Performance Modelling for Social VR Conference Applications in Beyond-5G Radio Access Networks

João Alberto Janeiro Horta de Morais

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. António José Castelo Branco Rodrigues
Prof. Remco Litjens

Examination Committee

Chairperson:	Prof. José Eduardo Charters Ribeiro da Cunha Sanguino
Supervisor:	Prof. António José Castelo Branco Rodrigues
Members of the Committee:	Prof. António Francisco Bucho Cercas

February 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of University of Lisbon.

Foreword

The research presented in this document was conducted in TNO as part of ‘Empowered Edge’, a project in a first year research programme on Social-Extended Reality. As such, and much for the sorrow of the author, the codebase will not be made public as of the publication of this document.

This document presents most of the work developed over the course of a year. Also, it is a first attempt in creating a reliable and readable documentation of this project to facilitate its development in the years to come.

Thousands of hours and I can only pick one quote? Do you think I write stuff like this every Tuesday? I am picking as many as I want!

João Morais (the author)

If I have seen further than others, it is by standing upon the shoulders of giants.

Isaac Newton

When solving a problem of interest, do not solve a more general problem as an intermediate step.

Vladimir Vapnik

We must be careful not to believe things simply because we want them to be true. No one can fool you as easily as you can fool yourself.

Feynman

Premature optimisation is the root of all evil (or at least most of it) in programming.

Donald Knuth

You can never cross an ocean unless you have the courage to lose sight of the shore.

Cristóvão Colombo (Christopher Columbus)

Acknowledgments

This is going to be long, I have plenty of people to thank and little shame about how long it takes. Their contributions amount to more than my effort since I am sure I could not achieve half of this without them. So I dedicate this work to them:

To my advisor in Portugal, Professor António Rodrigues who taught me my first Radio Systems course. I was helped unconditionally when needed, sometimes through the 'door of the horse', and allowed as much freedom as a student could wish for.

To my advising team in TNO, Professor Hans van den Berg, Professor Remco Litjens, and normal-person Sjors Braam, for crucial reviews, more than 50 meetings and the countless emails. To this day, *every time* we talk I learn something. Working with you truly humbles me.

To Remco, again, which was the main responsible for the invaluable opportunity of doing the thesis in TNO. You might have influenced me more than anyone in the last year. Sometimes, most likely by mistake, that influence was positive. Let me just defend this and I will get you your side of the deal for the tens, perhaps hundreds of hours you invested in me. A chocolate bar. Milk, no dark or bitter stuff.

To the people from TNO. I have met more than 60 people from the Networks department alone and I never felt so comfortable around so many incredibly experienced people. During the course of this project, particularly in the Radio Group, I learned and discussed many fascinating new concepts and ideas, some of which allowed me to collaborate in the creation of two patents in other projects within TNO.

Within TNO, Lucia D'Acunto deserves a paragraph of her own. As I write this you may be swinging on a trapeze at the Dominican Republic like a 'youngster', but for many months you were the most important pillar of my balance. I will leave it at that, not to risk getting you more confident than you already are. Thank you.

To Eduardo, Mike, Dinho, Rosa, Lucia (yes, again!) and others, for sharing movement in bouldering, capoeira, parkour and circus stuff, and being the scarce but essential training partners I had during the pandemic.

To Sandra Kizhakkekundil, an intern and now MSc student that I was fortunate to meet and co-advise. Our conversations often help me get clarity on complicated stuff. To Maria Raftapoulou, a PhD student that very directly contributed to this work by running link-level simulations. To Rodrigo Serrão, as you very well said, for raising and holding the bar high.

To my good friends Afonso and Bernardo. For amazing talks that provided the perfect escape after long days. For accountability with the studies and good habits. For the countless rides back and forth when I could not walk. And for the past +10 years of great stories.

And to my parents. I won't attempt to list why. For everything. Thank you.

Abstract

One of the most challenging applications targeted by evolving (beyond-)5G technology is virtual reality (VR). Particularly, ‘Social VR’ applications provide a fully immersive experience and sense of togetherness to users residing at different locations. To support such applications the network must deal with enormous traffic demands, while keeping end-to-end latencies low. Moreover, the radio access network must deal with the volatility and vulnerability of mm-wave radio channels, where even small movements of the users may cause line-of-sight blockage, causing severe throughput reductions and hence Quality of Experience (QoE) degradation or even lead to loss of connectivity. In this work we present and validate an integral modelling approach for feasibility assessment and performance optimisation of the radio access network for Social VR applications in indoor office scenarios. Such modelling enables us to determine the performance impact of e.g. ‘natural’ human behaviour, the positions and configurations of the antennas and different resource management strategies. Insights into these issues are a prerequisite for setting up guidelines for network deployment and configuration as well as for the development of (potentially AI/ML-based) methods for dynamic resource management and tuning of radio access parameters to best support Social VR applications.

Keywords

Modelling, Virtual Reality, 5G, Radio Access Networks, Wireless Communications

Resumo

A quinta geração de comunicações móveis(5G) tornou possíveis serviços inovadores. Em particular, serviços de realidade virtual com componente social oferecem uma experiência totalmente imersiva e uma sensação de união entre usuários. Para suportar tais aplicações, a rede tem que lidar com enormes volumes de tráfego e manter baixas as latências entre os extremos. Além disso, a rede de acesso de rádio deve lidar com a volatilidade e vulnerabilidade dos canais de rádio em ondas milimétricas, onde até mesmo pequenos movimentos dos usuários podem causar bloqueio de linha de vista entre antenas, causando graves reduções de taxa de transferência e, portanto, degradação da qualidade de experiência ou até mesmo perda de conectividade. Neste trabalho, apresentamos e validamos um modelo completo para avaliação e otimização do desempenho da rede de acesso rádio para aplicações de realidade virtual social. Tal dimensionamento permite determinar o impacto no desempenho de factores como o comportamento humano, as posições e configurações das antenas e diferentes estratégias de gestão de recursos rádio. Este conhecimento é imprescindível para definir diretrizes relativas ao equipamento rádio necessário e configuração de rede. Adicionalmente, permite o desenvolvimento de métodos, potencialmente baseados em inteligência artificial, para a gestão dinâmica de recursos e ajuste autónomo de parâmetros no acesso rádio com o intuito de melhor servir utilizadores de realidade virtual.

Palavras Chave

Modelação, Realidade Virtual, 5G, Redes de Acesso Rádio, Communicações Móveis

Contents

List of Figures

List of Tables

List of Abbreviations

3GPP	3rd Generation Partnership Project
4G	4th Generation
5G	5th Generation
5GI	5G QoS Identifier
ADC	Analog-to-Digital Converter
AI	Artificial Intelligence
AR	Augmented Reality
BB	Baseband
BLER	Block Error Rate
BICM	Bit-Interleaved Coded Modulation
BS	Base Station
BPSK	Binary Phase Shift Keying
CB	Code Block
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
CRI	Channel Report Indicator
CSI	Channel State Information
CSI-RS	Channel State Information - Reference Signal
DAC	Digital-to-Analog Converter
DL	Downlink
eMBB	extreme Mobile Broadband
EXP/PF	Exponential/Proportional Fair
F	Flexible
FDD	Frequency Division Duplex
FNBW	First Null Beam Width
FOV	Field Of View
FR	Frequency Range
FPS	Frames Per second
GBSM	Geometry-Based Stochastic Model
GoB	Grid of Beams

GoP	Group of Pictures
HEVC	High Efficiency Video Coding
HMD	Head Mounted Display
HOL	Head Of Line
HPBW	Half-Power Beam Width
ID	IDentifier
IQ	In-phase and Quadrature
LDPC	Low-Density Parity Check
LoS	Line-of-Sight
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MCU	Multi-point Control Unit
MIMO	Multiple-Input Multiple-Output
MI-ESM	Mutual Information Effective SINR Mapping
M-LWDF	Maximum-Largest Weighted Delay First
mMTC	massive Machine-Type Communications
mmWave	Millimetre Wave
ML	Machine Learning
MR	Maximum Ratio
MRC	Maximum Ratio Combining
MRT	Maximum Ratio Transmission
MTP	Motion-to-Photon
NLoS	Non-Line-of-Sight
NR	New Radio
OFDM	Orthogonal Frequency Division Multiplexing
OLLA	Outer Loop Link Adaptation
PDSCH	Physical Downlink Shared Channel
PDU	Protocol Data Unit
PER	Packet Error Ratio
PF	Proportional Fair
PHY	Physical
PMI	Precoding Matrix Indicator
PRB	Physical Resource Block
PSK	Phase Shift Keying
PUSCH	Physical Uplink Shared Channel
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
QoE	Quality of Experience
QoS	Quality of Service

Quadriga	QUAsi-DeteRministic RadIo channel GenerAtor
RAM	Random Access Memory
RAN	Radio Access Network
RAT	Radio Access Technology
RE	Resource Element
RF	Radio Frequency
RGB	Red Green Blue
RS	Reference Signal
RTT	Round Trip Time
RX	Receiver
SE	Spectral Efficiency
SINR	Signal-to-Interference-plus-Noise Ratio
SRS	Sounding Reference Signal
SSB	Synchronization Signal Block
TB	Transport Block
TBS	Transport Block Size
TDD	Time Division Duplex
TRX	Transceiver
TTI	Transmission Time Interval
TX	Transmitter
UE	User Equipment
UL	Uplink
ULA	Uniform Linear Array
URA	Uniform Rectangular Array
URLLC	Ultra-Reliable Low-Latency Communications
VR	Virtual Reality
XR	Extended Reality

List of Symbols

Latin alphabet

a_ϕ	azimuth lower limit for GoB
a_θ	elevation lower limit for GoB
b_ϕ	azimuth upper limit for GoB
b_θ	elevation lower limit for GoB
B	bandwidth
$BLER_0$	target BLER
C_m	coordinates of centre of mass of the room
C_t	coordinates of centre of the table
d_f	distance to the front of the user for camera placement
d_F	Fraunhofer distance
d_s	distance to the side of the user for camera placement
d_{out}	distance outwards from the head centre for HMD antenna offset
d_{up}	distance upwards from the head centre for HMD antenna offset
D	largest dimension of radiator to estimate effective area
E_b	energy per bit
$\boldsymbol{H}_{bul}$	channel matrix between BS b and UE u in layer l
I	interference
I_0	antenna element feed current amplitude
j	imaginary unit $\left(j = \sqrt{-1}\right)$
k	wave number
k_B	Boltzmann constant
L_{max}	maximum latency for the radio link
N_0	noise power spectral density
N_r	number of receive antennas
N_t	number of transmit antennas
N_{CSI}	number of beams with CSI-RS
N_{bs}	number of BSs
N_{cam}	number of cameras
N_{phy}	number of physical users

N_{vir}	number of virtual users
N_{ue}	number of UEs
N_{users}	number users or participants
N_x	number of antennas along the x-dimension
N_y	number of antennas along the y-dimension
N_{ant}^{UE}	number of antennas on the UE side
N_{ant}^{BS}	number of antennas on the BS side
N_{slots}^{DL}	number of DL slots in a transmission period
N_{slots}^{UL}	number of UL slots in a transmission period
N_{slots}^{TDD}	number slots in a transmission period
N_{slots}^{CSI}	number slots between CSI updates
N_{slots}^{SCH}	number slots between scheduling updates
$N_{PRB,bu}$	number of PRBs allocated for link between BS b and UE u
$N_{PRB,bul}$	number of PRBs allocated for link between BS b and UE u in layer l
N_{symb}^{PRB}	number of symbols per PRB
N_{bits}^{symb}	number of bits per symbol
$N_{infobits}^{symb}$	number of information bits per symbol
N_{bits}^{slot}	number of bits per slot
N_{bits}^{PRB}	number of bits per PRB
$N_{bits,bul}$	number of bits to be sent between BS b and UE u in layer l
NF_{BS}	BS noise figure
NF_{UE}	UE noise figure
NF_r	noise figure at the receiver
p	scheduling priority
P_N	noise power
P_u	position of user u
P_r	received power
P_s	signal power
P_t	transmit power
$P_{t,max}^{UE}$	maximum transmit power per UE
$P_{t,max}^{BS}$	maximum transmit power per BS
$P_{r,bu}^{UE}$	received power by the UE for a link between BS b and UE u
$P_{t,bu}^{BS}$	transmit power at the BS for a link between BS b and UE u
$P_{t,bu}^{UE}$	transmit power at the UE for a link between BS b and UE u
$P_{t,bul}$	transmit power for a link between BS b and UE u in layer l
$P_{t,bul}^{UE}$	transmit power at the UE for a link between BS b and UE u in layer l
$P_{t,bul}^{BS}$	transmit power at the BS for a link between BS b and UE u in layer l
P_{IaCI}	interference power from intra-cell interference
P_{IeCI}	interference power from inter-cell interference

P_{ILI}	interference power from inter-layer interference
Q_m	modulation order
r_ϕ	azimuth resolution for GoB
r_θ	elevation resolution for GoB
r_t	radius of table
r_u	radius of user circumference of disposition
$r_{DL/UL}$	ratio between DL and UL application throughputs
$r_{P/I}$	ratio between P-frame and I-frame sizes
R_b	instantaneous bit rate
R_c	code rate
R_{packet}	packet arrival rate
R	instantaneous throughput
$\overline{R}$	average throughput
$\overline{R}_{DL}$	average application throughput in the DL
$\overline{R}_{UL}$	average application throughput in the UL
R_F	frame rate
s	speed of user head position change
s_{TDD}	TDD split
$SINR_{eff}$	effective SINR, i.e. aggregated over all scheduled PRBs
$SINR_i$	SINR of the i-th PRB
S_{GoP}	size of a GoP
S_I	size of I-frame
S_P	size of P-frame
S_{packet}	size of a packet
S_{TB}	size of a TB
$S_{TB,max}$	maximum size of a TB
t_w	exponential smoothing window size for proportional fair ratio
T	noise temperature
T_{sim}	simulation duration
$T_{slot,\mu}$	slot duration for numerology μ
T_{rot}	interval between consecutive orientations when head rotating
$\boldsymbol{w}$	vector of beamforming weights
$\boldsymbol{w}_{\phi,\theta}$	beamforming weights from a GoB with direction (ϕ, θ)
$\boldsymbol{w}_{i,j}$	beamforming weights vector with grid indices (i, j), from a GoB
$\boldsymbol{w}_{bu}^{BS}$	beamforming weights between BS panel b and UE u, at the BS
$\boldsymbol{w}_{bu}^{UE}$	beamforming weights between BS panel b and UE u, at the UE
$\boldsymbol{w}_{t,bul}$	transmit weights vector between BS panel b and UE u, in layer l
$\boldsymbol{w}_{r,bul}$	receive weights vector between BS panel b and UE u, in layer l

