation mandates more and more to replace cables by wireless communication systems. In the past, several such solutions have been proposed, among them 5G Rel16 being the most important one addressing ultra-reliable low-latency communication (URLLC) features. Nevertheless, alternative solutions are of interest due to complexity and cost reasons. In collaboration with the industrial partner VAHLE (www.vahle.com) and under supervision of KTH stockholm, by conducting on-site measurements on the slotted microwave guides provided by VAHLE using our software-define radio (SDR) setup with well-designed automation scripts, taking aspects of layer 1,2 and 3, our objective is to gather valuable data by conducting measurements on the slotted microwave guide (SMG) for VAHLE in order to explore a URLLC solution that meets well-defined requirements regarding the round-trip delay, packet loss rates, payload size, and data throughput.

Keywords: Industry 4.0, ultra-reliable low-latency communication, micro slotted guides, sit, software-defined radio, round-trip latency, packet loss rate, automation scripts.

Acknowledgements

Throughout the writing of this dissertation I have received a great deal of support and assistance.

I would first like to thank my supervisor, James Gross, whose expertise was invaluable in formulating the research questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I would also like to thank our PHD group in Information Science and Engineering (ISE) division at KTH, especially Samie Mostafavi, who offered me valuable guidance on the SDR tools that I needed to choose the right direction and successfully complete my dissertation.

In addition, special thanks to my friend Xiaoman Chu, who provided stimulating discussions as well as happy distractions to rest my mind outside of my research.

Pan Hu, Gothenburg, July 2023

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

SDR	Software-Defined Radio
USRP	Universal Software Radio Peripheral
SMG	Slotted Microwave Guide
SMGM	Slotted Microwave Guide Middle Size
SMGX	Slotted Microwave Guide Extra Size
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
3GPP	Third Generation Partnership Project
IRTT	Isochronous Round-Trip Tester
URLLC	Ultra-reliable and Low-latency Communication
eURLLC	Enhanced Ultra-reliable and Low-latency Communication
UHD	USRP Hardware Driver
API	Application Programming Interface
RFNoC	Radion Front Network-on-Chip
FPGA	Field Programmable Gate Arrays
GNU	Unmanned Aerial Vehicle
MIMO	Multiple Input Mutilple Output
SoC	System On Chip
NIC	Network Interface Card
PL	Processing System
PS	Programmable Logic
RPU	Reconfigurable Processing Unit
CPU	Central Processing Unit
MCS	Modulation and Coding Scheme
PER	Packet Error Rate

Contents

List of Figures

List of Tables

1
Introduction

The world has been witnessing the fourth industrial revolution and the digital transformation of the business world, commonly referred to as Industry 4.0, at the dawn of the 21st century. The fourth industrial revolution has been proved not just hype, but a hit, according to several studies [1]. With the prosperity of Industry 4.0, industrial automation is receiving more and more attention, especially in the wireless real-time communication field. Packet delay, error rates, data throughput, and other relevant performance metrics in the field of wireless communication, in the meanwhile, are becoming more and more of interest from both the perspective of academic research and revenue generation for enterprises.

Several solutions have been proposed to enhance wireless communication systems in terms of performance regarding the latency and reliability of the packet. Among them, 5G Rel 16 was proposed as the most influential one in the mid-2020s in the field of mobile communication. Enhancements to the 5G eURLLC (Enhanced Ultra-Reliable and Low-Latency Communication) were proposed in the 5G Rel 16 by 3GPP to deliver better reliability along with reducing the latency as much as possible. After the release of 5G Rel 16, several methods of enabling URLLC (Ultra-Reliable and Low-Latency Communication) were successively proposed in Wi-Fi systems, providing a more affordable, easier-to-operate, and more suitable solution for limited space arrangements.

In this section, the definitions of URLLC in 2 different contexts, one in the mobile communication field defined by 3GPP, and the other one in the industrial automation field defined by the master thesis student together with the industry partner VAHLE Corporation, will be clarified in detail. Following that, the project objective, along with the challenges encountered throughout its execution, will be outlined.

1.1 URLLC in different Contexts

1.1.1 5G NR — 3GPP Context

URLLC, an abbreviation for Ultra-Reliable Low-Latency Communication, was initially introduced by 3GPP. The 3rd Generation Partnership Project (3GPP) is a global collaboration between telecommunications standards organizations that play a pivotal role in developing technical specifications for mobile communication systems, including 5G and future generations.

URLLC stands as a prominent use case that 3GPP has placed at the forefront of their

efforts to advance 5G and future iterations of mobile networks. Its primary objective is to deliver communication services of exceptional reliability and minimal latency, catering specifically to critical applications that demand real-time responsiveness and ultra-high reliability, such as industrial automation, autonomous driving, and remote surgery. In addition to URLLC, 3GPP has also taken into account two other service categories, namely eMBB (enhanced mobile broadband) and mMTC (massive machine-type communications), during the design of 5G New Radio (NR).

Extensive research has been conducted within the Service and System Aspects (SA) working group of 3GPP to comprehensively examine the significant use cases and applications from various vertical domains. Among these, future factory applications, distributed utility grid protection, and autonomous driving are widely recognized as the most critical and prominent use cases that demand ultra-low latency and exceptionally high reliability [2]. Table 1.1 presents an illustrative representation of the Key Performance Indicators (KPIs) for these use cases. Significantly, the table focuses on end-to-end latency and reliability rate as fundamental metrics, while excluding other communication aspects, such as data rate, payload size, and IPDV (Inter-Packet Delay Variation). This selection reflects our emphasis on these two key parameters, which are of paramount importance in our research.

Scenario	**End-to-End Latency**	**Reliability Rate**
Discrete automation - motion control	1ms	99.9999%
Electricity distribution - high voltage	5ms	99.9999%
Remote control	5ms	99.999%
Doscrete automation	10ms	99.99%
Intelligent transport systems - infrastructure backhaul	10ms	99.9999%
Process automation - remote control	50ms	99.9999%
Process automation - monitoring	50ms	99.9%
Electricity distribution - medium voltage	25ms	99.9%

Table 1.1: An example of latency and reliability rate for URLLC use cases [2]

1.1.2 VAHLE Context — Industry Context

With a preliminary understanding of the 3GPP's definition of URLLC, our attention now turns to its implications for VAHLE, our industrial partner. This section will delve into three pivotal aspects, highlighting the significance of URLLC in the context of VAHLE. Table 1.2 presents a comprehensive overview of the three fundamental distinctions between the 5G URLLC System and the VAHLE-Specific System. In the subsequent sections, we will provide a comprehensive analysis of

each of these three distinctions.

	VAHLE-Specific System	**5G URLLC System**
Wireless Channel	SMG	Air
Transmission System	Wi-Fi based (802.11)	5G NR (New Radio)
Application Requirement	Low Latency Highly Reliable	Latency of 1ms Reliability of 99.9999%

Table 1.2: Comparisons between VAHLE-Specific system and 5G URLLC System

1.1.2.1 Wireless Channel

In contrast to the wireless channel over the air in the Fifth Generation (5G) cellular network, this project utilizes Slotted Microwave Guides (SMGs) as the wireless channel. An illustrative example of our industrial partner's product, named Slotted Microwave Guides Middle Size (SMGM) with the antenna attached, is depicted in Figure 1.1.

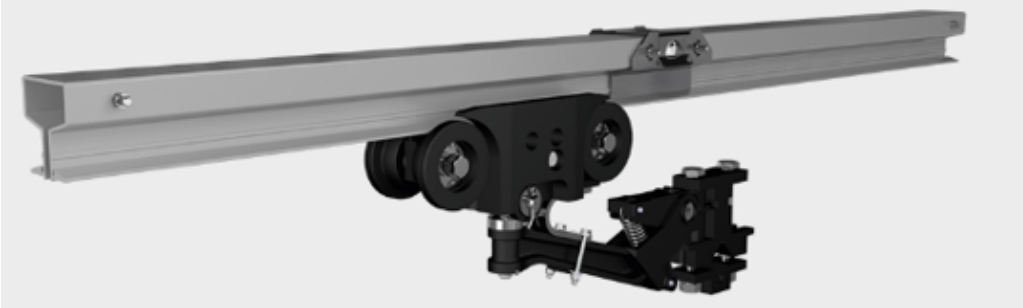


Figure 1.1: Slotted Microwave Guide Middle Size with the antenna attached

SMGs offer three prominent advantages, making them an ideal choice for the industrial automation field. Firstly, the utilization of SMGs minimizes channel interference and multi-path fading, allowing for the transmission of only line-of-sight (LOS) signals. Secondly, because SMGs are made of inductive material, it effectively blocks external signal propagation, thereby significantly reducing interference for the signal inside. Moreover, this attribute alleviates concerns regarding frequency certification, as the contained signals remain isolated, enabling flexibility in selecting the desired frequency for the SMG setup.

1.1.2.2 Transmission System

In contrast to the utilization of a cellular network as the transmission system in the 5G URLLC System, this project will deploy a Wi-Fi system on SMGs. Several compelling reasons justify Wi-Fi as a preferable choice for industrial applications over the 5G cellular network.

Although cellular network technology offers faster speeds, lower latency, and supports a larger number of devices, enabling high-quality video streaming through cell towers, Wi-Fi technology, as a type of local area network (LAN) technology, possesses unique and indispensable advantages. Primarily, Wi-Fi setup costs are minimal, as reliable communication environments can be established using affordable routers. Additionally, developers have complete control over the Wi-Fi system, with low maintenance costs. Individuals or organizations have the freedom to select Wi-Fi routers that suit their needs, enabling them to take advantage of various Wi-Fi standards and features. Consequently, both setup and maintenance fees remain relatively low. Considering the cost-effectiveness, flexibility in development, and ease of setup, Wi-Fi technology is an excellent choice for establishing limited-area networks in the industrial field.

1.1.2.3 Application Requirement

The 5G URLLC System adheres to stringent end-to-end latency and reliability rate values for all use cases, as outlined in Table 2.1. Conversely, this project does not impose strict requirements on latency and reliability rate. Instead, extensive measurements will be conducted to attain higher reliability rates on SMGs while also investigating the channel characteristics of SMGs.

1.2 Software Defined Radio (SDR)

Our objective is to achieve convenient access to a wide range of settings for various parameters, including packet size, modulation and coding scheme (MCS) index, Wi-Fi protocol, guard interval, carrier frequency, and more. To accomplish this, we employ a software-defined radio (SDR) solution.

1.2.1 Overview of SDR

The primary technology employed in this project is Software Defined Radio (SDR). SDR is also referred to as reconfigurable radio and flexible architecture radio. One typical software-defined radio architecture is shown in Figure 1.2.

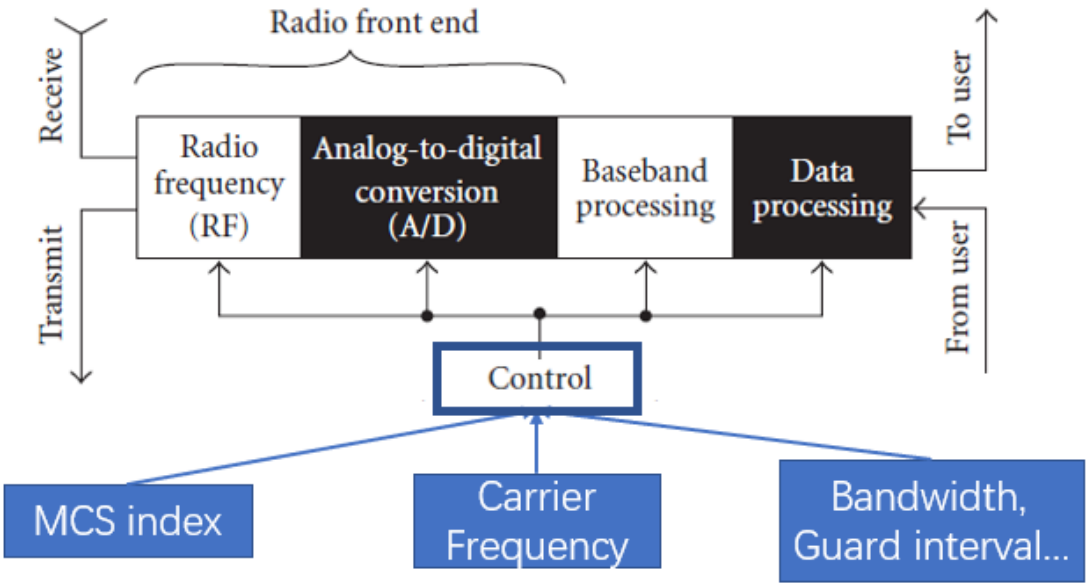


Figure 1.2: A typical SDR architecture

As depicted in Figure 1.2, the reception process in software-defined radio (SDR) closely resembles the transmission process, and thus our focus will be on the reception aspect. Similar to conventional transceivers, the received radio frequency signal is first captured by the receiver antenna. Subsequently, analog-to-digital converters (ADCs) convert the signal into a digital form. The signal is then down-converted from radio frequency to baseband frequency using mixers. An amplifier is employed to enhance the signal strength. Following these stages, digital signal processors (DSPs) take charge of processing the digital baseband signal. It is important to note that, to highlight the emphasis on analog-to-digital conversion and baseband processing, the mixer and amplifier are not depicted in Figure 1.2.

The analog-to-digital conversion plays a vital role in radio receivers, and in fact, an ideal software-defined radio is often perceived as an ADC (analog-to-digital converter) directly connected to an antenna. The function of the ADC is to convert continuous-time signals into discrete-time binary-coded representations. Recent advancements in SDR development, particularly in the flexibility of wireless devices at the physical layer, have spurred improvements in ADC performance. While various types of ADCs exist, their performance can be effectively summarized by several key parameters: signal-to-noise ratio (SNR), resolution (expressed as the number of bits per sample), spurious-free dynamic range, and power dissipation [3].

Baseband signal processing also plays a crucial role, with digital signal processors (DSPs) being responsible for executing the processing tasks within SDR. DSPs provide crucial development flexibility and are primarily utilized for computationally intensive tasks in signal processing algorithms. Traditionally, DSP techniques were employed for pre-modulation and post-detection functions in radio receivers. How-

ever, in recent times, DSP techniques have found extensive application in advanced digital communications transceiver designs, encompassing detection, equalization, demodulation, frequency synthesis, and channel filtering. The Fourier transform is a fundamental operation performed by DSP, with the Fast Fourier Transform (FFT) commonly used for software-based spectrum monitoring. By extracting frequency domain information from time domain samples, the FFT enables signal resolution into discrete frequency bins [3].

SDR distinguishes itself from the conventional transceiver by its ability to be reconfigured via a control bus. This enables the processing units to finely adjust both hardware and software parameters to meet desired specifications. This distinction enables convenient configuration of numerous features, such as the MCS index, center frequency, bandwidth, guard interval, and various PHY/MAC layer parameters.

1.2.2 Advantages of SDR

The following list outlines the advantages of SDR: [4]

- High Computing capacity: an SDR device is commonly accompanied by a field-programmable gate array (FPGA), which possesses substantial computing capacity.
- Flexibility: SDR offers high flexibility in terms of reconfigurability and adaptability to different communication standards and protocols.
- Upgradability: SDR systems can be easily upgraded through software updates, enabling the incorporation of new features and functionalities without hardware modifications.
- Spectrum efficiency: SDR enables efficient utilization of the available spectrum by employing advanced signal processing techniques and adaptive algorithms.
- Upgradability: SDR systems can be easily upgraded through software updates, enabling the incorporation of new features and functionalities without hardware modifications.
- Cost-effectiveness: SDR technology allows for the consolidation of multiple hardware components into a single platform, reducing costs associated with specialized hardware.

1.3 Project Goal & Challenges

1.3.1 Project Goal

The objective of this project is to provide our industrial partner, VAHLE, with a Wi-Fi based system design incorporating Wi-Fi communication parameters of their interest and acquire the transmission characteristics of the SMG through measurements focused on latency and packet error rate (PER). By gathering data from the SMG, VAHLE employees will be able to investigate a highly reliable and low-latency

solution tailored to their specific requirements.

1.3.2 Challenges

The challenges of this project are mainly focused on the following:

- Due to the scarcity of existing literature regading the SMG, our knowledge regarding the reliability of the SMG with respect to different Wi-Fi signals remains limited.
- The lack of experience in software-defined radio (SDR) adds to the considerable challenge of exploring PHY/MAC layer features on SDR devices.
- The design of the measurement campaign plays a pivotal role in this project. Careful consideration must be given to factors such as sample size, measurement time duration, data collection methods, data analysis tools and experimental conditions to ensure the campaign's effectiveness and reliability.
- This project involves the implementation of automated measurement scripts, necessitating the assurance of their stability.

2

System Architecture

In this project, Xilinx USRP E320s were used as the SDR devices. In this chapter, an overview of the E320s' hardware features would be provided. Additionally, I will illustrate the software architecture used in this project in detail. Finally, I will elaborate on the measurement preparation, including the setup of the virtual machine, the measurement conducting program and procedures, the bit rate synchronization procedures, and the log recording.

2.1 Hardware Platfrom – USRP E320

As the next generation of the E31 series, the E320 boasts improved features and capabilities. The USRP E320 is available in both 3U board-only and fully enclosed form factor variants. The fully enclosed version is shown below as Figure 2.1.

Figure 2.1: USRP E320 (Fully Enclosed Version)
[5]

2.1.1 Hardware Features

The hardware features of the USRP Xilinx E320 are listed below:

- Xilinx Zynq-7045 SoC

  - Dual-core ARM Cortex-A9 800 MHz CPU with 1 GB DDR3 RAM
  - 7 Series FPGA with 2 GB DDR3 RAM
- Single-plane PCB
- 1 SFP+ port (1 Gigabit Ethernet, 10 Gigabit Ethernet, Aurora)
- 1 RJ45 port (1 GbE)
- 1 Type A USB host port
- 1 micro-USB port (serial console, JTAG)
- Watchdog timer
- RF Network on Chip (RFNoC™) FPGA development framework
- Wide frequency range: 70 MHz to 6 GHz
- Up to 56 MHz of instantaneous bandwidth
- AD9361 transceivers
- RX, TX filter bank

2.1.2 Board Features

With robust computation capacity provided by the FPGA module, the USRP E320 implements enhancements in streaming, synchronization, integration, fault-recovery, baseband processing, and remote management capability.

There are 4 key advantages of the USRP Xilinx E320 that make it a superior choice for the SDR development.

2.1.2.1 Baseband Features

Utilizing the Xilinx Zynq-7045 SoC, the baseband processor provides a substantial FPGA that can be programmed by the user to process data in real time and with minimal latency, along with a dual-core ARM CPU for independent functioning.

2.1.2.2 Vast Frequency and Bandwidth Range

The flexible 2 X 2 MIMO AD9361 transceivers from Analog Devices continue to be used in this field deployable SDR devices, which can provide frequencies from 70MHz - 6GHz and offer up to 56MHz of bandwidth [5].

2.1.2.3 Synchronization Features

The USRP E320 features a synchronization architecture that is flexible and capable of supporting conventional SDR synchronization methods, like clock reference, PPS time reference, and GPSDO, making it possible to implement MIMO systems with a high channel count.

Note that our project involves designing and evaluating Wi-Fi-based URLLC solutions, where latency will be a crucial parameter that requires careful consideration. Therefore, a good synchronization architecture will be very necessary and helpful.

2.1.2.4 Low-cost Development

Application deployment on the preloaded embedded Linux operating system or sample streaming to a host computer can be achieved by users via high-speed interfaces like 1 Gigabit Ethernet, 10 Gigabit Ethernet, and Aurora.

Furthermore, the open-source USRP Hardware Driver (UHD) API and RF Network-on-Chip (RFNoC) FPGA development framework reduce software development effort and integrate with industry-standard tools such as GNU Radio. They enable users to rapidly prototype and reliably deploy designs for a variety of SDR applications, including spectrum monitoring and analysis, base station and UE emulation, mobile radio, and UAV communication/detection [5].

2.2 802.11 MAC/PHY Design

This thesis project utilized a MAC/PHY development stack designed and maintained by Mango Communications. With Mango's support and the use of several Linux command tools especially "iw" and "ip", this project was able to design and explore the 802.11 MAC/PHY layers.

2.2.1 Mango Communications

2.2.1.1 History

Mango Communications was founded in 2008 by Patrick Murphy. The company's origins can be traced back to the Wireless Open-Access Research Platform (WARP) project at Rice University. Initially funded by the NSF, WARP grew into an open-source wireless research platform that distributed hardware to select institutions in 2007. In 2008, Mango took over the manufacturing, sales, support, and design of WARP hardware. Today, hundreds of wireless researchers worldwide have adopted WARP.

The Mango team has concentrated on constructing the 802.11 MAC/PHY Design on third-party hardware platforms, collaborating with customers to extend and implement the 802.11 design. The team was granted US Patent 10841140 in 2020 [6].

2.2.1.2 Supported Hardwares

The 802.11 MAC/PHY Design is built in Xilinx Vivado is a powerful software suite used for designing and programming Xilinx FPGAs, or field-programmable gate arrays, and can be used in any Xilinx device supported by Vivado. Reference implementations are provided by Mango Communications on a variety of hardware platforms:

- Xilinx ZCU104 (MPSoC) with ADI FMCOMMS2 radio
- Xilinx ZCU111 (RFSoC)
- National Instruments USRP E320
- Analog Devices ADRV9361

2.2.1.3 Functionality Overview

Mango Communications provides a fully programmable 802.11 MAC/PHY development stack for Xilinx FPGAs. The implementation of Mango Communications 802.11 MAC/PHY stack in Xilinx FPGAs enables the deployment of a high-performing and real-time 802.11 MAC and PHY, which can interact with typical Wi-Fi devices. With our MAC and PHY implementations, users can easily prototype 802.11 standard extensions, investigate personalized wireless waveforms, and evaluate customized MAC protocols in real-world scenarios [7].

Included in the Mango 802.11 MAC/PHY design is a Linux driver called mango_wlan, which generates up to four wireless network interfaces on the Xilinx SoC running Linux alongside the 802.11 MAC/PHY design. These network interfaces function as standard Linux wireless NICs and support the same OS and application interfaces as regular Wi-Fi NICs (network interface card) but rely on the Mango 802.11 stack for all MAC/PHY processing. This integration enables a full view of the wireless networking stack, from the lowest RF/PHY interface to the highest application layer.

The 802.11 design fully implements the MAC protocol in embedded C code that operates on dedicated CPUs within the FPGA. On Zynq MPSoC devices, the MAC runs on the ARM R5 cores in the RPU, while on other devices, it runs on MicroBlaze CPUs within the FPGA. The MAC executes in real time and is designed to support custom protocol implementations.

To support custom MAC designs, the MAC software is split between the two CPUs in the FPGA design. Each CPU's software project is built on a MAC framework that provides common building blocks shared by multiple MAC designs. These frameworks simplify the development of custom MAC designs while allowing for maximum flexibility. The MAC frameworks are written in bare-metal C [8].