Greek alphabet

α_u	angle from the centre of the table to user u
β_x	upper limit on uniform distribution for rotation around x axis
β_y	upper limit on uniform distribution for rotation around y axis
β_z	upper limit on uniform distribution for rotation around z axis
γ	burstiness parameter for application traffic
γ_{OLLA}	step size for OLLA parameter (Δ_{OLLA}) update
Δ_{OLLA}	outer loop link adaptation parameter
η_{OH}	efficiency due to overhead
η_{slot}	efficiency in bit rate from slot format
λ	wavelength
o	overlap parameter for application traffic
σ_x	standard deviation of normal distribution for position coordinate x
σ_y	standard deviation of normal distribution for position coordinate y
σ_z	standard deviation of normal distribution for position coordinate z
τ_{CSI}	CSIs delay, in number of TTIs
τ_{ACK}	delay before acknowledgement, in number of TTIs

Sets:

$\mathcal{B}$	base station panels
$\mathcal{U}_b$	users served by base station b
$\mathcal{L}_{bu}$	layers scheduled between base station b and user equipment u
$\mathcal{F}$	frequencies for simulation

Other nomenclature

$\boldsymbol{A}$	matrix
$\boldsymbol{a}$	column vector
$\lvert\boldsymbol{a}\rvert$	euclidean norm of vector $\boldsymbol{a}$
$\boldsymbol{A}^{\mathsf{T}}$	transpose of $\boldsymbol{A}$
$\boldsymbol{A}^{\mathsf{H}}$	Hermitian of $\boldsymbol{A}$, also know as the transpose conjugate of $\boldsymbol{A}$
$\lceil a \rceil$	ceil a, i.e. round up a to the nearest integer
$\lfloor a \rfloor$	floor a, i.e. round down a to the nearest integer
$\hat{a}$	estimate of a
$\overline{a}$	average of a

1

Introduction

Contents

1.1 Motivation

Information and Communication Technologies (ICT) have seen significant advances in the last decades, and they have empowered many new applications. Today, with the emergence of the new 5th Generation (5G) of mobile communications, everything seems to be at the edge of change.

The objective of 5G is to meet service requirements from various economic sectors, e.g. Automotive, Media & Entertainment, Health and Industry and Energy. To achieve such feat, three main generic services are defined: extreme Mobile Broadband (eMBB) concerns with supplying extreme data-rates, massive Machine-Type Communications (mMTC) aims to connect the highest number of devices, usually with low data-rates, and Ultra-Reliable Low-Latency Communications (URLLC) guarantees essential Quality of Service (QoS), like millisecond latencies and 99.999% reliabilities, for mission-critical applications. Figure 1.1 illustrates how some applications relate to these generic services.

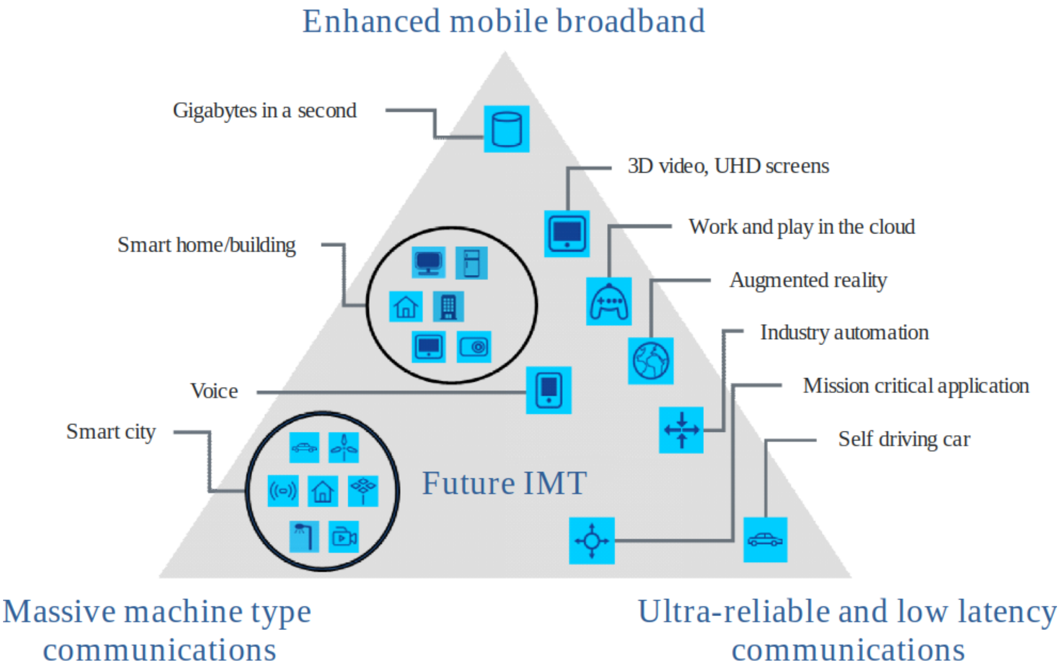

Enhanced mobile broadband
Gigabytes in a second
3D video, UHD screens
Work and play in the cloud
Smart home/building
Augmented reality
Industry automation
Voice
Mission critical application
Smart city
Self driving car
Future IMT
Massive machine type communications
Ultra-reliable and low latency communications

Figure 1.1: Usage Scenarios of 5G. [1]

The service specific characteristics place it closer to eMBB (e.g. Virtual/Augmented Reality (VR/AR) entertainment), closer to mMTC (e.g. the many sensors and actuators distributed in a Smart City) or closer to URLLC (e.g. remote VR-based surgery). Indeed, it this vast requirement heterogeneity across services and markets that has been driving 5G development [13], far surpassing previous generations. Figure 1.2 shows a comparison between the requirements of last generation of mobile communications and 5G's requirement.

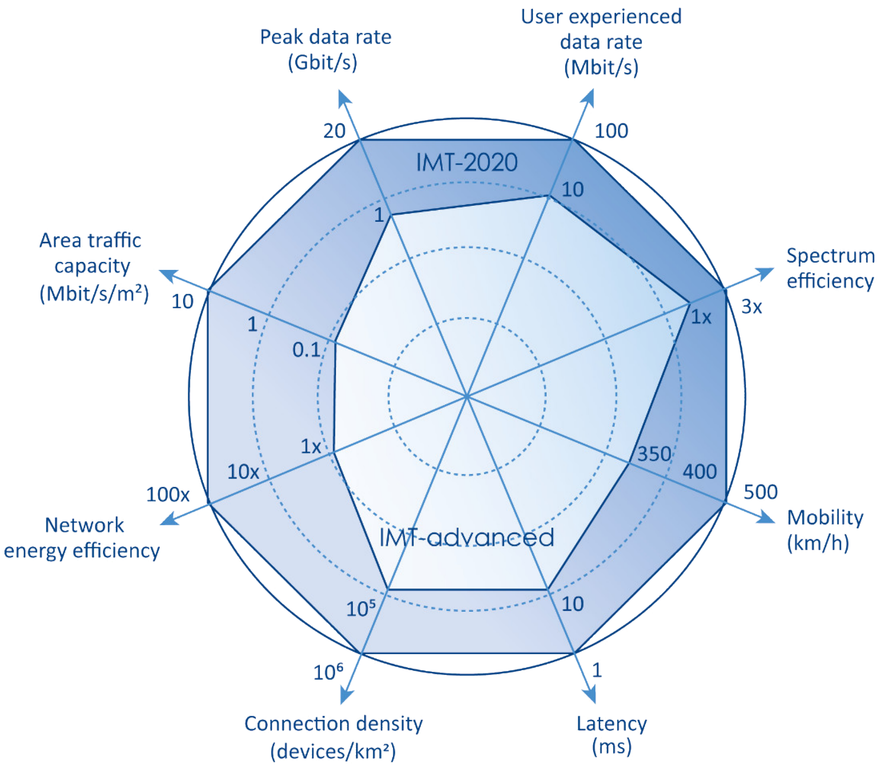


Figure 1.2: Spider diagram comparing 4G and 5G requirements. Source [2]

To meet these challenging requirements, it is not just a matter of proper network planning and management. 5G brings many new advancements and technologies to make such requirements attainable, e.g. high-frequency spectrum, constant beamforming-based operation, moving intelligence to the network edge and New Radio (NR) access technologies (5GNR). In particular, the role played by the Radio Access Network (RAN) is crucial in achieving this feat. As such, the efficient management of radio resources has been a pivotal challenge network operators have to face. Such radio resource management comprises a suite of mechanisms, including admission control, scheduling, beam management and adaptive modulation and coding. Said mechanisms operate on different timescales and need to be suitably configured to fit spatio-temporal changes in traffic, user mobility, propagation environment and service mix.

This task is of utmost complexity. In fact, the software complexity of RAN in a Base Station (BS) exceeds that of Boeing 787 aircraft [14]. Naturally, with evergrowing demands traditional mechanisms start to lack the required performance. Especially given the recent wide range of applicability, Artificial Intelligence (AI) and Machine Learning (ML) techniques promise to be capable of network management strategies that achieve optimal resource adaptation to a given context [15]. As opposed to traditional methodologies that struggle with increased amounts of data, ML tools like neural networks perform better with more data.

Moreover, how can one physical network adapt to multiple types of services with fundamentally different requirements? There are countless configurations across the network that can be optimised to the fulfilment of a given service, but what mix of configurations represents the best trade-off for a given mix of services? And such configurations need to be dynamically managed to cope with instantaneous network and propagation conditions by balancing and allocating resources accordingly.

The answer is Network Slicing. 5G's network architecture enables the multiplexing of virtualised and independent logical networks (called slices) on the same physical network infrastructure. Therefore, application-specific programs running concurrently can automatically tune parameters and configurations across the network in order to best service the user.

Networks slicing is, from the conceptual/architectural point of view, a well-investigated topic [16]. However, (resource) management for network slices, in order to realize the required service level in a resource-/cost-efficient way, is still an open research challenge, in particular for the radio access network.

In this thesis, we focus on enhancing the radio access for an emerging and incredibly demanding application that would not only benefit but certainly require such optimisations to have its requirements fulfilled. The application in question is social Extended Reality (XR) conferences. It consists of virtual or augmented reality meetings where people can see and interact with each other virtually. It requires the reliable transmission of photo-realistic images of the body of each participant to all other participants hence necessitating very high throughput. It also needs very low latency to enable seamless human interaction and realistic sense of togetherness.

Optimising a cutting-edge application with tremendously high requirements at such a large scale promises not to be an easy task. Nonetheless, it is a task that operators require in order to confidently guarantee provision of a service at a given quality.

1.2 Aim of this work

We aim to study what are the best radio access configurations to optimally serve social XR conferences. Here ‘configurations’ consist of all parameters that control how radio resources are shared, ranging from algorithms to simple constants. Of course, such configurations are very scenario-specific, and the best match to one application, channel state and network state seldom is the best match to another.

To achieve that we need to derive the impact of each configuration on the QoS for a given application. Logically, if it is known how each configuration impacts performance, it becomes trivial to choose the configurations that lead to the best performance. This is useful not only to optimise the physical deployment by saving costs, but also the management of that service, since ultimately the more efficient the provision of a service is, the less resources it requires to provide that service.

A solid way of obtaining insights about how such configurations impact performance is to model and simulate the application, the network equipment, the radio resource sharing mechanisms and the radio channel and then to measure the impact those configurations have on performance. This work aims to complete the first part consisting of modelling. The second part, which is based on extensive simulations, should be completed outside of this thesis.

More concretely, we introduce, implement and test a modelling framework for radio-layer optimisation and performance assessment in indoor social XR conferences. Such framework fills modelling gaps in the literature. We intend this work to be a stepping stone for future research on cellular communications, namely by providing a simulation environment not only for sensitivity analysis but also for development and testing of new management mechanisms, possibly AI-based.

Finally, although we are considering one specific application with a well-defined use-case, which we will clearly define and model the necessary components further ahead, we expect many of the conclusions to also apply to other applications. Furthermore, the methodologies here presented can be replicated to derive application-specific conclusions for different applications.

1.3 Outline

In Chapter 1 we have motivated the relevance of studying radio layer optimisations to improve the performance of an XR conference. We also integrated this study in a broader context by mentioning its applicability in management of future services in a virtualised and automatic manner.

Chapter 2 provides a solid background for the main contribution of this research, contained in Chapter 3. Firstly, in Section 2.1 we survey XR applications' requirements, packet traffic characteristic, aspects that influence the radio channel, namely human behaviour, and we look into optimisation attempts to VR applications performance from the physical layer perspective.

Subsequently, Section 2.2 we review current radio layer techniques and most promising technologies to achieve the demanding requirements. After, Section 2.3 presents how these techniques play a role in reality by examination of the relevant 5G physical layer standards and introducing radio access equipment, e.g. antenna systems. In Section 2.4 we survey radio channel simulators and find one that fits all our requirements. Lastly, the contributions we make to the state-of-the-art are listed.

Chapter 3 presents the Methodology. Here we disclose all modelling steps and assumptions. First we model the XR conference use case. We do so in Section 3.1 by addressing room sizes and how users are seated. Then model the antennas, user behaviour, and traffic. Section 3.2 we use the selected channel generator to assess how the propagation environment changes in light of application use case assumptions, such as user position and behaviour.

Next, in Section 3.3 we present all functions executed by the network equipment to enable data transmission. We start off by stating how channel state information is acquired, how to create a grid of beams and select the best beam. Then user scheduling is addressed, consisting of how channel quality and instantaneous throughput estimation is done. Finally, we present a flexible and general framework to assess the quality of the transmission, we compute errors and we save the relevant metrics to assure good decisions also in the next transmission interval.

In Chapter 4 we present results of an initial simulation study. To start, we clearly define the simulation in Section 4.1 in view of the parameters introduced in Chapter 3. Then we investigate and compare a single and multiple user scenario, respectively, in Sections 4.2 and 4.3. We also discuss the results and take conclusions throughout.

Chapter 5 we conclude, reiterate the most important results and suggest directions for future work.

2

Literature Review

Contents

2.1 Social XR Applications

This section provides background about the application use-case/service we intend to support. It is relevant for the remaining of this work since several of its aspects influence the propagation channel. Additionally, radio resource management aims to fulfil application traffic, therefore the traffic characteristics are of importance as well.

The service whose provision we aim to support, and eventually optimise, is social XR conferences. More specifically, first and foremost, we need to find what are the application requirements in terms of throughput and latency. Then, we survey literature for Uplink (UL) and Downlink (DL) application traffic models, like a random packet arrival distribution.

Furthermore, we require a model for user behaviour in terms of movement since user mobility influences the propagation conditions and radio channel variability, which are important factors to take into account during radio resource management procedures. And finally, it is useful to review attempts from other authors to optimise QoS provision over wireless for XR applications.

Let us start by clarifying the term XR. By definition, it refers to all real-and-virtual combined environments and human-machine interactions generated by computer technology and wearables. In essence, it includes Augmented Reality (AR), Virtual Reality (VR) and everything in between. Movies such as “Ready Player One” [17] anticipate what can be the future of these technologies; see Figure 2.1 for a snapshot of an AR meeting use-case taken from a movie.

Figure 2.1: AR meeting portrayed in movie ‘Kingsman: The secret service’. [3]

Regarding the state of the art in XR applications optimisation from the wireless perspective, the research topic has not been very active. There is no lack of reasons to research how to remove the wire that connects the Head Mounted Display (HMD) to the network, such as improving the immersiveness of experience and reducing the risk of tripping hazards. But there has not been much attention from the wireless communications research community, with the vast majority of well-cited papers dating no later than 2017 [18].

Requirements

Fortunately, with respect to requirements there has been plenty of speculation. Elbamby *et al.* [19] does a back-of-the-envelope calculation. He considers that each human is able to see up to 64 million pixels (having 150° horizontal and 120° vertical Field Of View (FOV), and resolution of 60 pixels per degree), at 120 Frames Per second (FPS) (required to generate a real-like view), thus resulting in up to 15.5 billion pixels per second of raw information. He concludes that even compressing the stream by 600 times with a state-of-the-art H.265 encoder, it requires 1 Gbps (gigabit per second) speeds to transmit if each coloured pixel (provided that is stored with a high resolution of 36 bits). Other authors [20] reach even higher numbers.

Of course, as a back of the envelope calculation it has its utility, but some important factors are missing. Namely, there exist many application-layer techniques to reduce the required bitrate, e.g. frame prediction in applications where 360° video is involved [21]. Also, 64 million pixels per eye equates to almost double the pixel count of an 8K screen (7680 by 4320 pixels). Therefore very high-end conditions are far from realistic in the near future. The current best display achieves 1700 by 1440 pixels per eye, which is less than 2K, with a refresh rate of 90 Hz [22]. Any display advertising higher numbers uses pixel interpolation [23] which hurts quality.

The throughput estimation can also be higher. High Efficiency Video Coding (HEVC), or H.265, is a complex set of algorithms that, as the name suggest, aims to optimise coding efficiency, i.e. to reduce the data as much as possible while keeping the quality imperceptible unchanged. These complex algorithms take many tens or even hundreds of milliseconds to encode, making them less suitable for real-time transmissions. Within the standard there are options that achieve lower encoding and decoding latencies, but at expense of compression ratio.

In summary, the application resolutions and frame rates differ considerably, and it is yet unknown what would constitute good Quality of Experience (QoE). Since [19] and [24] agree an entry-level VR would need around 100 Mbps, and considering 3rd Generation Partnership Project (3GPP) [25] 50-100 Mbps estimate (for the most common streaming strategy) and [20] proposal of 100-200 Mbps, we may settle for 100 Mbps of throughput requirements for the near-future VR experience, in order to establish a concrete first target for near-future VR.

To finalise the required throughputs requirements a streaming strategy needs to be defined and 3GPP [25] defines mainly two, viewport-independent (most common) and viewport-dependent streaming. The difference lies preparing the scene to send to a user independently of the user thus sending the full XR scene, or based on what the user is looking at, respectively. When the user point of view is considered, only the data that user requires can be sent, which should allow for a reduction in the required bitrate by a factor of two to four compared to sending the full XR scene.

Regarding latencies, the distinction between two latencies should be made. There is the Round Trip Time (RTT) latency and there is the Motion-to-Photon (MTP) latency. The latter is basically an headset requirement: from the time a movement occurs to the changes be reflected in the display there must not go more than 20 ms [19, 25]. The first, on the other hand, is on the order of several tens or hundreds of milliseconds and is the time it takes for the information to go from the user to the main XR server and an answer to come back to the user.

Two-way delay contributions include, sensor sampling, encoding, one-way network delay (router and access point processing delays, queuing delays, transmission delays and propagation delays), decoding, image processing algorithms on the cloud, encoding once again, another one-way network delay, decoding, local frame rendering and display refresh delay. Most of these delays are practically imperceptible compared to others. For instance, the delay from sensor sampling is less than 1 ms while the display delay tends to be 10-15 ms [20] although it is expected to drop to less than 5 ms [24].

Additionally, currently just the computation delay alone exceeds 100 ms [24]. However, by making a smart use of caching, bringing the processing power closer to the access (edge computing) and improving of communications [19, 20, 25] this delay is foreseen to drop to below 10 ms [24].

The target for RTT of 50 ms is given by [25]. [24] says 30 ms is a better value. Also, 3GPP defines in [25] a 5G QoS Identifier (5GI) of value 82 for AR applications mapping to QoS characteristics of 10 ms radio interface latency and 10^{-6} Packet Error Ratio (PER). Thus, we conclude that 10ms is an hard upper limit of accepted latency, and that the preference is to as low as possible.

Architecture and Capture System

The standardisation of architecture of the network for XR conference applications by 3GPP [25] is in agreement with [26]. Remarks relevant to our study lie in terms of traffic patterns and relationships between uplink and downlink. Figure 2.2 shows the simplified capture system and relevant architecture details.

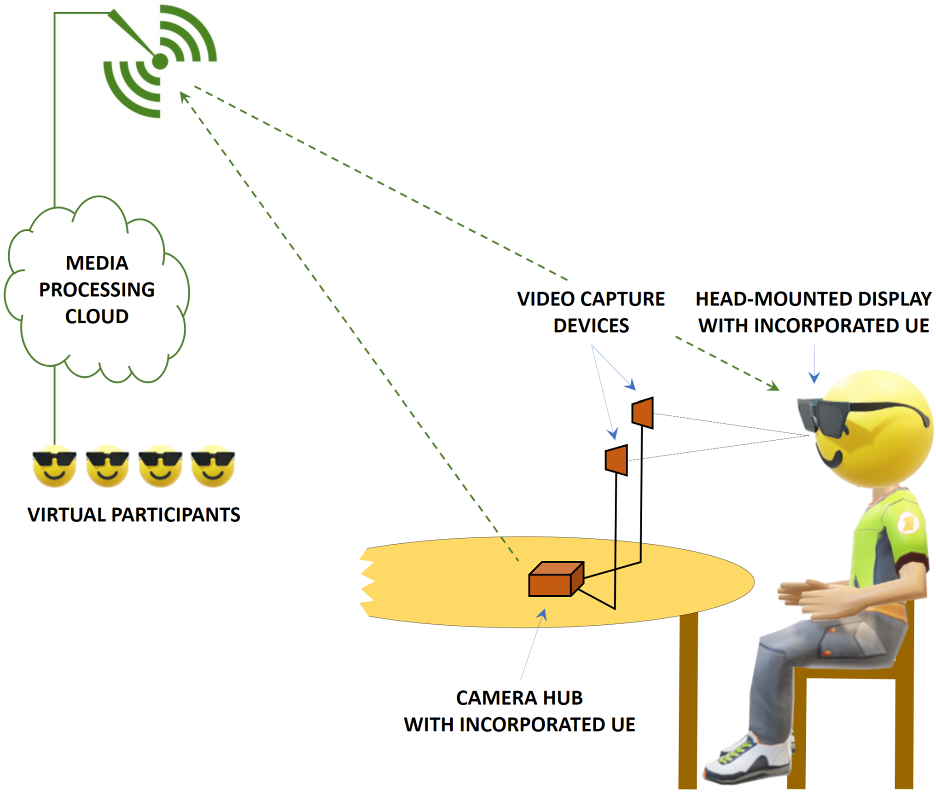

MEDIA PROCESSING CLOUD
VIRTUAL PARTICIPANTS
VIDEO CAPTURE DEVICES
HEAD-MOUNTED DISPLAY WITH INCORPORATED UE
CAMERA HUB WITH INCORPORATED UE

Figure 2.2: Simplified architecture and capture system for XR conference

There is a single uplink stream per user for pre-transmission stream synchronisation. Therefore, the information captured from both cameras is aggregated into one single stream before transmitted, which facilitates the synchronisation in the cloud. This aggregation should happen as close to the source as possible to avoid having the destination wait to receive both recordings of the same user with the same timestamps. Such loss of synchrony can happen when each camera sends its information separately due to the delays of different paths in the network.

Furthermore, at this stage, the information recorded constitutes the user perspectives. Although user perspectives recorded from different cameras can be aggregated into a 3D user representation making it more compact compared to simply stitching camera streams, this is infeasible in the near future due to latency constraints - the processing time required is very long. Therefore, each participant receives the information of each other participant, which is the information recorded by two (or more) capture devices, in case of viewport-independent streaming. With viewport-dependent streaming the users receive only the users they are looking at. Also, in case of an AR meeting, the users only receive information on the remote (non-physically present) users, since the physically present users can be seen through the AR glasses.

With respect to how the capture should be made, [27] tests and proposes a dualcamera setup where each camera can record Red Green Blue (RGB) and depth information. The cameras should be placed at head height, roughly 30 ° from the normal to the user. Exactly how information is stored and transmitted, being double RGBD (RGB plus depth) or 3D mesh or as a point cloud, is yet to be determined, despite playing an important role in the throughput and latency requirements [25].

Model for Human Head Movement and Traffic

To the best of our knowledge, there are no human head movement models in literature that suit a conference. Some authors have recorded head and eye movement for different scenarios of 360° video [28, 29]. However, not only does 360° video have a considerably different dynamic than a real-time conference, but none of the videos were remotely close to a meeting room.

Likewise, we did not find traces in literature of XR conferences traffic models. Traffic models on its own are relatively hard to find, and it has proven to be an impossible task for such a new application. After all, there still is no agreement [25, 26] in many important details of the application

Different Optimisation Approaches

In [30] is presented a framework that analyses the performance of VR services over wireless networks. The framework captures the tracking accuracy, transmission delay, and processing delay, but most radio characteristics such as frequency-selective fading (signal oscillations), antenna configurations and blockage effects are not considered. The authors of [31] study the impact of blockage by hand, head and body on wireless Millimetre Wave (mmWave) links, and suggest an algorithm to overcome the corresponding challenges. The proposed solution uses a fixed relay to increase robustness against blocking and is assessed in an experimental setup. The attainable gains strongly depend on numerous assumptions and deployment configurations which are not described in any detail.

Other more common approach to the problem of optimising VR meetings provision taken by other authors [32, 33] is to focus on the network perspective, and disregards the radio interface. One may conclude there is a clear lack of research on physical layer optimisation targetting virtual reality applications' QoS requirements.

2.2 Key Technologies and Techniques

This section presents how higher throughputs and lower latencies that far surpass what previous generations of mobile communications were capable of are achieved in today's wireless communications. Firstly we start off from a fundamental equation that relates bandwidth and spectral efficiency with throughput. Then we present the major key advancement for each term and make a connection to improved latency as well. Finally, we bridge to standardisation to show how these advancements are integrated in the current 5G standard.

Equation (2.1) summarises in a simple way how to increase throughput in a single-cell system. In essence, we either increase the amount of resources (bandwidth) or we increase how well we use the available resources. From 4th Generation (4G), there have been advancements in both domains and we will present them subsequently.

$$\text{Throughput (bits/s)} = \text{Bandwidth (Hz)} \times \text{Spectral efficiency (bits/s/Hz)}. \tag{2.1}$$

Regarding the first term, increasing the available bandwidth leads to performance gains and there are many examples as to why. Two examples are of particular relevance to show ahead how the increase in spectrum is exploited by the standards. Figure 2.3 shows a rectangular pulse in time and its respective footprint in frequency, a normalized sinc function. As such, we see that shorter pulses in time, i.e. smaller τ, require more bandwidth. Naturally, we want the pulses as short as possible to send as many in as little time, hence increasing the information transfer rate.