The Linux Device Interface application, AKA *linux_dev* from MangoCommunications, was used in this project. The *linux_dev* application is responsible for facilitating communication between the MAC and the *mango_wlan* kernel module in order to enable Linux to utilize the 802.11 MAC/PHY as a wireless network interface on CPU High. The application programs in shell format provided by the Linux Device Interface application can be divided into three parts:

- Wireless AP mode
- Wireless STA mode
- Wireless Monitor Mode

As implied by its name, when running in the first mode, Wireless AP mode, the device operates as an access point. It transmits beacon signals at a specified frequency and searches for connections with devices that intend to connect to the AP device. In STA mode, Station mode, the device operates as a station point. And for the monitor mode, users can monitor all the transmitted and received packets.

2.2.1.4 System Architecture

As mentioned in the last section, the Mango 802.11 MAC/PHY design supports operation as a Linux wireless NIC (non-Zynq-7000 and Zynq UltraScale+ MPSoC devices. The hardware architectures for both Zynq MPSoc and Zynq-7000 are illustrated below as Figure 2.2 [9].

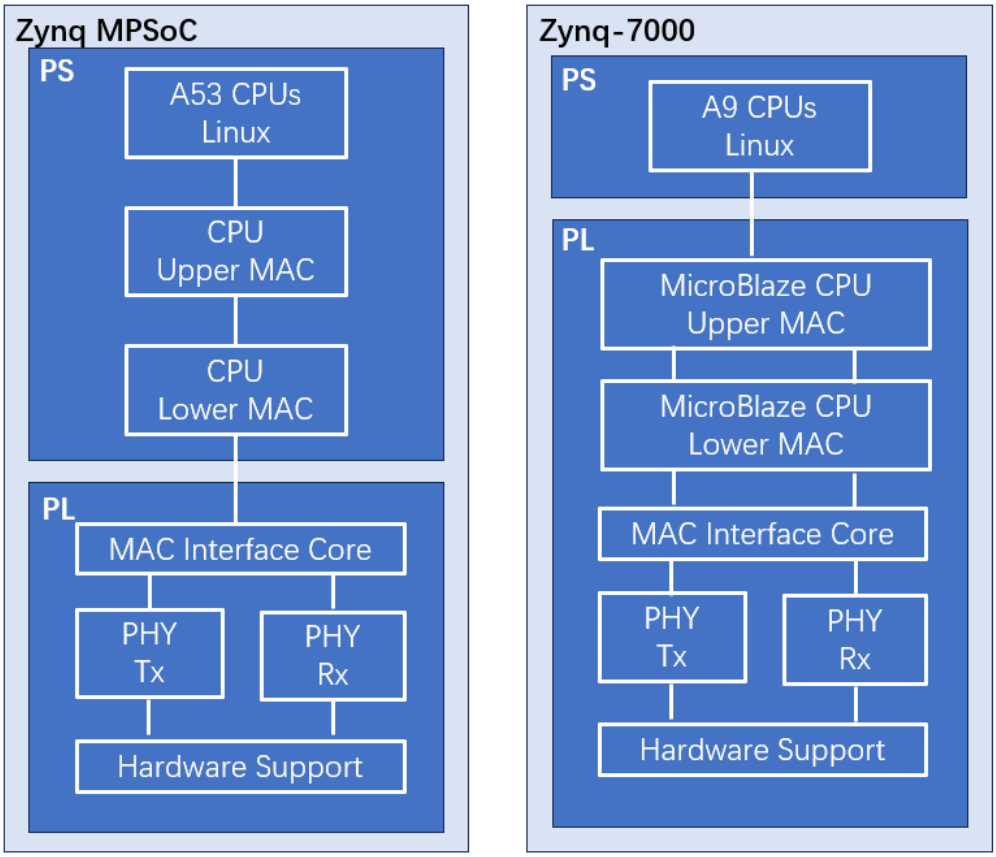


Figure 2.2: Linux Design Hardware Architecture
[8]

The implementation of the 802.11 PHY and radio interface on both platforms takes place in the PL (FPGA fabric). However, the MAC software operates differently on the two platforms. While on Zynq-7000 devices, the MAC runs on two MicroBlaze CPUs also located in the PL, on MPSoC, it runs in the hard ARM R5 CPUs in the PS RPU subsystem. Notably, the MPSoC design does not use any MicroBlaze

CPUs. Notice that the device used in this project is the USRP E320 which belongs to the Zynq-7000 family of devices [9].

Apart from the hardware architecture, to design a robust and scalable system, it is essential to prioritize the software architecture. Therefore, this thesis will emphasize the software architecture and its components. The software architecture was illustrated below as Figure 2.3.

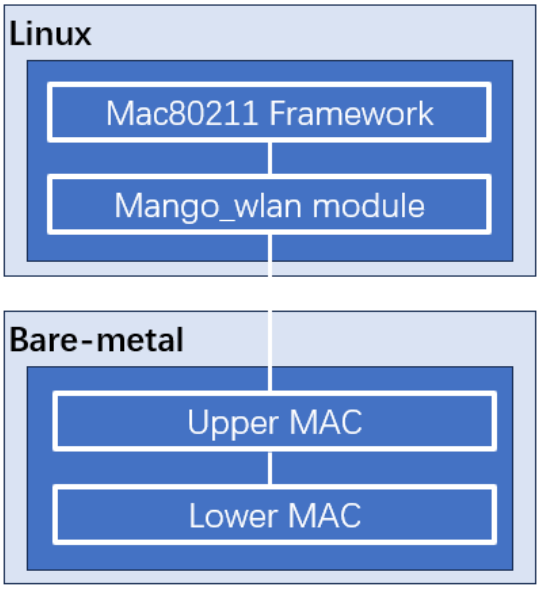


Figure 2.3: Linux Design Software Architecture [8]

The Upper MAC and Lower MAC are executed as bare-metal applications on separate R5 CPUs within the RPU (Reconfigurable Processing Unit). The mango_wlan kernel module serves as the 802.11 driver that connects to the mac80211 framework and networking stack of the kernel. By functioning as a standard wireless NIC driver, the mango_wlan driver enables the use of all applications compatible with regular Wi-Fi NICs [9].

2.2.2 Linux Command Line Tool

This project explored and utilized several features of Linux command line tools especially 'iw' and 'ip'. The 'ip' command is a very powerful and useful command for Linux operating systems. It is a robust utility that not only displays the current IP address of a system but also permits the viewing and management of network interface configurations, IP addresses, and routes. In this project, the tool was utilized to verify both the static IP address and the wireless IP address assigned by the user. Since 'ip' command will be further clarified in the System Booting section, no further explanation regarding the 'ip' command will be provided here.

The emphasis of this section will be on the 'iw' command, with particular attention given to its functionality, features, and usage in wireless network configuration and management.

2.2.2.1 Adding Wireless Interfaces

After the Linux system including MAC software applications provided by Mango Communications was booted up (the detailed booting procedures will be clarified in the System Booting part), the first step was to create wireless interfaces for both Xilinx USRP E320s, 'iw' command can help us do that in a very easy way. The interface can be created simply by typing the following command in the Linux Command Window:

```
iw phy mango–wlan–phy interface add wlan0 type managed addr
40:d8:55:04:20:ff
```

It should be noted that the prerequisite for adding a wireless interface operating in managed mode is that the hardware device must support monitor mode. To explain the command mentioned above, *mango-wlan-phy* refers to the name of our development stack as defined by Mango Communications. *wlan0* is the designated name for the wireless interface that we aim to add, while the last 12 digits represent the MAC address specifically assigned to this wireless interface. By executing the aforementioned command, a wireless interface operating in monitor mode, with the specified MAC address, can be successfully created under the Mango Communications development stack. Lastly, users can use the 'ip' command to verify whether the interface has been successfully created and configured correctly.

One important aspect that needs to be mentioned is the requirement to create two distinct MAC addresses for the AP (Access Point) and STA (Station) devices. Failing to do so will prevent the STA device from connecting to the AP device. In reality, it is highly unlikely for two devices to have the same MAC addresses. However, through the use of SDR (Software Defined Radio), it becomes possible to simulate devices with identical MAC addresses although it is not advised to do so.

2.2.2.2 Getting Link Status

Once the connection between an AP device and a STA device is established, users can type the following command to check the link status:

```
iw dev wlan0 info
```

We primarily use this command to check the status of the AP device, which provides information such as the name and MAC address of the wireless interface, the operating mode, the wireless channel (AKA the center frequency), the bandwidth, and the transmit power. See Figure 2.4 as an example of the shown output.

```
root@mango_wlan_linux_E320:~$ iw dev wlan0 info
Interface wlan0
        ifindex 5
        wdev 0x2
        addr 40:d8:55:04:20:ff
        ssid MANGO-AP2
        type managed
        wiphy 0
        channel 48 (5240 MHz), width: 20 MHz, center1: 5240 MHz
        txpower 20.00 dBm
```

Figure 2.4: The user interface of 'iw dev info'

It should be noted that the displayed transmit power is consistently shown as 20 dBm, but this does not accurately reflect the actual transmit power. Due to hardware limitations of the USRP E320, we are unable to set the transmit power through software. This is because the RF end of the USRP E320 contains a pre-amplifier. To determine the actual transmit power, we will assess and validate it in the Conducted Measurement section outlined in Section 3.

2.2.2.3 Obtaining Detailed Communication Parameters

Once a reliable connection is established between the AP and STA devices, at the STA end, one can enter the following command to obtain more comprehensive communication parameters:

```
iw dev wlan0 station dump
```

The figure below (Figure 2.5) shows all the features provided by running this command, including the number of transmitted and received bytes, the number of lost beacon packets, the average received power, the bit rates for the AP and STA devices, the beacon interval, and the used preamble...

```
root@mango_wlan_linux_E320:~$ iw dev wlan0 station dump
Station 40:d8:55:04:20:10 (on wlan0)
        inactive time:  10 ms
        rx bytes:       98826346
        rx packets:     898607
        tx bytes:       82145018
        tx packets:     684209
        tx retries:     22149
        tx failed:      32
        beacon loss:    5
        beacon rx:      107195
        rx drop misc:   6473
        signal:         -72 dBm
        signal avg:     -72 dBm
        beacon signal avg:      184 dBm
        tx bitrate:     9.0 MBit/s
        rx bitrate:     9.0 MBit/s
        expected throughput:    37.994Mbps
        authorized:     yes
        authenticated:  yes
        associated:     yes
        preamble:       long
        WMM/WME:        no
        MFP:            no
        TDLS peer:      no
        DTIM period:    2
        beacon interval:100
        short slot time:yes
        connected time: 11136 seconds
```

Figure 2.5: The user interface of 'iw dev station dump'

In this project, the main parameters of focus are the received power, the bit rates, the rx packet, and the tx packet. It is important to note that the two bit rates shown in Figure 2.5, namely 'tx bitrate' and 'rx bitrate,' are not the actual bit rates due to the limitations of "iw" command. We will elaborate in detail on this topic in the next sub-section.