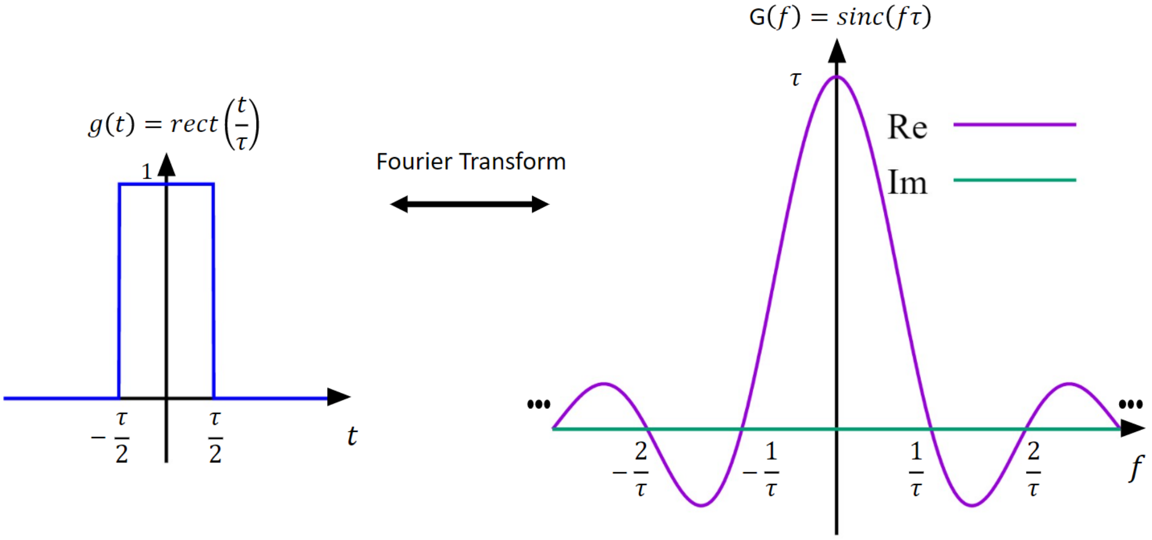


Figure 2.3: Rectangular pulse in time and equivalente. [4]

The second reason has to do with how resources are distributed to allow user multiplexing and multiple access. Current systems use several orthogonality techniques, like time orthogonality and frequency orthogonality. The latter means that the larger the bandwidth, the more data can be sent simultaneously, or more users can be server simultaneously, or both, therefore resulting in higher aggregated throughputs.

More Spectrum

The advancement consists in using higher frequencies. Frequencies between 30 and 300 GHz are part of the millimetre wave (mmWave) spectrum since its have wavelengths ranges from 1 centimetre to 1 millimetre, respectively. Frequencies from 24 GHz are also commonly included out of convenience. The mmWave spectrum is considerably less occupied than the spectrum below 6 GHz and yields more than 10 times the available bandwidth at a fraction of the cost [34]. And, more available spectrum permits higher bit rates and lower latencies.

One advantage of using the newly available bandwidth to make the transmitted signals shorter in time, besides taking less time to transmit, is shortening the time to interact. We will revisit this concept in the next section when we introduce the concept of numerologies which is how 5G New Radio achieves more agile transmission.

However, using higher frequencies also introduces some new propagation challenges [35, 36], listed below. Figure 2.4 illustrates some of the propagation terms.

- Rapid channel fluctuations - given the smaller wavelength, smaller spatial shifts cause. Mathematically, the coherence time (time during which the channel can be considered non-changing) is inversely proportional to the carrier frequency, therefore higher frequencies yield a more volatile channel [37]; The spectrum is more volatile due to the smaller wavelength. In other words, the quality of the signal fluctuates more due to multipath propagation;
- Susceptibility to shadowing - The waves diffract (bend around obstacles) less and the penetration loss (or material absorption) is higher [37]. Although penetration depends on the material, for most materials the absorption increases linearly with frequency. Similarly, the power diffracted reduces with frequency. Therefore, obstacles cause higher attenuations in higher frequencies;
- More scattering - rays scatter more since irregularities in surfaces are comparatively larger due to a smaller wavelength, therefore reflections are more diffuse [37]. Nonetheless, since there is less penetration, effectively more power is reflected [38];

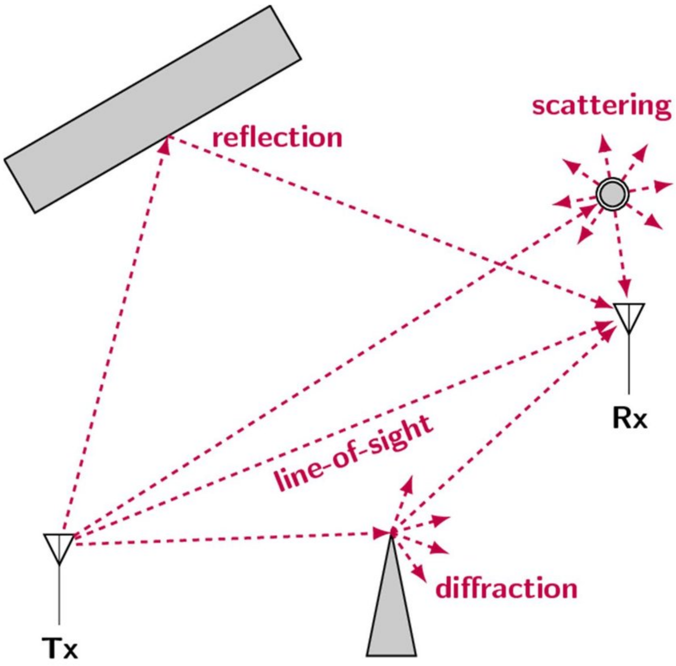


Figure 2.4: Illustration of propagation phenomena. [5]

In essence, the Line-of-Sight (LoS) path carries relatively more power than Non-Line-of-Sight (NLoS) multipath components, and this complicates the propagation when there is no LoS.

Lastly, there is one very important difference when using higher frequencies. Since the effective radiator size is proportional to the wavelength, the antennas in mmWaves will be proportionally smaller. This has two consequences: first, there will be an higher attenuation or path loss; second, there can be more antenna elements in the same physical area and this significantly compensates the higher path loss [36].

The Effect of Smaller Antennas

Let us introduce the Friis Equation [37] in (2.2) to carefully analyse this effect. We see the received power P_r is function of the transmit power P_t, the transmitter gain in the direction of the receiver G_t, the distance between the transmitter and the receiver d (since power spreads along the spherical surface with that radius), and the effective area at the receiver A_{er}, which takes into account how much of the energy present in the receiver's vicinity can be captured by its antenna.

$$P_r = P_t G_t \frac{1}{4\pi d^2} A_{er} \tag{2.2}$$

From antenna theory and reciprocity principles, the gain and the effective area of an antenna are fundamentally related. Equation (2.3) shows this relation. Note how the effective area is proportional to the square of the wavelength.

$$A_e = \frac{\lambda^2 G}{4\pi} \tag{2.3}$$

And applying (2.3) to (2.2) we obtain (2.4). This quick derivation is useful because it allows us to pinpoint exactly where the extra path loss at higher frequencies comes from. The direct dependence with frequency has to do with a smaller effective area, which is caused by a smaller antenna/radiator. Note that stepping from 3 to 30 GHz an half-wavelength dipole would get 10 times smaller leading to 100 times less effective area or 20 dB extra free-space path loss.

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2} \tag{2.4}$$

Fortunately, there is a major advantage of having smaller antennas. Smaller antennas allow us to form antenna arrays with more antenna elements than previously. As such, a tenfold increase in frequency allows a hundredfold increase in number of elements in the same physical area, which traduces to 20 dB extra directional gain per antenna array if all antenna elements are optimally used with one hundredth of the power. Therefore, if we increase the number of antenna elements of the receiver and transmitter and we can use each element optimally, we effectively improve the received power by 20 dB with the same transmit power.

The technique of making antenna elements constructively interfere in the directions of interest, and destructively interference in the direction where the signal causes harmful interference to other connections is analysed next.

Higher Spectral Efficiency

The most promising technology to enhance spectral efficiency is massive Multiple-Input Multiple-Output (MIMO) [39]. Massive MIMO consists of having at least 10 times more antennas than the number of users intended to be served simultaneously [40]. Statistically under Rayleigh fading assumptions, this leads to a high likelihood of obtaining independent channels to all users simultaneously. Essentially, this enables all users to be served simultaneously in the same time-frequency resources by spatial separating the streams. However, that may not lead to the best Spectral Efficiency (SE) [41], proving the relevance of modelling real scenarios.

Since Marzetta's seminal paper on the asymptotic results of increasing the number of antennas in BSs [42], massive MIMO has played a critical role in enhancing the performance of wireless systems. Figure 2.5 represents how increasing the elements plays a role in spectral efficiency by increasing the number of simultaneously served users.

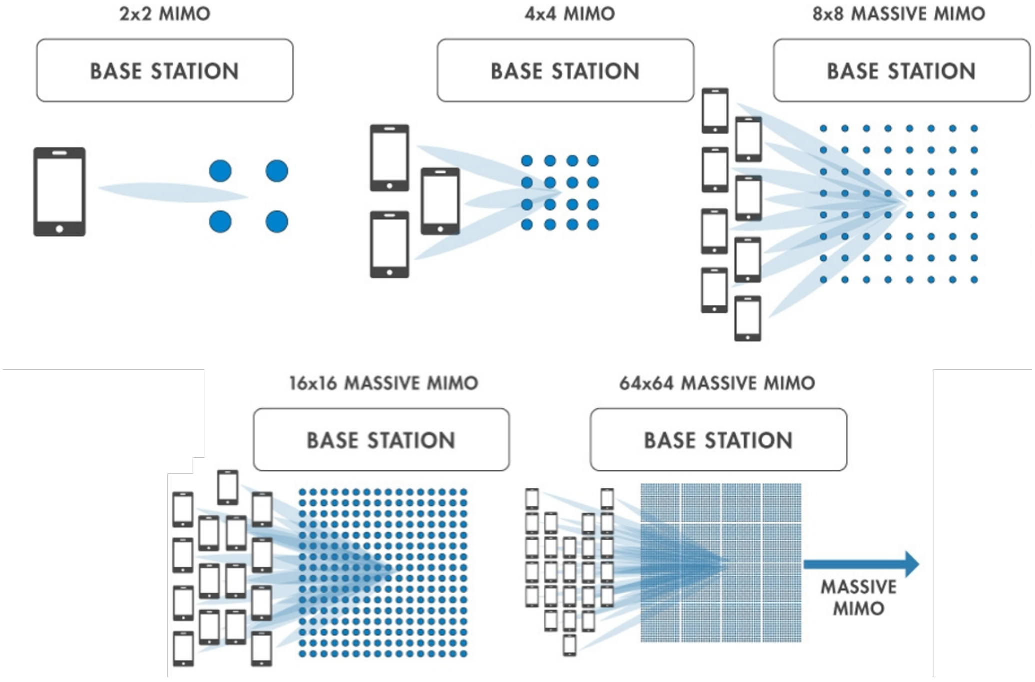


Figure 2.5: Antenna element impact on simultaneously served users. From Mathworks.

One important massive MIMO benefit is when small-scale fading (fast variations) between antenna elements is sufficiently uncorrelated, which is the case when AEs in arrays are separated more than half-wavelength, then the channel will tend to a deterministic state as the number of AEs increases in BSs or terminals. This happens because the more diversity, the less likely it is that all channels have big oscillations. In essence, the oscillations average out and this average with little to no oscillations dictates the channel. This effect is called channel hardening [43].

But the main selling points of massive MIMO are two. First, increasing the number of antennas increases the number of simultaneously served users, which can lead to increased aggregated throughput. Note that the total transmit power needs to be shared among more users, and serving users simultaneously may increase the interference each user experiences. However, the possibility of opting to serve more users when the conditions favour it guarantees higher aggregated throughputs in those occasions. Second, it increases the quality of servitude of each of those users due to beamforming gains [44], as we will explore ahead.

Regarding the increasing the independent data streams over the air, more commonly called layers, it explores space diversity between combinations of receiver and transmitter antennas. The more antennas, the more likely it is that a certain propagation path is sufficiently orthogonal to another.

When one or more data streams are sent to a single user, as in Figure 2.6, we call it SU-MIMO (Single-User MIMO) operation, and when different streams target different users, having one or more streams per user, we call it MU-MIMO (Multi-User MIMO) mode of operation. However, the system requires information about the channel, it needs to know the multipath information to decide which paths to exploit for transmission. Therefore, let us inspect how the channel is represented and how information about it can be acquired.

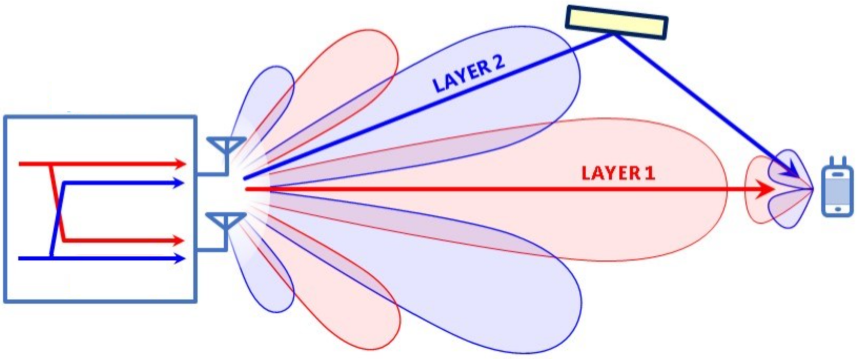


Figure 2.6: Example of single-user multilayer transmission. [6]

MIMO Channel

Figure 2.7 shows how the channel is seen from a system perspective. The complex scalars h_{ij} hold the amplitude and phase field transformations that occurs between the i-th antenna at the Transmitter (TX) and the j-th antenna at the Receiver (RX). They are called channel impulse responses or simply channel coefficients.