2.2.2.4 Setting Bit Rate

Consider that our project involves conducting measurements with different bit rates (or, to put it another way, MCS indices). Therefore, configuring the bit rate settings is a necessary step. The 'iw' command enables users to have access to the setting of the bit rate by entering the following command:

```
iw dev wlan0 set bitrates legacy-5 24 ht-mcs-5 0
```

Through the preceding steps, the MCS index of the device can be configured to 0, effectively establishing a bit rate of 6.5 Mb/s. Nevertheless, instead of "6.5", a value of "24" was designated following "set bitrates". This is due to our observation from

multiple tests, which revealed that the "iw" tool necessitates the presence of at least one basic bit rate in the legacy-5 list to successfully set the MCS index, and 24 Mb/s is one of the basic bit rates in legacy-5 list. In accordance with the specifications, for our Wi-Fi system (detailed communication parameters for the Wi-Fi system will be illustrated in the subsequent content), a bit rate of 65 Mb/s corresponds to MCS 7, a bit rate of 58.5 Mb/s corresponds to MCS 6, 52 Mb/s corresponds to MCS 5, 39 Mb/s corresponds to MCS 4, 26 Mb/s corresponds to MCS 3, 19.5 Mb/s corresponds to MCS 2, 13 Mb/s corresponds to MCS 1, and 6.5 Mb/s corresponds to MCS 0.

In consultation with our industrial partner VAHLE, we reached a mutual agreement to utilize the following communication parameters for all measurements:

- Wi-Fi Amendment : 802.11n (Wi-Fi 4)
- Center Frequency : 5.24 GHz (the default setting for the SMG)
- Guard Interval : 0.8 us
- Security Protocol : WPA2
- Bandwidth : 20 MHz
- Bit Rates: 6.5, 13, 19.5, 26, 39, 52, 58.5, 65 Mb/s

The information of different MCS indices from 0 to 7 regarding the spatial stream, modulation, coding schemes and the bit rates is listed below in Table 2.1. Notice that the table 2.1 applies to the Wi-Fi amendments including Wi-Fi 4 (802.11n), Wi-Fi 5 (802.11ac), and Wi-Fi 6 (802.11ax) under the same communication parameters as listed above.

MCS index	Spatial Stream	Bit Rate	Modulation	Coding
0	1	6.5	BPSK	1/2
1	1	13	QPSK	1/2
2	1	19.5	QPSK	3/4
3	1	26	16-QAM	1/2
4	1	39	16-QAM	3/4
5	1	52	64-QAM	2/3
6	1	58.5	64-QAM	3/4
7	1	65	64-QAM	5/6

Table 2.1: MCS Index Table, Modulation and Coding Scheme Index 11n, 11ac, and 11ax

[10]

3

Measurement Preparation

3.1 System Booting

The Mango 802.11 design for the USRP E320 incorporates a customized boot flow that facilitates effortless design iteration. This boot flow allows for standalone operation, where the E320 boots from a local flash drive, as well as direct design iteration via USB/JTAG using the Xilinx Vivado and SDK tools. Nevertheless, unlike MPSoC platforms, MicroBlaze processors require a separate loading process for the MAC code. This can be achieved through JTAG and the Xilinx SDK, but such a setup can be inconvenient. This section describes an alternative boot procedure that loads the design completely from the SD card.

3.1.1 Procedures in Petalinux

Petalinux, an embedded Linux development platform created by Xilinx that is specifically designed for developing and customizing Linux-based systems on Xilinx Zynq SoCs (System on Chips) and MPSoCs (Multiprocessor SoCs), will be used for the booting up procedures. Due to space constraints, we will not elaborate further on the introduction of Petalinux.

Follow the following 7 steps one by one to complete the boot-up process for USRP E320 in Petalinux: [11]

- Follow the instructions in PetaLinux Projects to build the PetaLinux project as well as the MAC Software Workspace.
- Run *petalinux-package –boot –fsbl –u-boot –force*. This will create a new BOOT.BIN that does not contain the FPGA design. We will create a bitstream that contains the FPGA design along with the bitstream that contains the MAC software for the MicroBlaze processors in the next step.
- Navigate to the *<proj_root>/components/wlan_mac_workspace* folder, execute *wlan-sdk/scripts/E320_sw_bit_gen.sh* or the file with the same name and path but in a .bat format (if using Linux or Windows respectively)
- Copy *<proj_root>/components/wlan_mac_workspace/bootimg/boot_wlan_lin_dcf.bit* to the root folder of an SD card and rename the file wlan.bit.

- Copy *<proj_root>/images/linux/BOOT.BIN* to the root folder of an SD card.
- Copy *<proj_root>/images/linux/image.ub* to the root folder of an SD card.
- Copy *<proj_root>/components/util/uEnv.txt* to the root folder of an SD card.

One final point to address in this section is to ensure that before constructing the Petalinux project, two distinct static IP addresses are assigned to the two E320 devices through modifying *<proj_root/components/meta-mango-wlan/meta-mango/recipes-core/init-ifupdown/files/interfaces* file for the static IP of *eth0* [12].

3.1.2 Procedures in Xilinx SDK

Xilinx SDK, short for Xilinx Software Development Kit, will also be utilized for the boot-up procedures, we will not further introduce Xilinx SDK in this section, more information regarding Xilinx SDK can be found on the Xilinx official website.

Once the files generated from Petalinux have been successfully copied, copy the *ec-firmware-no-watchdog.bin* binary file and the *wlan.bit* bit file, both of which are generated from Xilinx SDK 2019.1, after successfully building the source code for the Linux Dev application within Xilinx SDK 2019.1. With this, the SD card is now prepared for booting up the USRP E320s. Finally, insert the SD card and power on the E320s, which operate in boot mode as the default setting for USRP E320s. This action will initiate the boot-up process, resulting in the Linux system with the complete Mango Communications framework being launched.

3.2 Wireless Connection Establishment

Before diving into the procedures for establishing a robust wireless communication link between an E320 operating in AP mode and another E320 operating in STA mode, it is important to note the frequency band chosen for this project. After consulting with our industrial partner, VAHLE company, we have mutually agreed that the frequency band utilized for this thesis project will be the 5GHz band. To provide further specificity, the designated center frequency for this project will be 5.24GHz, which is the default value prescribed by VAHLE for their SMGMs (Slott Microwave Guides Middle Size).

Regarding the process of establishing a wireless connection between two USRP E320 devices, the initial step upon powering the devices involves creating wireless interfaces for each device. The comprehensive steps for creating wireless interfaces are elaborated in the Linux Command Line Tool section within the System Architecture chapter.

Once the interfaces for both the AP and STA devices are added, we can modify various parameters in the configuration scripts provided by Mango Communications. The *hostapd.conf* file is used for configuring the AP device, while the *wpa_supplicant.conf* file is used for configuring the STA device. Expecting the wireless interface name and the country code, the parameters we have access to

include the following:

- The name of Wireless Access Point (AKA ssid)
- The supported bit rates
- The center frequency (AKA the wireless channel)
- The hardware mode
- The Wi-Fi amendment
- The security protocol

After making the necessary parameter changes, we can execute the corresponding shell scripts provided by the development stack for Mango Communications, *run_ap.sh* on the AP side and *run_sta.sh* on the STA side, to activate the respective modes on the devices. Following this, a connection will be established between the AP device and the STA device, utilizing the pre-configured features.

3.3 Virtual Machine Setup

USRP E320s allow users to establish communication with their laptops or terminal devices using UART cables or Ethernet cables. While it is possible to conveniently connect two E320 devices using UART cables and control them through a single UART port using a terminal device, Ethernet cables are essential for several reasons. Firstly, this project involves conducting measurements on a considerably long SMG, with the SMGM's line-of-sight distance in the lab at VAHLE being approximately 3.5 meters. Consequently, long cables are necessary, and Ethernet cables are more readily available in longer lengths compared to UART cables. Secondly, with an Ethernet cable connection, users can achieve high-speed transfer of log files from the E320s to their laptops. Lastly, using the 'ssh' command, users can securely stably access the USRP E320s.

Since each USRP E320 setup requires an Ethernet cable, and considering the fact that there are two USRP E320s, theoretically two laptops would be needed, each connected to its respective Ethernet cable. However, taking into account the challenge of managing two laptops simultaneously, an alternative approach involving the use of virtual machines is preferred. This allows for the use of a single laptop to connect with both Ethernet cables, thereby providing a more practical solution.

Please note that my setup involves a Windows operating system running on a PC, a virtual machine with Ubuntu 18.04 installed, and a USB-to-Ethernet adapter. After plugging in the adapter, which connects the PC to the USRP E320 via an Ethernet cable, I accessed the settings and configured a static IP address for the Linux system within the virtual machine. By ensuring that both the Linux system and the E320 device are assigned IP addresses within the same subnet, communication with the device via the virtual system using 'ssh' became possible.

To provide further clarification, let's consider an example. Suppose the static IP

address assigned to the E320 device is 10.30.1.1, with a subnet prefix length of 24, which implies a subnet mask of 255.255.255.0. To properly configure the Linux system in my virtual machine, its static IP address was configured as 10.30.1.55, utilizing the same subnet mask.

3.4 Retry Counter Setting

This research project entails conducting Packet Error Rate (PER) measurements to investigate the channel characteristics of Slotted Microwave Guides (SMGs). This necessitates transmitting and receiving a substantial volume of packets between the Station (STA) device and the Access Point (AP) device over both wired and wireless channels. During these measurements, the number of correctly received packets will be recorded at both ends, and metrics such as packet error rate, latency, and single link loss rate will be calculated and documented.

To achieve accurate measurements, it is crucial to ensure that no packet retransmissions occur. This means that when users send out one packet at the application level, exactly one packet should be transmitted at the lower layers, as opposed to multiple identical packets. However, in Wi-Fi networks, packet retransmissions may occur due to factors such as interference compensation, client roaming issues, transmit power mismatch, disabling of low data rates, and other variables. Such occurrences will significantly undermine the accuracy of our Packet Error Rate (PER) measurements [13].

Therefore, for this project, it is necessary to disable packet retries within our experimental setup. The detailed procedure involves navigating to the source code in Xilinx SDK, where the *wlan.bit* file is generated. Subsequently, we need to locate the section that defines the short transmission count and long transmission count. In this case, we will set the short count to 1, effectively indicating zero short retry attempts.

The short retry count signifies the maximum number of retransmission attempts for an RTS (Request to Send) packet or a data packet in the absence of RTS/CTS (Request to Send/Clear to Send) mechanism. On the other hand, the long retry count pertains to the maximum number of retransmission attempts for a data packet when RTS/CTS is employed. Considering our project employs customized UDP packets, through multiple iterations of modifications and experiments, they indeed fall under the purview of the short count. By implementing these modifications, we ensure that each packet in our measurements is transmitted only once, eliminating any potential impact of packet retransmissions on our results.

3.5 Measurement Scripts

In order to enhance the intelligence, efficiency, and automation of Packet Error Rate (PER) measurements, this project will involve the utilization of four programming languages: C, Go, Python, and MATLAB. Furthermore, in addition to the utiliza-

tion of MATLAB for log analysis and data visualization, which will not be elaborated upon further, this section will focus on the development and utilization of the following components: our test program, IRTT (Isochronous Round-Trip Tester), written in Go language; the bit rate synchronization script in C language; and the automation script in Python language.

3.5.1 Isochronous Round-Trip Tester

"Ping" may serve as a suitable tool for measuring latency and packet loss rate. Nevertheless, in the context of the embedded setup, it is comparatively less robust and powerful than the Isochronous Round-Trip Tester (IRTT). Consequently, in this project, IRTT was employed as our PER measurement test program, mainly for the following reasons.

Primarily, the IRTT test program, developed in the GO programming language, possesses a compact executable size, rendering it highly suitable for utilization on embedded devices—a key advantage for our project. Moreover, IRTT demonstrates remarkable proficiency in accurately measuring and monitoring various network behavior metrics of our interest, encompassing but not limited to round-trip time (RTT), one-way delay (OWD), instantaneous packet delay variation (IPDV), packet loss rates for both single link and round trips, bit rate, and payload size. Figure 3.1 presents an illustrative example depicting the output of a PER measurement conducted using IRTT [14].

```
                     Min        Mean      Median         Max  Stddev
                     ---        ----      ------         ---  ------
             RTT    3.84ms     4.18ms     3.92ms    10.53ms   717µs
      send delay  34h44m58s  34h44m58s  34h44m58s  34h44m58s   272µs
   receive delay -34h44m58s -34h44m58s -34h44m58s -34h44m58s   676µs

    IPDV (jitter)       9ns      450µs      24.1µs     6.62ms   874µs
       send IPDV        2ns      118µs      9.91µs     2.78ms   373µs
    receive IPDV         0s      351µs      11.4µs      5.1ms   840µs

   send call time     48.5µs      127µs                380µs  12.2µs
      timer error      120ns      272µs               1.31ms   169µs
server proc. time     18.8µs     21.5µs               81.6µs  1.41µs

             duration: 5m0s (wait 31.59ms)
  packets sent/received: 9919/9918 (0.01% loss)
server packets received: 9919/9919 (0.00%/0.01% loss up/down)
    bytes sent/received: 8927100/8926200
      send/receive rate: 238.1 Kbps / 238.1 Kbps
          packet length: 900 bytes
            timer stats: 81/10000 (0.81%) missed, 0.91% error
```

Figure 3.1: Example of the output of using IRTT [14]

What is worth mentioning is as the embedded setup does not entail synchronized

clocks between the client and server, the displayed one-way delay (send delay and receive delay) may be inaccurate. However, it is crucial to note that clock synchronization between the client and server is not required for accurate measurement of both round-trip time (RTT) and instantaneous packet delay variation (IPDV), including IPDV values for both sent and received packets. RTT is solely measured using client timestamps, and as IPDV measures differences between successive packets, it remains unaffected by time synchronization. Consequently, given that the primary parameters of focus in this project are the round-trip time (RTT) and packet error rate, this discrepancy is not expected to impact our measurements.

Moreover, the extensive range of functionalities offered by IRTT renders it an ideal choice for this project. Users can specify a multitude of parameters, including packet length, time duration, send interval, and time-to-live value, enabling customization of packet transmission according to their requirements. Additionally, the *df* flag can be employed to disable packet fragmentation. Most notably, the *dscp* flag allows users to indicate the desired quality of service (QoS) for transmitted packets and assign different levels of priority and treatment to different types of network traffic [14]. In this particular project, all packets were set to the highest priority. An example of an IRTT command at the client end is as follows:

```
/home/root/irtt client -i 5ms -l 400 -d 1500s --fill=rand
--df=true --dscp=56 --status=count 192.168.11.1
```

By doing so, we can define specific parameters for the transmission process. These parameters include a send interval of 5ms, a packet length of 400 bytes, a time duration of 1500 seconds, and assigning the highest priority to all packets. Furthermore, we enable packets to be populated with random bytes from GO's math.rand library, and the server IP address is specified at the end of the configuration. It is important to note that the send interval should be appropriately configured to ensure precise measurements. It is advised to avoid excessively short send intervals, as they may affect the accuracy of the measurements.

Regarding the server end, a straightforward command suffices to enable the server to respond to the received packets from the client and send acknowledgments back to the client. Here we will not provide an extensive elaboration on this matter.

3.5.2 Bit Rates Synchronization Program

As previously mentioned, this project entails conducting measurements with various MCS indices, which correspond to different bit rates. It is important to note that configuring the MCS indices is a bilateral process, meaning that the same value should be set simultaneously at both ends, namely the AP (Access Point) and STA (Station).

To accomplish this, two C program scripts were modified: a UDP packet generator for the client end and a packet receiver for the server end. The process involves the client sending UDP packets with customized payloads, repeated *LOOP TIME* times, to the server. The server, upon receiving the packets, distinguishes the content of

the payload and executes the appropriate bit rate setting command (one of the 'iw' commands, as described in the previous chapter) to adjust the bit rate accordingly. Note that the client end will execute the bit rate setting command as soon as it sends out the packets. The user prompts messages will also be displayed in the SSH terminal at the server end, serving as an indication that the bit rate synchronization has been successfully performed in an appropriate manner. An example of the prompt message displayed at the server end when configuring the MCS index to 7 (equivalent to setting the bit rate to 65 Mb/s) at both ends is depicted in Figure 3.2.

```
root@mango_wlan_linux_E320:~$ ./udp_server.exe
Server starting successfully
Setting bit rate to 65Mb/s
Setting bit rate to 65Mb/s
Setting bit rate to 65Mb/s
Setting bit rate to 65Mb/s
Setting bit rate to 65Mb/s
Setting bit rate to 65Mb/s
Setting bit rate to 65Mb/s
Setting bit rate to 65Mb/s
Setting bit rate to 65Mb/s
Setting bit rate to 65Mb/s
```

Figure 3.2: Example of the user prompt message at server end

Moreover, it is also important to note that in the provided example, the value of *LOOP TIME* was set to 10. However, this value may vary depending on the specific circumstances, especially in low signal-to-noise ratio (SNR) scenarios. In such cases, it is recommended to increase the number of synchronization packets sent. Lastly, in order to edit and execute C programs on the USRP E320, it is necessary to enable the GNU Compiler Collection (GCC) in Petalinux during the project's build process in order to obtain the Linux image. Detailed procedures for enabling GCC can be found on the website of Centennial Software Solutions [15].

3.5.3 Automation Script

Once the IRTT test program and bit rate synchronization script have been configured, the subsequent step involves merging them into an automated script that facilitates the automatic execution of PER measurements. This script allows users to define the packet size and encompasses all seven MCS indices.

With the utilization of this Python-based automation script, sequential execution of the 8 MCS indices will be performed, each with a designated packet size. If multiple packet sizes are specified, the process will repeat accordingly for each subsequent packet size.

4

Measurements & Results

In this section, two types of measurements will be performed: conducted measurements and measurements on the SMG (Slotted Microwave Guide). Conducted measurements are carried out by directly connecting two E320 devices using a known attenuation cable. Conducted measurements primarily serve the purpose of troubleshooting and problem diagnosis. Additionally, the wired channel's inherent stability makes it less susceptible to packet errors when compared to wireless communication channels. Therefore, conducted measurements provide a reference for conducting measurements on the wireless channel, e.g. on the SMG. The data gathered from the conducted measurements plays a crucial role in evaluating the measurements performed on the wireless channel, specifically on the SMG.

The conducted reference measurements will serve as the initial step, wherein the transmit power of USRP E320 devices will be verified using the conducted setup. Subsequently, packet error rate (PER) measurements will be carried out. Following that, measurements on the SMG will be conducted, which consist of three distinct parts. Firstly, Path Loss measurements will be performed to investigate the path loss pattern of the SMG. Secondly, Noise Floor Measurements will be conducted to determine the SMG's noise floor and a minimum acceptable received power for transmission on the SMG. Lastly, PER measurements will be performed on the SMG to assess the channel characteristics of the SMG.

4.1 Conducted Reference Measurements

4.1.1 Transmit Power Verification

Given that the USRP E320 incorporates wideband pre-amplifiers that are constantly active during transmission, neither the Mango Communication platform nor the Linux network control platform possesses the capability to modify transmit power levels comparable to Wi-Fi devices. Therefore, it becomes imperative to conduct transmit power verification in such cases.

The verification of transmit power was performed by connecting the transmitter, which is the USRP E320, to a spectrum analyzer using a coaxial cable with a 3dB attenuation. As our SDR devices transmit pulse-modulated OFDM signals, the energy of each subcarrier varies from symbol to symbol, making the measurement of the OFDM signal challenging. In accordance with the method described in [16], we

utilized the *Max Hold* function to obtain the average power across a 20MHz signal bandwidth. This was achieved by analyzing a sequence of 10^4 packets, each carrying a random payload with a packet size of 600 bytes.

Figure 4.1 displays the maximum power observed across the entire 20MHz bandwidth using the Max Hold function. It is important to note that the average received power displayed on the analyzer was -10 dBm, but we needed to subtract the -10 dB attenuation (RF attenuation factor) applied to obtain the actual received power which was -20 dBm. Thus, considering the 3 dB attenuation of the cable, the transmit power of the E320 was verified to be -17 dBm.

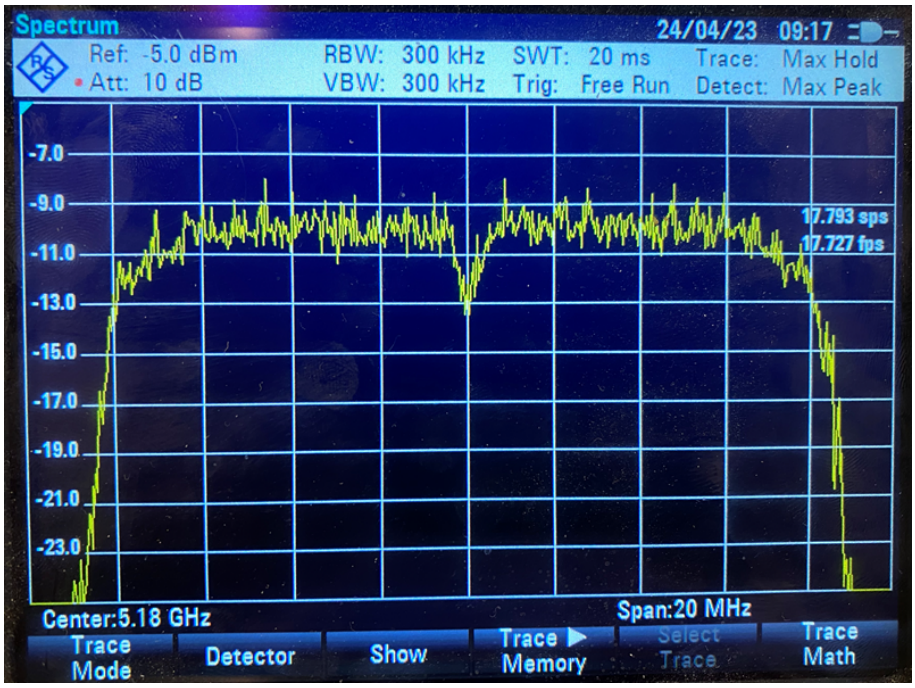


Figure 4.1: Display of Spectrum Analyzer

4.1.2 PER Measurements

In this section, PER measurements will be conducted in the conducted setup where two E320 devices are directly connected to each other using a known-attenuation cable. The primary objective of performing PER measurements in the conducted setup is to establish a reference for the lowest bound of error rates. This is because the wired channel is assumed to have fewer disturbances compared to the wireless channel, such as the SMG.

Regarding the experimental parameters, after consulting with VAHLE, we have agreed to conduct packet transmission measurements under the following conditions:

- Three different packet sizes: 60 bytes, 600 bytes, 1400 bytes.
- Eight different MCS indices: ranging from MCS 0 to MCS 7 (corresponding

to bit rates of 6.5 Mb/s, 13 Mb/s, 19,5 Mb/s, 26 Mb/s, 39 Mb/s, 52 Mb/s, 58.5 Mb/s, and 65 Mb/s).

- Two different received power : -60 dBm, -70 dBm

Thus, a total of 48 PER measurements will be performed in the conducted setup (3 packet sizes * 8 MCS indices * 2 received power levels). The number of measurements would double if we consider the PER measurements on the SMG. For each packet transmission measurement, 100, 000 UDP packets will be transmitted from the STA E320 device to the AP E320 device. Simultaneously, the round-trip and single-link latency, as well as the round-trip and single-link packet loss rate, will be recorded and documented. It is worth noting that the accuracy of the results is positively correlated with the number of transmitted packets. However, in order to ensure reliable results while minimizing the time required for measurements, we have chosen 100, 000 packets for each PER measurement.