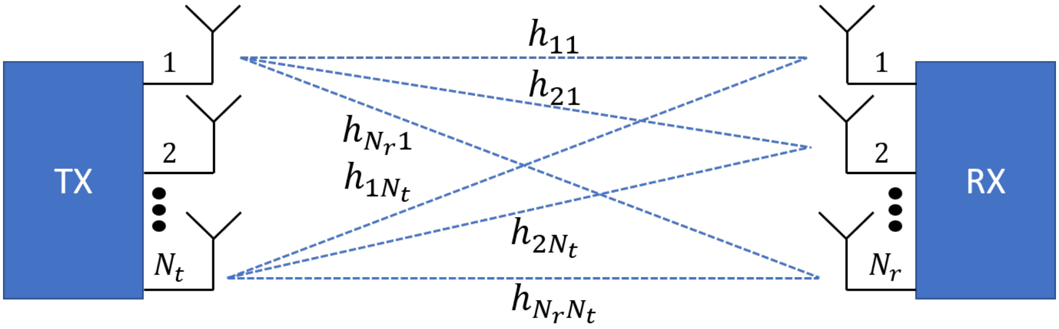


Figure 2.7: Mimo channel system representation.

Each channel coefficient has an amplitude and a phase. The amplitude is a positive real number smaller than one and shows how much the transmitted signal has been attenuated before it reaches the receiver. The phase tells us the phase difference between the transmitted and captured fields. Recall that the transmitting antenna excites a propagating disturbance in the electric and magnetic fields (an electromagnetic wave) and this disturbance propagates by oscillating having an associated phase. As such, the phase plays an important role in determining whether fields from different antennas interfere constructively or destructively, see Figure 2.8.

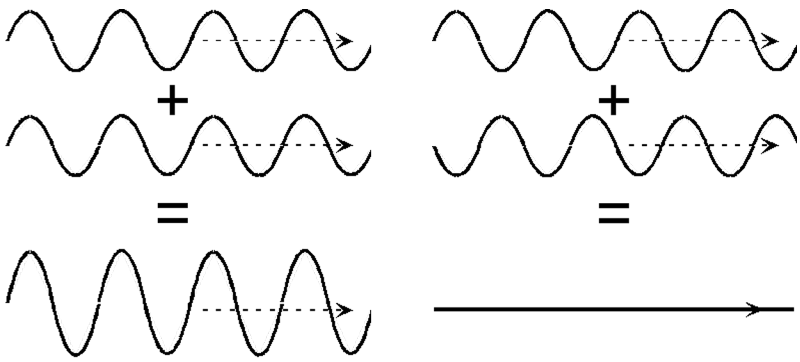


Figure 2.8: Constructive and destructive interference of waves. [7]

The channel matrix $\boldsymbol{H} \in \mathbb{C}^{N_r \times N_t}$ from Figure 2.7 is only valid for a specific time and frequency. It holds throughout the coherence bandwidth during the coherence time.

The precise definitions of these quantities require detailed channel knowledge namely about the exact powers and propagation delays of each path and maximum Doppler shift which is related with User Equipment (UE) speed [37, 45]. This knowledge is very hard to measure accurately in reality, so both the coherence time and bandwidth are estimated and parameters of the system are made to match the estimation, namely to the time duration and bandwidth of a Physical Resource Block (PRB) [46].

The channel matrix $\boldsymbol{H}$ relates the $N_r \times 1$ received signal $\boldsymbol{y}$ with the $N_t \times 1$ transmitted signal $\boldsymbol{x}$, plus the $N_r \times 1$ received noise $\boldsymbol{n}$. Equation (2.5) presents this relation.

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{2.5}$$

Therefore, to obtain an estimate of the channel matrix it is required to send known signals. In traditional massive MIMO formulations these signals (called pilots) are emitted by each single-antenna UE. Then the channel to each UE is estimated through linear algebra methods [47].

However, nowadays UEs have more than one antenna and each antenna needs to send a different signal to the BS. One can foresee complications due to an excessive need of reference signals that need to be orthogonal to prevent interference. Further ahead we will see how this is done precisely by visiting the 5G standards.

Additionally, it is not required that all the TX/RX antenna elements are located in the same place. Normally, antenna elements are only spaced half wavelength since it is enough to guarantee enough independence and also to reduce interference between elements. An antenna array has distances in this order between elements. But, one way of increasing the spatial diversity is to place antenna arrays spatially apart (several tenths or hundreds of wavelengths). This is called a multi-panel configuration [48].

Distinct panels can be used to jointly serve a user, enhancing throughput and coverage [49]. Allowing a user to access the network in different locations, and to be serve by one or a combination of access points, statistically improves the QoS and improves network efficiency by enabling more options for load balancing and interference mitigation [50].

We conclude that having more antennas allows serving more users because there will be more orthogonal paths encoded in $\boldsymbol{H}$ to be used for concurrent transmission. To exactly know how to perform those transmissions, we need to understand what beamforming is.

Beamforming

Beamforming is a signal processing technique. It consists of changing the amplitude and phase of the signal fed to each antenna element of an antenna array in order to create constructive interference in a particular point in space, thus improving the transmission or reception of the signal at that point. The same can be achieved to reduce interference by making the signals from each antenna destructively interfere. Furthermore, beamforming can be applied not only at the transmitter to enhance the signal at the receiver, but also at the receiver, to coherently combine the signals that each antenna captured. Normally, when applied to the transmitter, since the complex weights are multiplied to the signal before transmission it is called precoding, and by the same logic when applied to the receiver is called combining.

Signal superposition happens on a field level. Thus if the fields radiated by 100 elements add up constructively, the amplitude of the signal grows 100 times, which equates to 10000 times more power. Despite the ten-thousandfold array gain, the total gain will be one-hundredfold because we automatically reduce the transmit power of each element by 100 times (equal to the number of elements), in order to have the total array transmit power constant and independent of the number of elements.

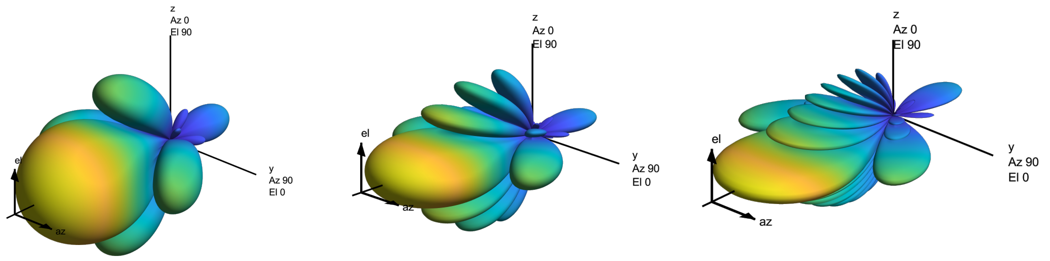
Table 2.1 shows the total gain (array gain plus element gain) obtained from beamforming with different antenna sizes with a Uniform Linear Array (ULA) - in this array the elements are placed in a line. We can see that as the number of elements N increases, the total power gain increases and the Half-Power Beam Width (HPBW) decreases - HPBW is the beamwidth between the points where the gain is 3 dBs below the maximum - i.e. the beams get narrower. Note that the increase in array gain by doubling the elements is 6 dB, but the element power gain decreases 3 dB because only half the power per element is available. The element used is the cross-polarised element described by 3GPP [51].

Since the table considers a linear (one-dimensional) array, the beamwidth measured

Table 2.1: Influence of element count in ULA on total power gain and HPBW

N	Gain [dBi]	HPBW [°]
1	8	64.98
2	11	44.17
4	14	24.44
8	17	12.53
16	20	6.29
32	23	3.06
64	26	1.31
128	29	0.52

is not the same in all planes - the beamwidth changes only in the plane along which the number of elements is changed, i.e. if the elements along the vertical increase, the beamwidth in a vertical plane decreases. Figure 2.9 illustrates the radiation pattern changing in the vertical plane according with the number of vertical elements. The array pattern further depends on inter-element spacing which is kept at half-wavelength throughout this thesis.



(a) 4 vertical by 4 horizontal **(b)** 8 vertical by 4 horizontal **(c)** 16 vertical by 4 horizontal

Figure 2.9: Radiation patterns for arrays with different elements along the vertical

From Table 2.1, Figure 2.9 and literature we take two important conclusions. First, we know the maximum gain of an array by the number of elements, Equation (2.6) summarises the gain progression from the table having as basis the number of elements and the gain of a single element G_{ele}. Secondly, we know the shape of the main beam from the antenna geometry - we just need to count the elements along a given direction and consult the corresponding line in the table to know the HPBW in the plane that contains that direction and the orthogonal to the array plane.

$$G_{total} = G_{ele} + 10\log_{10}(N) \text{ [dBi]} \tag{2.6}$$

More importantly, we know the beamforming gain is directly proportional to the number of antenna elements of an antenna array. Transmissions with enhanced directivity allow more power at the receiver and less power in other directions, decreasing the interference. Reception with beamforming also permits receiving more of the supposed signal and suppress sources of interference. Therefore, beamforming makes transmissions more efficient by increasing the Signal-to-Interference-plus-Noise Ratio (SINR) thus allowing higher Modulation and Coding Schemes (MCSs).

Antenna Architectures for Beamforming

Beamforming flexibility, like the number of users served simultaneously and the accuracy of directions to focus power, is directly connected with the antenna architecture. After all, independently of how good the software is, it is always limited by the hardware. Let us analyse what challenges the hardware poses before diving into the signal processing techniques.

With the increase in number of antenna elements, strategies to reduce the cost begin to develop [52]. Figure 2.10 shows three generic TX side architectures, each with a different associated cost and flexibility. The only differences to RX side architecture is the presence of an Analog-to-Digital Converter (ADC) instead of a Digital-to-Analog Converter (DAC) and the signal direction, represented by arrows, is reversed.

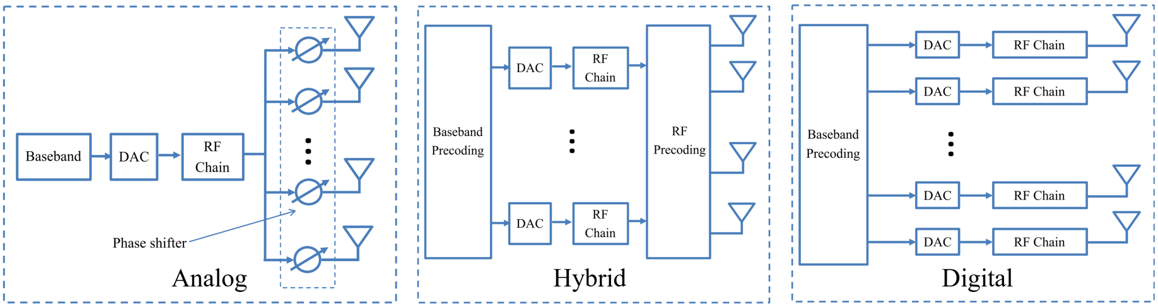


Figure 2.10: Transmit side antenna array architectures.

We need to understand what some basic components do to understand how they impact cost and flexibility. There are essentially three sets of components:

- The Baseband (BB) unit is where digital signal processing happens, at the natural frequencies of the signal, before they are used to modulate a high-frequency carrier;
- The TX chain is the physical processing chain used for transmitting signals. It comprises a DAC and Radio Frequency (RF) chain consisting of filters, mixers and amplifiers. In today's cellular antenna systems, it is often accompanied by a RX chain, consisting of the equivalent components for reception. The pair of two chains is commonly called Transceiver (TRX) or Transmit-Receive chain. Analog architectures only have one TRX chain, digital have as many as antenna elements and hybrid have more than one but fewer than digital;
- Phase shifters are used to add phase differences between antenna elements (represented by triangles) in the architectures where the same signal is fed to more than one antenna element, i.e. analog and hybrid. Digital architectures do not need phase shifters since the phase differences between individual elements are applied in baseband.

Regarding costs, the cost of a BB unit rises with the number of antenna elements due to the required processing power, and with the number of digital ports to TRX chains. But only the latter changes across architectures, although negligibly. Phase shifters cost rises only with frequency. However, both are insignificant compared with the cost of a full TRX chain.

With respect to flexibility, in the analog case, the same signal is fed to each element with the exception of a phase difference. This still allows for beamforming, but only one direction at a time since all signals are fed to the same phase shifters and thus directed to the same place. The digital counterpart is the most flexible; the signal can vary in amplitude and phase across elements. Therefore, it is able to direct in a specific manner as many signals as elements, since each signal needs its own TRX chain, e.g. to be connected to all elements.

The hybrid option provides a trade-off between the previous two. It has more TRX chains than analog, so it can send more simultaneous signals, but it looses in steering capabilities. Moreover, it requires connecting each RF chain to each antenna element with phase shifters in between, thus involving numerous connections and numerous phase shifters. In not fully-connected variants, hybrid beamforming looses even further beam steering flexibility, ultimately degenerating to the analog case.

In essence, the more TRX chains the antenna has, the more flexible and expensive it is. In industry, a 64T64R refers to the number of TX and RF chains, namely 64 of each. An array with 128 elements but only 64 TRXs, has an hybrid architecture with two-element sub-arrays, i.e. two antenna elements connected to each TRX. Subarraying is analysed further in Section 3.1.2. This is how beamforming flexibility is quantified in terms of hardware. Let us address software possibilities now.

Types of Beamforming

Beamforming is not a new technique [53] and has had applications in many fields such as radar, sonar, seismology, radio astronomy, acoustics and biomedicine. In today's mobile communications, it plays a crucial role. Expectedly, not all processing techniques are useful in all fields. So we distinguish the two different types of beamforming and survey the most useful techniques used in wireless communications.

The two types of beamforming used in wireless communications have several equivalent denominations but the difference is simple:

- Closed-loop / explicit / codebook-based / pre-determined / quantized / feedback-based beamforming - the possible beams are fixed and pre-established. The BS encodes reference signals in a few beams it thinks are most likely the best for a given UE, and the UE explicitly reports a feedback message stating which beam carried the reference signal received with the most success. This process serves to pick from a codebook, or Grid of Beams (GoB), which beam to use. The feedback overhead is smaller than its counterpart. The resultant beam is not a perfect fit to the channel, it represents instead a trade-off between optimal performance and prohibitive amounts of overhead.

- Open-loop / implicit / non-codebook-based / free-format / eigen / reciprocity-based beamforming - the range of possible beams is infinite and the actual beam is implicitly derived directly from the transmitted reference signals. This mode of operation does not use codebooks. It is based on the UL of orthogonal reference signals (pilots) to estimate the channel matrix. Multipath information is implicit in the channel matrix and vector of weights to use is derived with signal processing techniques. Note the reciprocal nature of this approach when UL and DL beamformers can be derives from the same process. The resultant beam optimally fits the channel measurements with respect to the quantities we aim to optimise with our processing techniques.

Take the following example to solidify the difference. Assuming the only propagation path from transmitter to receiver is the LoS, with feedback-based beamforming reference signals are sent (e.g. in three directions, -25° in elevation and 33° , 34° and 35° in azimuth) and awaits feedback on which beam suits the UE best, while with free-format beamforming the best direction to beamform (e.g. -24.5443° elevation and 34.1222° azimuth) is extracted from the channel matrix derived from reference signals coming from each of UE's antennas.

Conventional beam-steering [11] is a direction-based technique for feedback-based beamforming. With multipath propagation, since the beam is steered to one direction only, it is expected to perform less optimally due to focusing all energy in a single path [54].

In Appendix A we make an integral derivation of the complex weights to be applied to each antenna element in order to conventionally steer the signal to a certain direction. Nonetheless, it is worth surveying implicit beamforming techniques since they can be employed alongside explicit beamforming, and is motivated in the canonical massive MIMO formulation.

The most common implicit beamforming technique is Maximum Ratio (MR) [55]. When MR beamformer is applied to transmission it is called Maximum Ratio Transmission (MRT), and Maximum Ratio Combining (MRC) when applied at the reception. Equation 2.7 shows this computation of MR by calculating the Hermitian (conjugate transpose) of the channel vector and normalising such that the weights vector has unitary norm. Normalisation serves to prevent power scaling in the mathematics as we do not want to modify the total transmitted/received power with beamforming.

$$\boldsymbol{w}^{\mathsf{MR}} = \frac{\boldsymbol{h}^{\mathsf{H}}}{|\boldsymbol{h}|} \tag{2.7}$$

Observe that $\boldsymbol{h}$ is not a matrix. This is because MR can only optimise the transmission/reception to/from one point, therefore $\boldsymbol{h}$ is N_t by 1 in case of transmission, containing the coefficient that connect each of the transmit antennas to one of the receiver's antennas. And $\boldsymbol{h}$ is N_r by 1 when computing the MRC.

Currently, we have surveyed the most promising principles to cope with demands. Let us now analyse the relevant standardisation efforts to show how such principles are currently used in 5G access networks.

2.3 5G New Radio Physical Layer

5G NR is a new Radio Access Technology (RAT) developed by 3GPP for the fifth generation mobile communications network. In this section we present all the relevant radio access physical layer aspects in order to align our modelling decisions compatibly with the industry standards.

According to Qualcomm [56], the biggest 5G-compatible smartphone chips manufacturer, the five wireless inventions that define the global 5G standard are:

1. Scalable OFDM numerology with variable subcarrier spacing;
2. Flexible slot-based framework;
3. Advanced Low-Density Parity Check (LDPC) channel coding;
4. Massive MIMO;
5. Mobile mmWave;

All inventions play a crucial role in the radio access process. Therefore, let us organise this section according to the list above.

Scalable OFDM

Orthogonal Frequency Division Multiplexing (OFDM) is a digital multi-carrier modulation scheme. Rather than transmitting a high-rate stream of data with a single carrier, OFDM makes use of a large number of closely spaced orthogonal subcarriers that are transmitted in parallel. Each subcarrier is modulated with a conventional digital modulation scheme, such as Phase Shift Keying (PSK) or Quadrature Amplitude Modulation (QAM).

Figure 2.11 shows an example of OFDM waveforms. Note how the subcarrier nulls line with the peaks of neighbouring subcarriers. If each subcarrier is sampled at its peak, no trace of other subcarriers can be found, i.e. there is no interference.

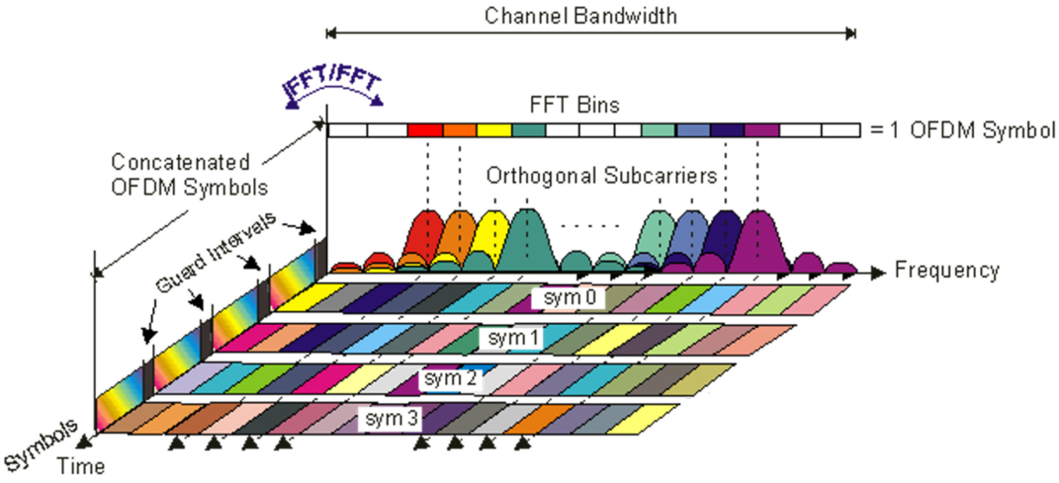


Figure 2.11: Example of OFDM-based system represented in time and frequency [8]

In the new air interface, OFDM is used for multiple access separating users in different time-frequency resources. To achieve this feat, resources are divided in blocks called PRBs. The symbol duration is equal to the inverse of the subcarrier spacing plus roughly 7% of this value for guard time. The duration of the PRB is equivalent to the duration of a slot and this quantity is often called the Transmission Time Interval (TTI) because it is the minimum division in time to perform operations.

The values for subcarrier spacing/width and PRB duration in 4G are 15 kHz and 1 ms, respectively. In 5G however the subcarrier spacing value is variable. As such, it is possible to use larger subcarriers to achieve quicker transmissions. The indices for given subcarrier spacing values are called numerologies and Table 2.2 shows the different numerologies and the respective subcarrier spacings and slot durations.

Table 2.2: Numerology influence on subcarrier spacing and slot duration.

Numerology	Subcarrier Spacing [kHz]	Slot duration [μs]
0	15	1000
1	30	500
2	60	250
3	120	125
4	240	62.5

Slot-based Framework

The coordination of UL and DL transmissions is dependent on the slot structure. However, differently from 4G, this structure can change at the end of each transmission period, and the duration of this transmission period can also change [57]. Figure 2.12 shows how the slot structure is defined in terms of DL slots, the composition of the transition slot in DL, guard and UL transition symbols, and UL slots at the end.

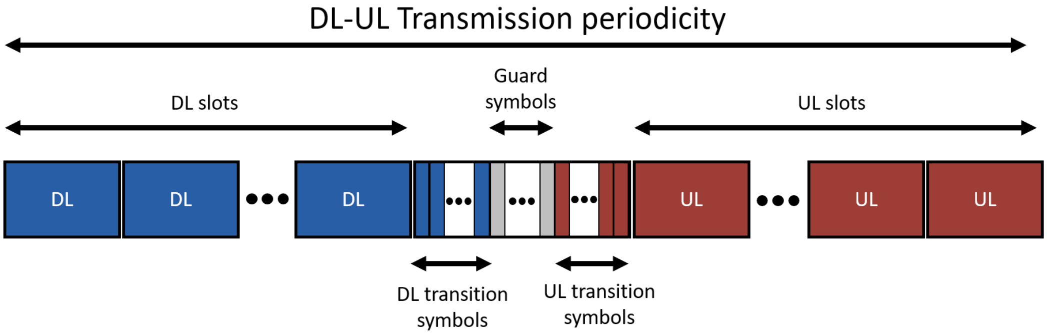


Figure 2.12: Slot structure.

Moreover, each slot can have numerous different formats where each symbol is either used for UL, DL or in a Flexible (F) manner. DL/UL slots are used for carrying DL/UL or F symbols. Figure 2.13 represents two of the least creative but most used options, however practically any combination of DL, UL and F symbols is available in the standards [57].

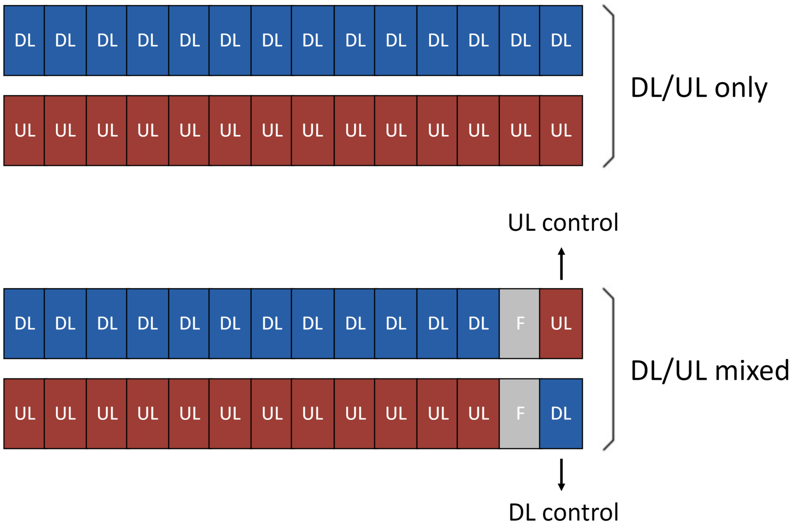

DL DL DL DL DL DL DL DL DL DL DL DL DL DL
UL UL UL UL UL UL UL UL UL UL UL UL UL UL
DL/UL only
UL control
DL DL DL DL DL DL DL DL DL DL DL DL F UL
UL UL UL UL UL UL UL UL UL UL UL UL F DL
DL/UL mixed
DL control

Figure 2.13: Slot format.

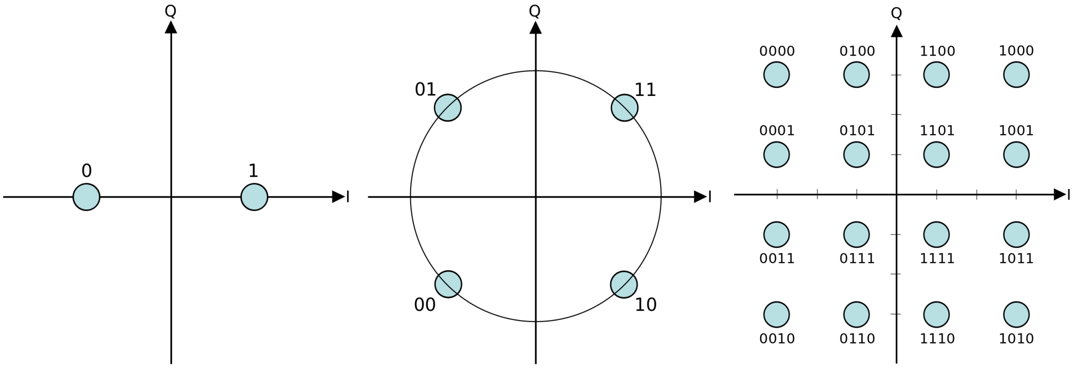
On the bottom of Figure 2.13 is a format that can be used for self-contained slots, where the F symbol is used as a guard interval. Self-contained slots are meant to enable a quick ACK/NACK feedback in low-latency communications, or for massive MIMO UL pilots [56]. The guard symbols between UL and DL are needed for synchronisation and their numbers can be smaller for shorter distances. In indoor applications, it is sufficient to use a single guard symbol [58].

In essence, the standards allow DL and UL assignment to change on a symbol level, thus making transmissions more efficient since resource needs can be met more precisely, with less resource waste by excess [59].

Modulation and Coding

We have so far surveyed how transmissions are organised in time and frequency. Now we review the different ways of coding and modulating the data streams into transmissions. The MCS influences the rate at which one can convey information.

Streams of data are organised in groups of bits to be encoded into symbols. More bits are sent per symbol with higher modulation orders. Figure 2.14 shows the In-phase and Quadrature (IQ) plane constellation diagrams for Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK) and 16-QAM. 5G additionally supports 64-QAM and 256-QAM. Each symbol encodes $\log_2(M)$ bits, where M is the modulation order - respectively, 1, 2 and 4 for the modulations in the figure.



(a) BPSK **(b)** QPSK **(c)** 16-QAM

Figure 2.14: Constellation diagrams for common digital modulations. [9]

More concretely, different symbols have different phases (e.g. Figure 2.14b) or a combination of different amplitudes and phases (e.g. Figure 2.14c) where the distance to the origin marks the amplitude and the complex argument holds the phase. Moreover, the receiver will perceive a different symbol, i.e. in a slightly different place in the IQ plane, than it was transmitted (e.g. due to noise). Thus, the closer symbols in the IQ plane supposedly are, the more likely they are to be mistaken by other symbols, and we see that symbols get closer with higher modulation orders.