Please note that while IRTT can offer us valuable insights regarding transmission, our primary focus in this study is on latency and packet loss rate. Figure 4.2 presents the round-trip latency and single link latency observed in the conducted setup across various measurement conditions. Upon observation of all six figures, it is evident that the mean latency closely aligns with the minimum latency, indicating the presence of only a limited number of outlier packets with significantly higher latency. Furthermore, as the packet size increases, there is a corresponding increase in the overall latency.

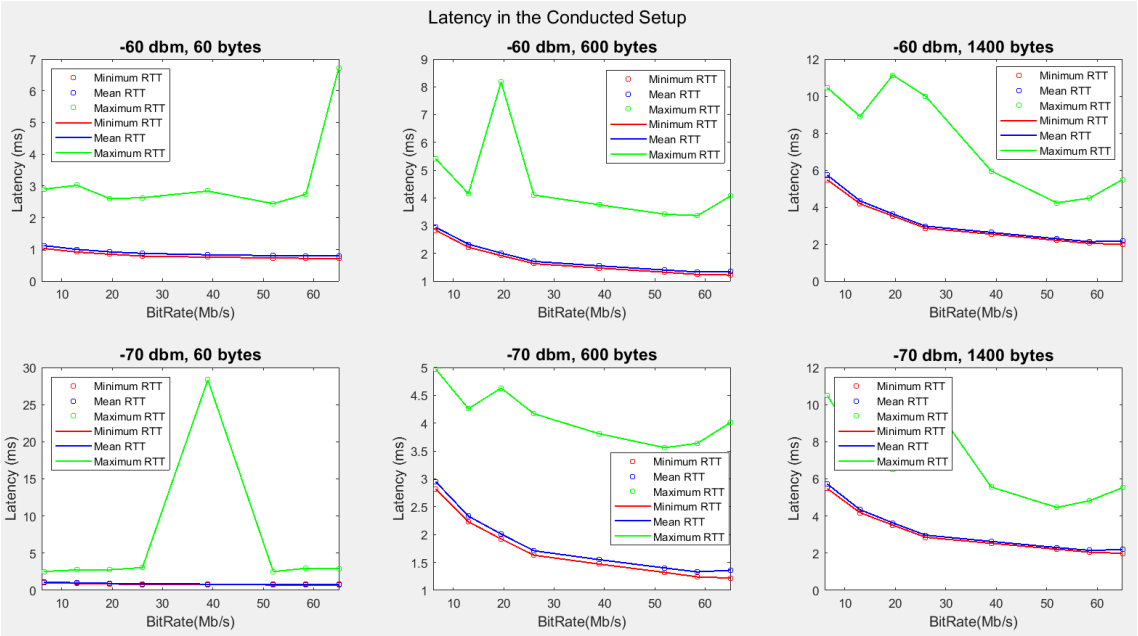


Figure 4.2: Latency in the conducted setup

Figure 4.3 and Figure 4.4 depict the round-trip and single link packet error rates at

the received power levels of −60 dBm and −70 dBm, respectively. In both figures, a logarithmic scale was applied to enhance the visualization of the trends presented in the three subplots below.

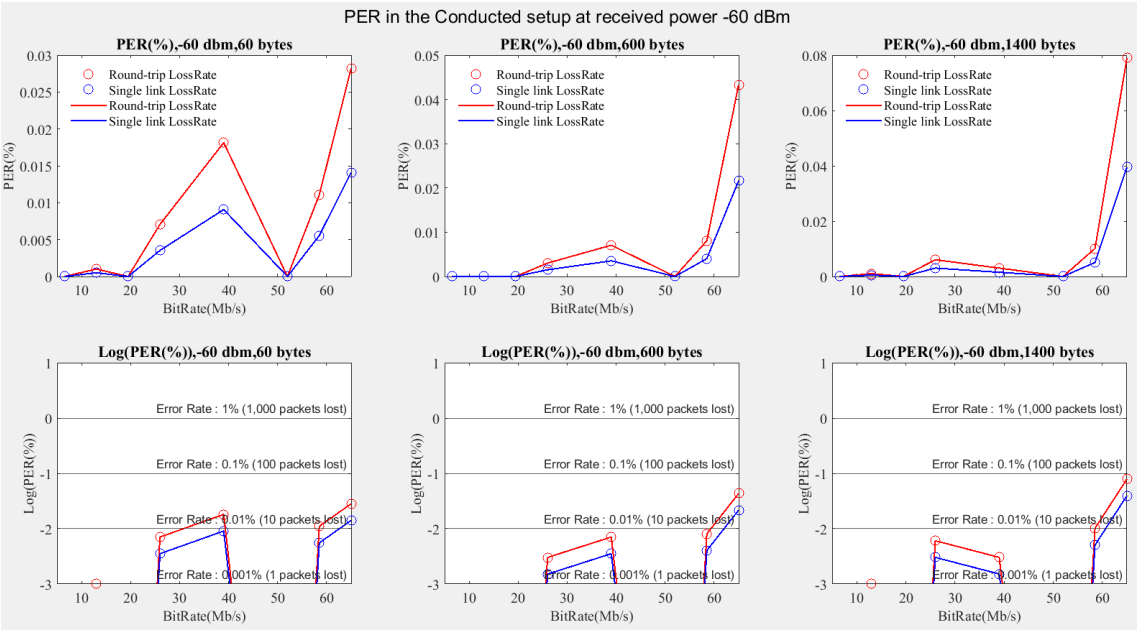


Figure 4.3: PER in the conducted setup at the received power -60dbm

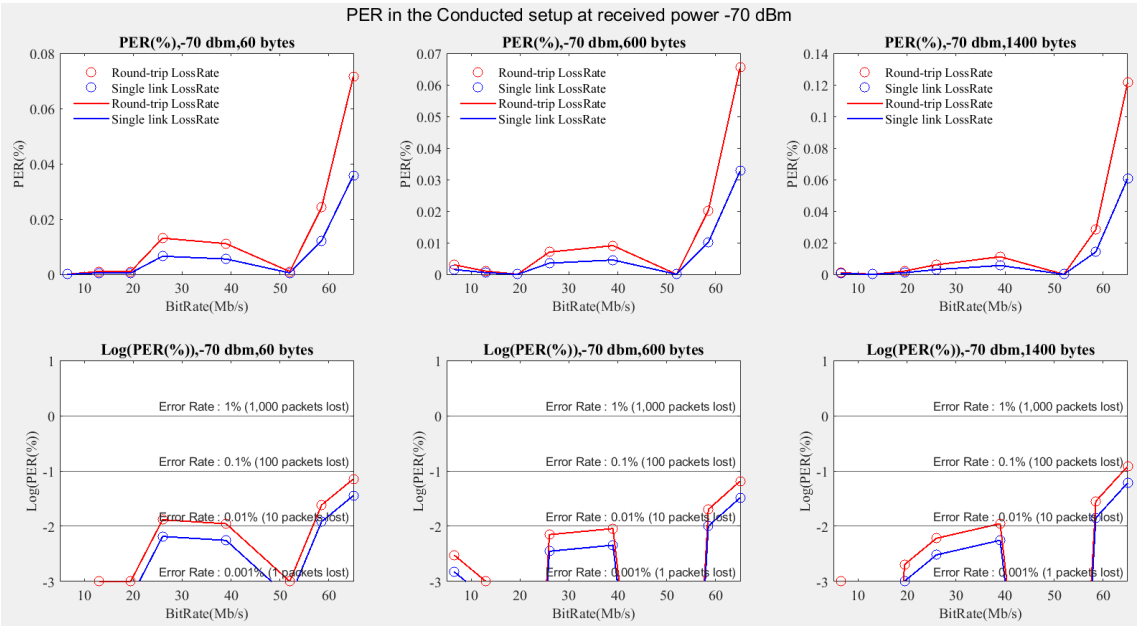


Figure 4.4: PER in the conducted setup at the received power -70 dbm

For each individual plot in Figure 4.3 and Figure 4.4, it was anticipated that packet error rates would exhibit a positive correlation with the MCS indices (or bit rates). Theoretical reasoning suggests that the probability of encountering lost packets should be higher when higher modulation schemes and coding schemes are employed. For instance, it is expected that the packet error rate of BPSK would be lower than that of QPSK and 16-QAM, and that the packet error rate of a 1/2 coding scheme would be lower than that of 5/6 coding scheme. However, while there is an overall increasing trend, some outliers are observed in each plot. Notably, the MCS index 5 (or bit rate 52 Mb/s) demonstrated a packet error rate of 0 in Figure 4.3 and Figure 4.4. To explain this result:

Firstly, it is important to consider that each transmission involved a substantial number of UDP packets, specifically 100,000. Given the large number of packets transmitted, the occurrence of approximately 10 lost packets (equivalent to 0.01%) is considered negligible. Consequently, for all the figures presented in the conducted part (Figure 4.3 and Figure 4.4), all MCS indices below 5 are deemed acceptable.

In addition, it is conceivable that the transmission process may have been subtly influenced by the intricacies of hardware circuitry or the network stack. This consideration prompted us to prioritize the conducted measurements. Conducted measurements were carried out specifically to alleviate the potential impact of these factors. By comparing these measurements with the data obtained from the SMG, we were able to effectively eliminate external influences and concentrate exclusively on assessing the impact of the channels under investigation. Further elaboration on the result comparison between the conducted channel and the SMG channel will be provided in section 4.3.

Lastly, due to time constraints, we limited the packet transmission to 100,000 packets. It should be noted that increasing the number of packets, say by a factor of 10, would likely result in packet loss for MCS index 5.

In both Figure 4.3 and Figure 4.4, which represent received power levels of -60 dBm and -70 dBm respectively, it is crucial to highlight the significant occurrence of packet errors associated with MCS indices 3 and 4. These indices correspond to bit rates of 26 Mb/s and 39 Mb/s respectively. Despite the relatively small number of lost packets, approximately 10, there were also notable instances of errors observed with MCS indices 6 and 7, corresponding to bit rates of 58.5 Mb/s and 65 Mb/s. Notably, in the majority of cases, the number of error packets for higher MCS indices (6 and 7) was approximately ten times higher than that for the lower MCS indices (3 and 4). Furthermore, the results indicate a clear correlation between an increase in packet size or a decrease in received power and a heightened likelihood of encountering a greater number of lost packets.

It is also noteworthy that while an overall increasing trend in the packet error rate with respect to the bit rate was observed, it is important to highlight that in most cases with the same modulation scheme, the packet error rate exhibits a strong positive correlation with the bit rate. For example, It is worth noting that MCS 7 and MCS 6 share the same modulation scheme, 64-QAM, but differ in their coding

schemes. Specifically, MCS 6 employs a coding scheme of 3/4, whereas MCS 7 utilizes a coding scheme of 5/6 (see Table 2.1). This observation further reinforces the notion that larger coding schemes tend to introduce a higher probability of encountering packet loss.

4.2 Measurements on the SMG

4.2.1 Path Loss Measurements

The path loss model for a wireless channel is essential for comprehending and forecasting the attenuation or signal loss encountered during wireless signal propagation through a medium. Additionally, it facilitates the determination of the signal-to-noise ratio (SNR) value. Consequently, the initial stride of measurements on the SMG is to acquire its path loss pattern.

The sketch of the SMG setup at VAHLE Company is depicted in Figure 4.5, spanning a total length of approximately fifteen meters. The setup comprises four segments of SMGs suspended from the ceiling, each labeled as segment 1, segment 2, segment 3, and segment 4. Considering that segment 1 has the longest line-of-sight (LOS) distance, our Path Loss measurements are conducted based on this segment.