Several combinations of modulation and coding schemes are available. However, it is required to choose the appropriate scheme that best fits the quality of the channel. Table 2.3 holds the MCS that correspond to a certain Channel Quality Indicator (CQI) that is reported when the channel state is measured to select what MCS to use. Moreover, the table contains the transmitted number of information bits (i.e. excluding coding) per symbol in the 'Efficiency' column.

Table 2.3: CQI Table 5.2.2.1-3 from [12]

CQI index	Modulation	Code rate x 1024	Efficiency
0	out of range		
1	QPSK	78	0.152
2	QPSK	193	0.377
3	QPSK	449	0.877
4	16QAM	378	1.477
5	16QAM	490	1.914
6	16QAM	616	2.406
7	64QAM	466	2.731
8	64QAM	567	3.322
9	64QAM	666	3.902
10	64QAM	772	4.523
11	64QAM	873	5.115
12	256QAM	711	5.555
13	256QAM	797	6.227
14	256QAM	885	6.914
15	256QAM	948	7.406

We have addressed multi-layer transmissions previously, however there are constraints regarding how those layers are coded and modulated. As such, we need to define more carefully terms such as QoS flow, codewords and layers.

The Physical Downlink Shared Channel (PDSCH) is the main downlink data bearing channel and is allocated to users on a dynamic and opportunistic basis. The Physical Uplink Shared Channel (PUSCH) is its UL equivalent. Such shared channels carry data in Transport Blocks (TBs) which correspond to a Medium Access Control (MAC) layer Protocol Data Unit (PDU). They are passed from the MAC layer to the Physical (PHY) layer once per TTI.

The TBs from the QoS flow have a Cyclic Redundancy Check (CRC) added to them. Then they are segmented into smaller chunks called Code Blocks (CBs) and each of those chunks has its own CRC. These CBs are then coded with LDPC which is a code that outperforms polar and turbo codes achieving higher code efficiency at considerable lower complexity and with less implementations challenges [56]. Finally, the CBs are put back together. Figure 2.15 represents the transformation from TBs to codewords described in this paragraph.

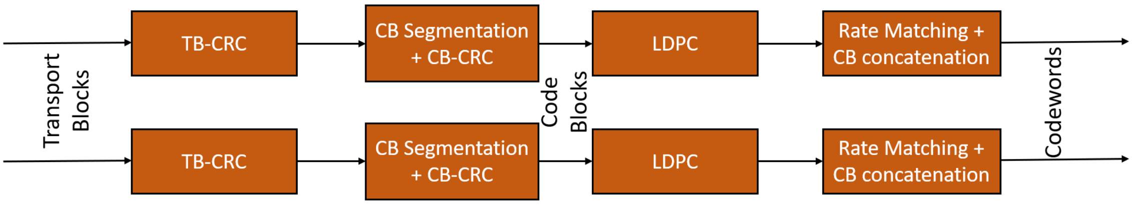


Figure 2.15: Block diagram of channel coding chain at the transmitter.

A codeword is scrambled before it is modulated so as to give the data to be transmitted useful properties that facilitate demodulation. After modulation, each codeword is separated into a maximum of four layers. Then each layer is precoded. Frequently, layers are called independent data streams therefore attention to context is required.

Posteriorly, the precoded signals are mapped to the respective Resource Elements (REs). Reference signals, in particular, require this process so that the receiver knows where that reference signal can be found. Each RE consists of one subcarrier used for the duration of one OFDM symbol. Given a PRB, by definition, is the use of 12 subcarriers for the duration of 14 OFDM symbols, each PRB consists of 168 REs.

Then, OFDM symbols are generated and mapped from the logical antenna ports to the physical antenna ports. Such mapping depends on the manufacturer antenna geometry. We review these considerations in more detail the next section.

Figure 2.16 summarises the chain of events from codewords to layers to actual transmission in physical antennas. Although both Figure 2.15 and 2.16 have transmitter-side block diagrams and descriptions, the receive side is the same, only reversed.

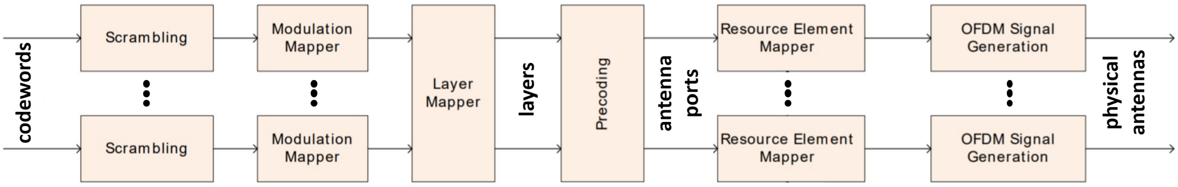


Figure 2.16: Block diagram at the transmitter: from codewords to physical antennas.

There can be a maximum of two codewords per user. The maximum number of transmitted layers also has a limit at eight layers per user when the user is served alone (SU-MIMO) and two layers per user when more than one user is served simultaneously (MU-MIMO) [12].

Massive MIMO

As mentioned, 5G heavily relies on massive MIMO, and thus on beamforming, to increase its spectral efficiency. Therefore, the performance massive MIMO systems can achieve in 5G is closely related to how beams are created.

There are essentially two types of beams in 5G: access beams, for the initial access to the network, these are broad beams designed to cover large sections of the cell; and traffic beams, which are more directive beams made to serve UEs.

The access beams, or Synchronization Signal Block (SSB) beams, carry synchronisation signals and the cell ID [60]. The access procedure is based on beam sweeping where all the SSB beams in a GoB are used, one at a time, and the UE chooses the cell of the best received beam. Given that we intend to analyse application performance, there could be cases where access is important, e.g. when the connection breaks and a new access is required. However, we want to avoid loss of connection since it would reflect in major performance dips. We focus on managing the already established traffic beam such that connectivity is never lost.

After the access, Channel State Information (CSI) is required to manage the connection with UEs. 5G's CSI framework holds two major phases [61]. The first is the beam management phase where the beams to be used by the BS and UE are derived. The second is the CSI acquisition phase where, for instance, CQI are reported and the MCSs are chosen.

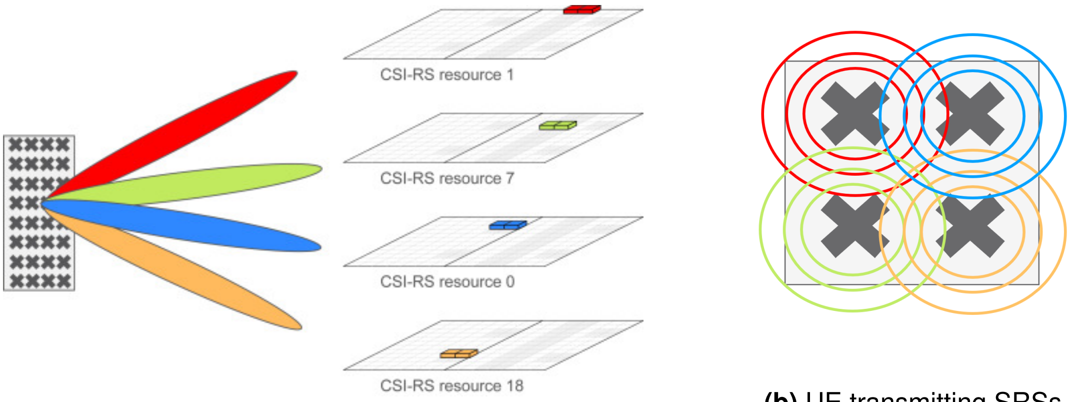
Beam Management

Beam management consists of tuning the beams used to serve UEs in a way that channel quality improves, often consisting of optimising beam direction and increasing its directivity. There is a clear trade-off between beam directivity and robustness since more directive narrower beams can also fall out of alignment more easily causing significant decreases in signal quality or even resulting in loss of connectivity. Beam alignment requires accurate beam tracking based on channel knowledge.

There are two ways of performing beam alignment in 5G, based on different Reference Signals (RSs) [61]:

- Channel State Information - Reference Signal (CSI-RS) - downlink RS. Each RS is sent in UE-specific time-frequency resources and beamformed in a specific direction, as shown in Figure 2.17a. Upon reception, the UE reports the IDs, and optionally the powers, of up to the four best received CSI-RSs, in a feedback quantity called Channel Report Indicator (CRI). To send the report, the UE transmits with the optimal beam to received the best CSI-RS, unless it had been previously instructed by the BS to use another beam. Alternatively, the UE can feedback a Precoding Matrix Indicator (PMI) suggesting a beam from the GoB based on the measurements performed on the received CSI-RSs;

- Sounding Reference Signal (SRS) - uplink RS. Each RS also uses well-defined REs, but contrary to the CSI-RS, each antenna sends one, see Figure 2.17b. Upon reception, the BS derives a beam to transmit to the UE, which then either derives the best beam for reception and uses it for transmission too, or uses the beam instructed by the BS [62];



(a) BS transmitting CSI-RSs [63]

(b) UE transmitting SRSs

Figure 2.17: Transmission of reference signals for beam management.

To bridge the gap to the two types of beamforming defined in Section 2.2: codebook-based beamforming mentioned is CSI-RS-based, while the non-codebook-based beamforming is SRS-based. Naturally, since each is based on a different reference signal and involve different procedures, they effectively result in different beams, exactly like the two types of beamforming do. Note that exactly as feedback-based beamforming, beam management with CSI-RS is based on partial channel knowledge, deriving information only about the best beam. Conversely, SRS-based beamforming, much like free-format beamforming, requires full channel knowledge, i.e. the channel responses from the UE antennas.

Channel State Information Acquisition

The CSI acquisition is the same for both cases of beam management. It consists of the BS sending a CSI-RS, the UE deriving CSI and reporting it back to the BS. However, using beam management that is CSI-RS based, both procedures can be compressed in one, and the UE reports the best beam IDs in the CSI feedback.

As seen, beam management and CSI acquisition procedures are not instantaneous nor can be performed in a single time slot. Therefore, there is a delay between CSI measurements and usage. For example, a beam is derived in certain conditions but it is used a few moments later, and the conditions may have changed.

There is a good reason why CSI-RS is used for acquiring CSI but SRSs are not. SRSs need to be transmitted in orthogonal sequences across all UEs such that they do not interfere with one another at the BS. The CSI-RS, on the other hand, only has to be orthogonal across the UEs that could receive that CSI-RS in that RE, and there may be a single UE in that list. Therefore, more CSI-RSs can be sent per UE since they require less orthogonality. More precisely, 32 CSI-RSs can be scheduled for a UE, but only four SRSs can be sent in the uplink by each UE, and the flexibility of using CSI-RSs is greater due to less orthogonality constraints [62]. Also, for the case of interference measurements, they can only be measured at the receiving end.

In essence, massive MIMO works with both types of beamforming and the standards also allow both. The academia has supported SRS-based beamforming, while industry has been waging primarily CSI-RS-based [64], judging also by the 3GPP standards [12]. And contrary to the academia traditional massive MIMO formulations of having UEs with a single antenna and focusing all complexity at the BS side, the tendency is the growth of antenna arrays in UE as well. An example is the modules of 64 dual-polarised antenna elements for mmWave communications in the latest 5G smartphone chips [65]. And such high number of antenna elements makes channel sounding impractical, thus rendering reciprocity-based massive MIMO less likely.

The uplink of an SRS allows the BS to have information on up to four distinct channels to the UE, because up to four orthogonal SRS sequences are available per UE. Therefore, the UE may use up to four antenna ports, each antenna port mapping an SRS sequence to a set of physical antennas. This means that if the UE has four or less antennas, it can send a different SRS per physical antenna and the BS may estimate the channel to every single antenna of the UE. However, when the number of antennas on the UE is greater than four, it will not be possible for the BS to know the transformation that occurred between each UE antenna and each of its antennas. Many authors [66] consider an infinite amount of orthogonal SRSs available to be sent by each UE. However, when there is a limited amount of orthogonal sequences allowed per UE, the spatial filter applied to map one SRS to several physical antennas will be part of the transformation on the SRS, and since the BS is only be capable of inverting the complete transformation that occurred on the reference signal, it cannot target individual antennas. As a result, the performance of reciprocity-based massive MIMO will fall short the asymptotic results in academia.

The interference that results from excessive uplink of SRSs goes by the name of pilot contamination. Although there are some strategies to mitigate it [67, 68], they do not work well with so many UE antennas. Moreover, massive MIMO is supported both in Time Division Duplex (TDD) and Frequency Division Duplex (FDD) modes [69], but in FDD only feedback-based works due to the lack of reciprocity of having UL and DL in different frequencies. Therefore, considering the standards, massive MIMO with a GoB seems to be the most promising direction.

Mobile mmWave

We overviewed how beam-tracking works in 5G, which will play a crucial role in supporting mobility. It becomes an harder challenge in mmWaves where antenna dimensions make it easier to have high number of antennas, thus allowing for much more directive beams. Let us see what are the standardised differences between mmWave and lower frequencies.

There are two Frequency Ranges (FRs) in 5G. The specific frequencies of each range are represented in Table 2.4, but not all numerologies are available in all bands. TDD and FDD operation are band-dependent as well.

Table 2.4: Differences between Frequency Ranges in 5G.