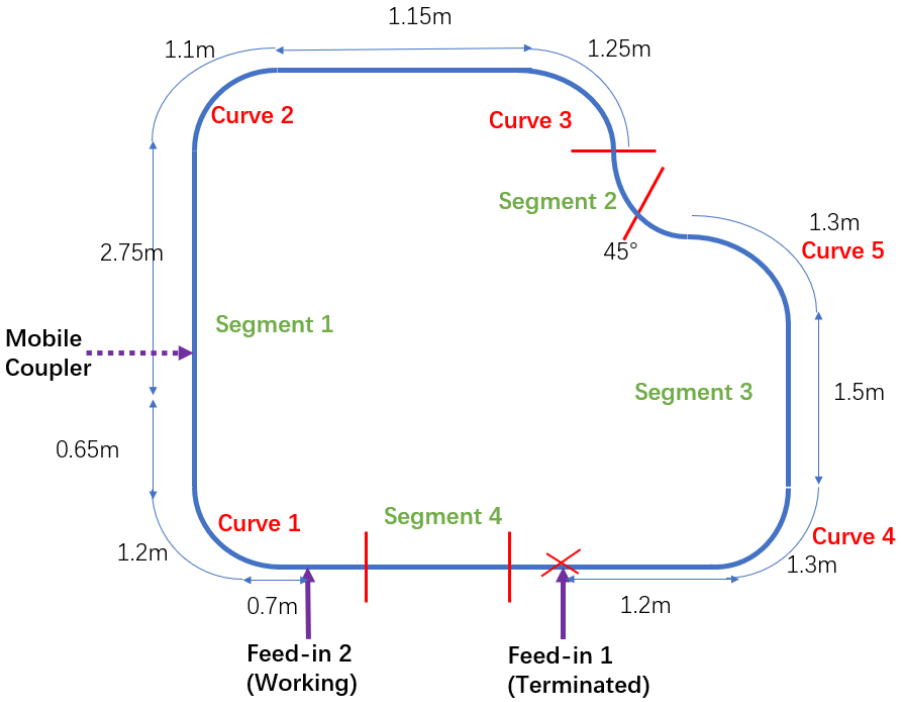


Figure 4.5: The Sketch of SMGM at Kufstein

As illustrated in Figure 4.5, to ensure utmost caution and enhance result precision, the initial step I took was to terminate Feed-in 1 with 50 ohm resistor. This

measure prevented signal leakage, which could have adversely affected the accuracy of our results. Additionally, with the AP antenna fixed at Feed-in 2, I gradually moved the STA antenna away from the AP antenna while concurrently recording the distance between them using the measurement tool. Simultaneously, the average received power at the STA E320 device was measured and recorded by observing the command window of the STA E320. The recorded data in Table 4.1 presents the average received power corresponding to the distance between the mobile coupler and Feed-in 2.

Based on the observations from Table 4.1, it is evident that the SMG provides a remarkably stable channel for line-of-sight transmission, there was only 1 dBm decreased for the received power when the distance between the AP antenna and STA antenna was roughly 1 meter (the worst cast) or above 3 meters (the best case). However, a substantial decrease of approximately 4 dBm in received power was observed when the STA antenna was relocated to curves. Consequently, subsequent discussions with VAHLE employees confirmed that this phenomenon was indeed attributable to the unavoidable imperfections in antenna attachment on the SMG, leading to the antenna movement out of the slot by 1mm, resulting in a lower received power.

Distance (m)	**Difference in placement angle (°)**	**Average Received Power (dBm)**
0.35	0	-70
0.55	0	-70
0.7	0	-70
0.8	20	-74
1	45	-74
1.6	75	-74
1.9	0	-70
...	...	...
3.6	0	-71
...	...	...
5.3	0	-71
5.5	20	-72/-73
5.85	45	-75/-76
...	...	...
6.4	0	-71/-72
...	...	...
7.55	0	-72/-73
7.95	20	-76
8.2	45	-76/-77
...	...	...
8.8	0	-73/-74

Table 4.1: Path Loss Pattern Measurement Record

4.2.2 Noise Floor Measurements

To ascertain the noise floor of the SMG and establish a minimum acceptable received power for transmission, we will carry out Noise Floor Measurements.

In order to determine the noise floor level, a conventional approach involves gradually moving the STA antenna away from the fixed-position AP antenna until the STA antenna no longer receives any signal at all MCS indices, indicating complete submersion of the signal in the noise. The received power in this scenario would approximate the noise floor. However, implementing this method in our SMG setup is impractical due to the limited line-of-sight distance. Therefore, an alternative approach was adopted where attenuators were introduced to the receiver to adjust the received power, thereby circumventing the need to physically move the STA antenna.

Without adding any attenuator to the receiver, the received power was approximately -70 dBm, which perfectly aligns with our calculations: -17 dBm (transmit power) - 18 dBm (attenuation of the mobile couplers with antennas) - 18 dB (the second mobile coupler) - 8 dB (attenuation of the 3-meter long RG316D cable) - 8 dB (the second RG316D cable) - 1 dB (path loss of the SMG for approximately 2-meter distance) = -70 dBm. Subsequently, we began adding attenuators at the receiver end while monitoring the received power. With a 30 dB attenuator and 3 dB attenuator (in total 33 dB attenuation), the received power measured approximately -107 dBm. Packet Error Rate (PER) measurements were then conducted, revealing an approximately 87.5% packet loss rate at the lowest MCS index (6.5 Mb/s bit rate) and a 100% packet loss rate at the remaining MCS indices. An additional 3 dB attenuator was introduced, resulting in the complete absence of signal transmission and the inability of the STA device to establish a Wi-Fi connection with the AP device. Based on these observations, we determined that the noise floor should be approximately -110 dBm.

4.2.3 PER Measurements

PER measurements were conducted on SMGs under the same measurement conditions as described in the previous PER measurements conducted in the Conducted setup. These conditions included three different packet sizes (60 bytes, 600 bytes, 1400 bytes), eight different MCS indices (ranging from MCS 0 to MCS 7, corresponding to bit rates of 6.5 Mb/s, 13 Mb/s, 19.5 Mb/s, 26 Mb/s, 39 Mb/s, 52 Mb/s, 58.5 Mb/s, and 65 Mb/s), and two different received power levels (-60 dBm and -70 dBm).

Figure 4.6 presents the round-trip latency and single link latency observed in the conducted setup across various measurement conditions. Similar to the conducted measurement, the same conclusion regarding the latency could be obtained on the SMG.

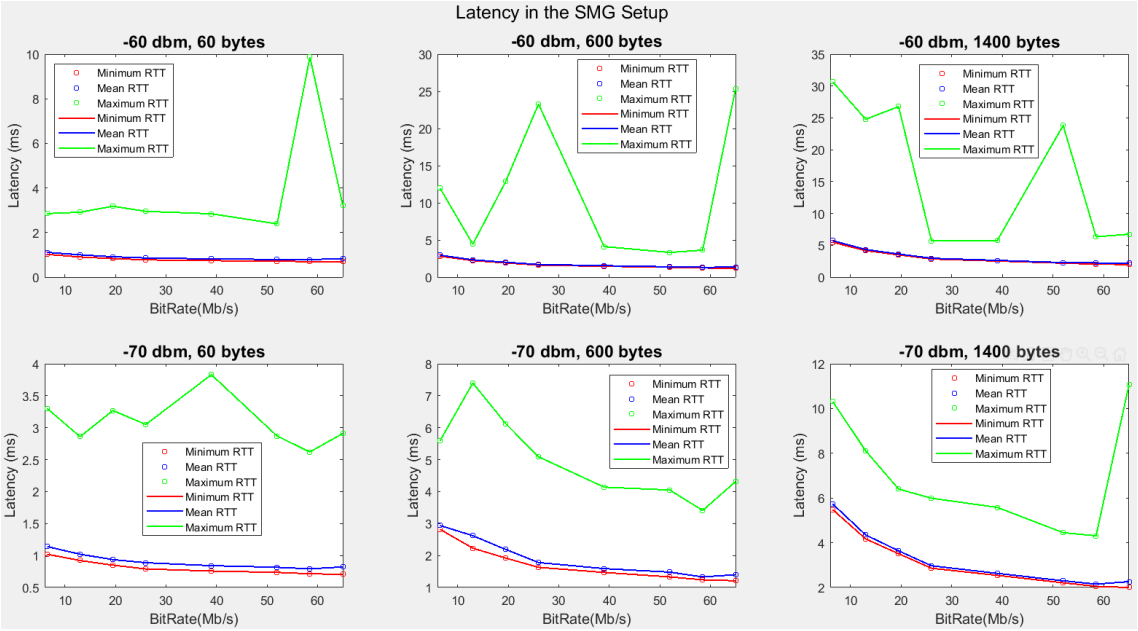


Figure 4.6: Latency in the SMG setup

Figure 4.7 and Figure 4.8 illustrate the round-trip and single link packet error rates at the received power levels of −60 dBm and −70 dBm, respectively. In both figures, a logarithmic scale was applied to enhance the visualization of the trends presented in the three subplots below.