	Frequency Range 1	Frequency Range 2
Frequencies	410 - 7125 MHz	24250 - 52600 MHz
Numerologies	0, 1, 2	2, 3
Duplexing	TDD, FDD	TDD
Overhead	UL: 0.08 / DL: 0.14	UL: 0.10 / DL: 0.18

Although FR1 is commonly called ‘sub-6 GHz’, the upper limit has been increased to around 7 GHz [70]. Additionally, Table 2.4 contains the signalling overheads of each frequency range for UL and DL, in accordance with 3GPP [71]. One reason overhead in mmWave are bigger has to do with more feedback needed for beam management and CSI acquisition since mmWave channels change faster.

In FR1 the standard allows practically all frequency bands. In FR2 however, there are more specific bands. Roughly speaking, portions of spectrum are currently considered in the standards: from 24 to 30 GHz and from 37 to 43 GHz, depending on the location around the globe. In Europe, the assigned mmWave band is 24.25 to 27.5 GHz [13].

2.4 Propagation Channel

We aim to model the propagation channel for an indoor environment. In this section we present and justify requirements and choice of channel model. Modelling the radio channel is needed because the quality of connections is directly related with the quality of the channel. Moreover, the radio access network equipment relies on several management mechanisms to optimise link given different propagation conditions. As such, modelling those conditions is a must.

Traditional models are not sufficiently precise to our application [72]. For instance, we need a channel model that takes into account 3D radiation patterns of antenna arrays (for massive MIMO) and the majority of traditional models only use the maximum gain of the antenna. We justify the requirements, survey literature for a channel model that fulfils them and weight the most important considerations regarding our choice.

Firstly, with the increase in frequency and the decrease in proximity between UEs and BSs, there is a concern about the commonly used far-field approximations in propagation equations incurring in significant errors. Therefore, we assess whether spherical waves can be assumed as plane waves.

Near-field Verification

The Rayleigh or Fraunhofer distance d_F corresponds to the distance of the interface between the Fresnel and Fraunhofer regions, respectively, the near-field and the far-field regions. It is defined in Equation (2.8). If any of our TX-RX pairs are distanced less than d_F, then near-field equations must be used.

$$d_F = \frac{2D^2}{\lambda} \tag{2.8}$$

Above, D is the largest dimension of the radiator or radiator array, namely the diagonal in square arrays, and λ the wavelength. Using Equation 2.8 for a frequency of 30 GHz (to get a small wavelength of 10 mm), and using the conservative value of $D = 20$ cm, we obtain $d_F = 8$ metres, which may be beyond room dimensions, thus making the whole room inside the near-field zone.

Some authors propose more complex approaches to make this decision [73], but the proposed thresholds lead to higher decision distances, therefore we can be sure that support for spherical waves is necessary.

Requirements and Choice

In summary, taking into account previously identified requirements, the channel model must support:

- Spherical waves
- Indoor scenarios
- 3GPP compliant
- Time-evolution
- Massive MIMO
- Sub-6 GHz and mmWave frequencies
- 3D antenna and propagation modelling
- Spatial consistency

The requirement yet to justify is spatial consistency. Spatial consistency of slow-fading comprises having large scale signal oscillations correlated in space since in actuality similar positions in space have correlated propagation conditions.

Furthermore, the model should be implemented, instead of only providing the guidelines for implementation [74]. And it should be open-source as one needs to know what happens *under the hood* at all times, it facilitates reproducibility and integration with remaining components.

Our choice is facilitated with an extensive survey of more than 50 channel models for 5G [75]. This survey characterises models with respect to modelling approach (e.g. stochastic or deterministic), compatible frequency range, support for large arrays, spherical waves, mobility, blockage, gaseous absorption, among others. And searching for our requirements, the only channel model that checks all boxes is Quadriga [76], a Geometry-Based Stochastic Model (GBSM) widely accepted by the community. Refer to [75] for an in-depth comparison of channel models for 5G. Nonetheless, we justify our choice further.

One of the main decisions while choosing a channel model is between deterministic ray-trace-like approaches and stochastic approaches based on channel measurements. The first is more precise however more computationally demanding, contrary to the second. Given the complexity involved with developing deterministic generators, it is unlikely that one complies with such a diverse set of requirements. The only deterministic option that fulfilled a satisfactory set of requirements was Remcom's Wireless InSite [77]. However, it is closed-source and perhaps the complexity may be problematic down the line. Thus opting for a QUAsi-DeteRministic RadIo channel GenerAtor (Quadriga) appears to provide the best trade-off between the complexity of deterministic models and speed of stochastic models.

Other alternative could be to side-step channel generation and modify a fully working system-level simulator. However, complete simulators tend to be oriented towards certain types of applications or environments. NYUSIM [78] does not support indoor environments and Vienna Simulator [79] only simulates the downlink. In essence, they lack generality to include our requirements. To the best of our knowledge, these two are the simulators most validated and accepted by the community.

The few that model the channel in compliance with 3GPP, miss one or more requirements from the list above and cannot be used solely for channel modelling because obtaining the channel model requires dissecting the simulator. Some simulators are too low-level for a system-level simulation and require settings at the bit-level, e.g. Matlab's 5G Toolbox [80].Contrary to Matlab's 5G Toolbox, many simulators are poorly documented. Other incompatible simulators we reviewed are, for example, CloudRT [81], WiSE [82] and 5G K-SIM [83].

2.5 Contributions

As any research work, the objective of this project is to evolve the state-of-the-art. Here we present what contributions this work makes to current research in the area.

This thesis focuses on radio access challenges for supporting indoor Social XR applications. We present and test a modelling framework covering all relevant aspects needed for feasibility assessments and performance optimisation.

More concretely, and in an orderly manner, our contributions are:

- Integral modelling framework comprising a XR conference use case, traffic characteristics, network deployment including spectrum assignment and antennas, propagation environment and the key 5G traffic handling and resource management mechanisms. The framework fills modelling gaps in literature, namely with a head movement model and an application traffic model based on video streaming;
- An integration of the developed models in a system-level simulator, which enables extensive sensitivity analysis on antenna deployments, selection of frequency bands, MIMO algorithms, packet scheduling strategies, as well as application use case aspects such as the number of physical and virtual participants, their behaviour in terms of movement, among other;
- Validation of the modelling framework through extensive simulations. We prove the modelling considerations generate realistic and coherent results.

Insights on how configurations impact performance can apply to different use cases and applications. It is so because application-layer models presented in this project can be adapted to apply to similar applications or use-cases, e.g. changing the human behaviour model to fit a virtual reality tennis match. Likewise, the structure of our modelling and simulator can be used to simulate entirely different applications.

Obtaining such insights is a pre-requisite to set up guidelines for local network equipment deployment in order to support services in a cost-efficient manner. Moreover, the proposed framework constitutes a solid base for testing and development of new dynamic resource management methods, potentially AI/ML-based, and radio access parameter tuning strategies.

This work is a step towards autonomously managed network slices that independently configure the network and make trade-off-aware decisions to provide the best possible service with the available resources given the current network state. With ever-growing requirements, autonomous, flexible and optimal use of the network resources is a must to guarantee services, especially in the wireless access.

3

Methodology

Contents

3.1 SXR Conference Application

In this chapter we define all methods and models employed in this work. Namely, a SXR conference, the propagation channel and the radio access network.

This section contains all modelling initiatives and assumptions for a SXR conference meeting. Firstly, the physical placement of BSs, users and cameras is presented. Then we address antenna placement, orientation, geometry and architecture. The placement and orientation are particularly important since it defines how user movement will influences channel quality variability. Subsequently the user behaviour is modelled. Lastly, we present our model for the traffic characteristic based in the packet arrival profile of real-time video streaming.

3.1.1 Physical Setting

Physical and virtual users are uniformly distributed around the table. Naturally, depending on the number of participants/users N_u, the table radius r_t should be changed to resemble reality. Also, the physical and virtual users are as intercalated/interlayed as their numbers, respectively, N_{phy} and N_{vir}, allow. Realistic measures for the case of $N_{phy} = N_{vir} = 4$ are in Figure 3.1.

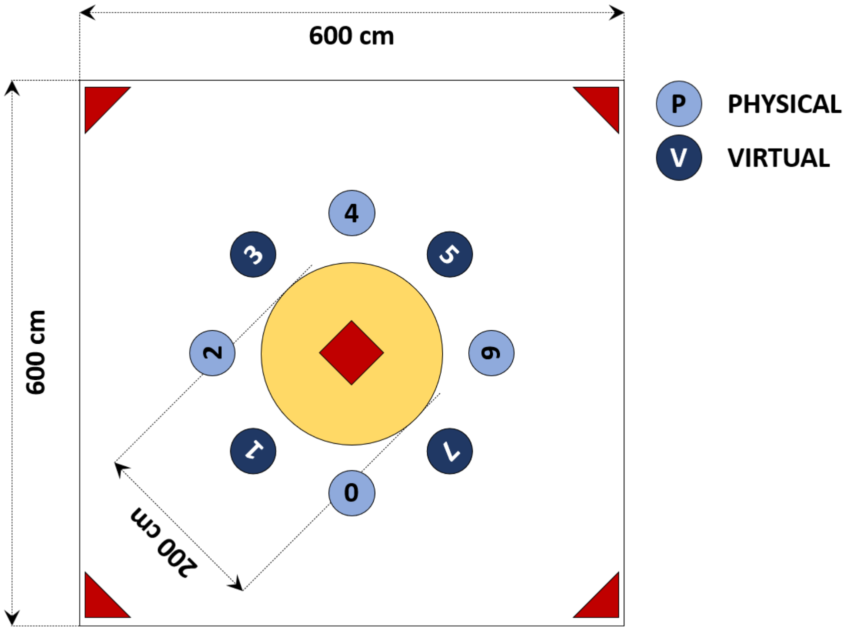
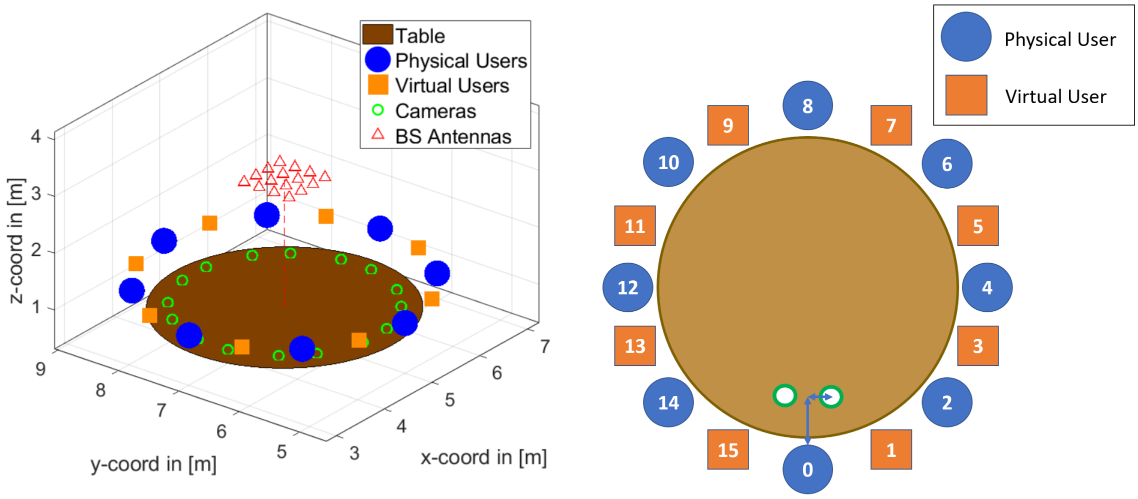


Figure 3.1: Example of room and table size measures.

The cameras are placed in front of the physical users at a distance of d_f, in direction of the centre of the table, and at a distance d_s to each side. In case UE aggregation is preferred, the camera hub is placed in between cameras, with the position solely determined from the user position and d_f. The number of cameras N_{cam} equals two in Figure 3.4a, and the mentioned distances are represented in Figure 3.4b.

The BS panels are placed on the ceiling, at the centre and/or at the corners, e.g. at 3 metres height. The BS is represented in Figure 3.1 by the red square and triangles. Also, a 3D view of the room in presented in Figure 3.4a.



(a) 3D view

(b) Top-view illustration

Figure 3.2: General perspective of metting with users, cameras and base station.

The table is placed at the centre of the room and the precise user position is determined from the centre of the table $C = (C_x, C_y, C_z)$. In practice, the position of each user is given by Equations (3.1) and (3.2) where P_u is the position of user u, r_u is the radius to the user's circumference along which users are placed - users are slightly outside of the table, therefore $r_u > r_t$ - and α_u is the angle in radians from the centre of the table to each user with origin in accordance with user indices.

$$P_u = (C_x, C_y, C_z) + (r_u \cos(\alpha_u), r_u \sin(\alpha_u), 0) \tag{3.1}$$

$$\alpha_u = u\frac{2\pi}{N_u} - \frac{\pi}{2} \ , \ \forall \ u \in \{0, ..., N_u - 1\} \tag{3.2}$$

3.1.2 Antennas

Antenna arrays with half-wavelength distance between elements are used. As a reference, consider the antenna is initially placed in the yOz plane, before being translated to its position and rotated to its orientation. We consider antenna elements from the BS, from user's HMD and from cameras to be single-polarised and part of a cross-polarised element defined by 3GPP.

Having the antenna in the plane of reference, let us call N_V and N_H, respectively, to the number of vertical (parallel to z-axis) and the number of horizontal (parallel to y-axis) cross-polarised antenna elements. We also assume all BS panels have the same number and type of elements. And users and cameras also have the same antennas among themselves.

The antenna size of a BS panel is $N_{V,bs}$ by $N_{H,bs}$. Likewise, a user's antenna size is $N_{V,u}$ by $N_{H,u}$, and a camera's is $N_{V,cam}$ by $N_{H,cam}$. However, the physical proportions of the antenna array should be kept unchanged since one of advantage of higher frequencies is the higher element count. As such, we fix the BS size to e.g. 20×20 cm, and change the element count according to frequency. Figure 3.3 shows this for 3.5 GHz and 26 GHz.