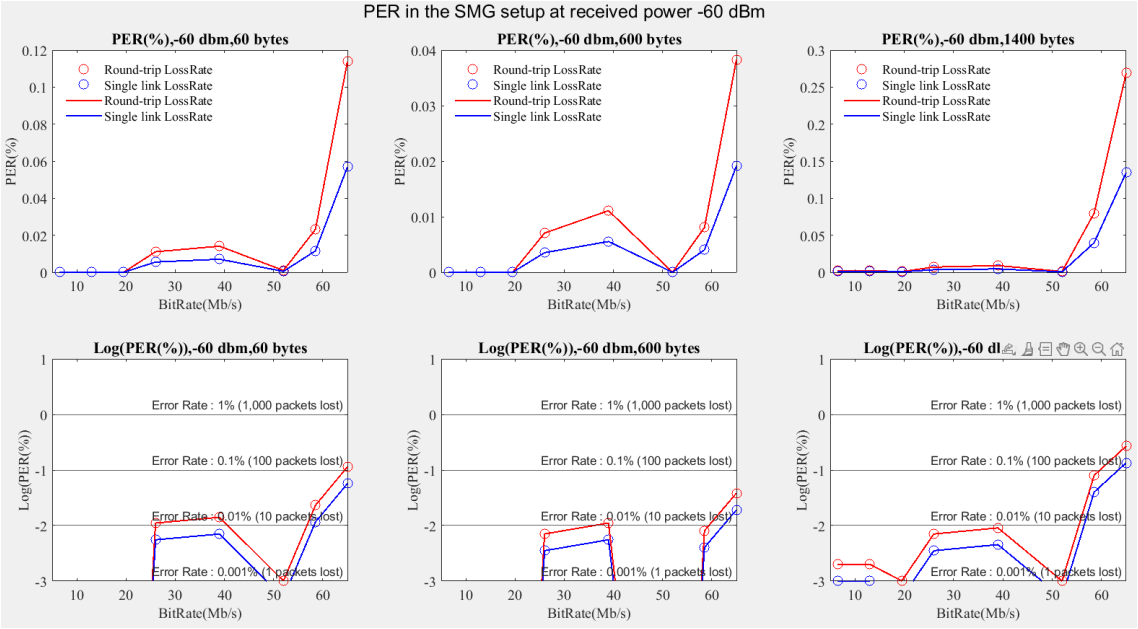


Figure 4.7: PER in the SMG Setup at received power -60 dBm

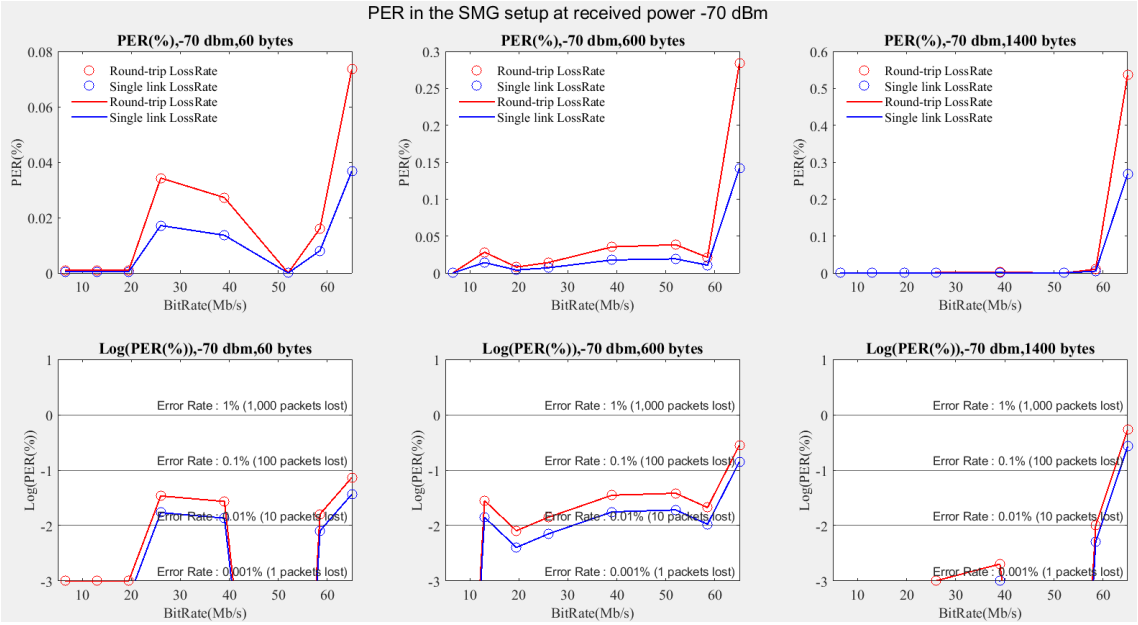


Figure 4.8: PER in the SMG Setup at received power -70 dBm

4.3 Comparison & Analysis

To enhance the comparison and analysis of the measurement results obtained in the conducted setup and on the SMG, Figure 4.9 has been included. This figure illustrates the PER results obtained from both the conducted PER measurements and the SMG PER measurements, specifically at a received power of -70 dBm. The upper three plots represent the conducted measurements, while the lower three plots depict the PER results obtained on the SMG.

Through an analysis of the PER results obtained from SMG measurements in comparison to the conducted measurements, we can eliminate the effects of non-channel factors on the results regarding packet error rates. These factors include the hardware device (USRP E320), attenuators, adaptors, and the test program (IRTT), as both the conducted measurements and SMG measurements utilized the same aforementioned factors. The only differing factor between these two types of measurements is the channel itself. Therefore, our attention is solely directed toward understanding the impact of the channel on our PER results. Moreover, since the wired channel for the conducted measurements is considered to have a more stable performance regarding packet error rates compared with wireless channels, we can accentuate the characteristics of the wireless channel. In other words, the channel characteristics of the SMG will be magnified.

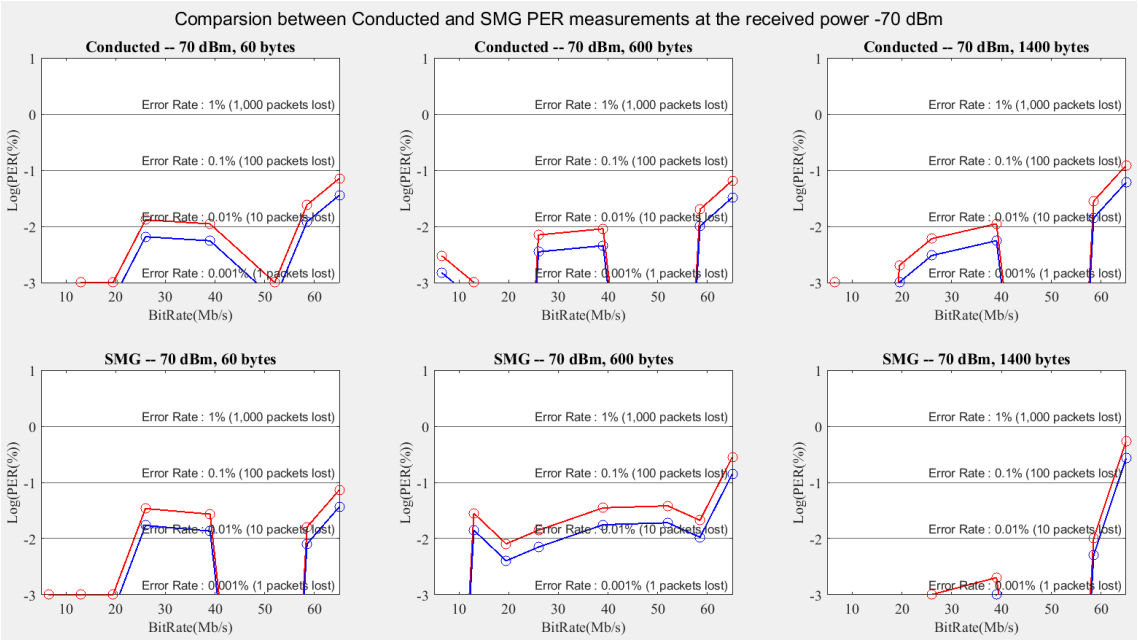


Figure 4.9: Comparison between the PER results between the conducted measurement and SMG measurement at the received power -70 dBm

Upon observing Figure 4.9, we can observe that approximately 10 out of 100, 000 packets were lost at MCS index 3 and 4 in the conducted setup, which corresponds to bit rates of 26 Mb/s and 39 Mb/s. This accounts for only about 0.01% of the total transmitted packets. Similarly, this phenomenon was observed in the SMG setup for both small packet size (60 bytes) and middle packet size (600 bytes) cases, albeit with slightly higher packet loss rates. Surprisingly, for the large packet size, the packet error rates on the SMG at MCS index 3 and 4 were even lower than those in the conducted setup.

Moreover, a consistent trend was observed for MCS indices 6 and 7, representing bit rates of 58.5 Mb/s and 65 Mb/s, where higher packet loss rates were consistently obtained in the SMG measurements compared to the conducted measurements. Notably, around 600 packets (equivalent to 0.55% of the total packets) were lost in the SMG setup for the highest MCS index 7 and the large packet size. This value is significantly higher compared to the loss rate of approximately 0.1% observed in the conducted setup under the same measurement conditions. This disparity can be attributed to the inherent instability of the wireless channel in the SMG setup. Furthermore, a considerable number of packet errors were also observed at the remaining MCS indices (MCS index 0, 1, 2, and 5) in the SMG setup for the middle payload size which is 600 bytes, which occurred rarely in the conducted setup.

In conclusion, compared with the conducted setup, the SMG exhibited consistently high packet loss rates and exhibited unstable packet transmission quality overall.

5

Conclusion

In conclusion, we leveraged the Mango Communication 802.11 MAC/PHY network stack to establish a robust 5GHz connection between the AP USRP E320 device and STA E320 device. Additionally, we utilized Linux network control command tools on our Xilinx USRP E320 devices for the design of our MAC/PHY layer, enabling us to monitor and verify real-time wireless network parameters such as center frequency, bit rate, and bandwidth. Through an extensive examination of SDR techniques, we explored various wireless link features and gained access to their configuration settings.

Moreover, we undertook an extensive investigation of the Isochronous Round-Trip Tester (IRTT) Program, implementing essential adjustments to various automated measurement scripts. These modifications facilitated the successful execution of automated packet transmission measurements, which were conducted under predefined communication conditions as specified by our industry collaborator, VAHLE. The measurements were carried out in both the conducted setup and on the slotted microwave guide (SMG). Lastly, leveraging our data analysis scripts, we meticulously visualized the measurement outcomes and conducted a detailed analysis of the results. This approach allowed us to gain valuable insights and draw meaningful conclusions from the collected data.

The aim of this project was to provide our industrial partner, VAHLE, with a Wi-Fi based system design incorporating Wi-Fi communication parameters of their interest, along with valuable data regarding latency and packet error rates across all MCS indices, three packet sizes (small, medium, and large) and specific received power levels that aligned with their areas of interest on their SMG products. The data acquired from these measurements will not only provide valuable insights for VAHLE's current system design but also greatly aid them in their exploration of other systems aimed at different industrial applications that align with their specific requirements.

In the end, we would like to express our gratitude to VAHLE for their collaboration and support throughout this research endeavor. Their partnership has been instrumental in our ability to conduct these measurements and gather valuable data.

Bibliography

[1] Morteza Ghobakhloo. Industry 4.0, digitization, and opportunities for sustainability. *Journal of Cleaner Production*, 252:119869, 2020.

[2] Zexian Li, Mikko A. Uusitalo, Hamidreza Shariatmadari, and Bikramjit Singh. 5g urllc: Design challenges and system concepts. In *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–6, 2018.

[3] M.N.O. Sadiku and C.M. Akujuobi. Software-defined radio: a brief overview. *IEEE Potentials*, 23(4):14–15, 2004.

[4] Tore Ulversoy. Software defined radio: Challenges and opportunities. *IEEE Communications Surveys Tutorials*, 12(4):531–550, 2010.

[5] Ettus Research. "usrp embedded series : Usrp e320". https://www.ettus.com/all-products/usrp-e320/. (accessed May.14, 2023).

[6] Mango Communications. "the introduction of mango communications". https://mangocomm.com/about/. (accessed May.16, 2023).

[7] Mango Communications Official Website. Mango support | 802.11 mac/phy user guide. https://mangocomm.com/802.11-mac-phy. (accessed May.14, 2023).

[8] Mango Communications. "the introduction of 802.11 mac/phy". https://support.mangocomm.com/docs/wlan-user-guide-v2/intro.html. (accessed May.23, 2023).

[9] Mango Communications. "mango support | 802.11 mac/phy user guide | linux driver". https://support.mangocomm.com/docs/wlan-user-guide/linux/arch.html. (accessed May.26, 2023).

[10] ACCESS AGILITY. "mcs index table, modulation and coding scheme index 11n, 11ac, and 11ax". https://www.accessagility.com/wifi-scanner. (accessed May.1, 2023).

[11] Mango Communications. "boot flow for usrp e320". https://support.mangocomm.com/docs/wlan-user-guide-v2/usage/usrp-e320/boot_flow.html. (accessed May.3, 2023).

[12] ExPECA TestbedConfig. "sdr node". https://kth-expeca.gitbook.io/testbedconfig/prepare/sdr. (accessed May.3, 2023).

[13] TIM MAIER. "wi-fi retry rate". https://www.csbtech.net/blog/wi-fi-retry-rate. (accessed May.8, 2023).

[14] Pete Heist Heistp. "irtt (isochronous round-trip tester)". https://github.com/heistp/irttdocumentation. (accessed May.29, 2023).

[15] Centennial Software Solutions. "advanced embedded platform development and maintenance for uniquely complex systems". https://www.csstechhelp.com/post/adding-gcc-to-petalinux-builds-compiling-code-on-fpga. (accessed May.3, 2023).

[16] M. Briggs, J. Martinez, and D. Bare. Power measurements of ofdm signals. In *2004 International Symposium on Electromagnetic Compatibility (IEEE Cat. No.04CH37559)*, volume 2, pages 485–488 vol.2, 2004.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY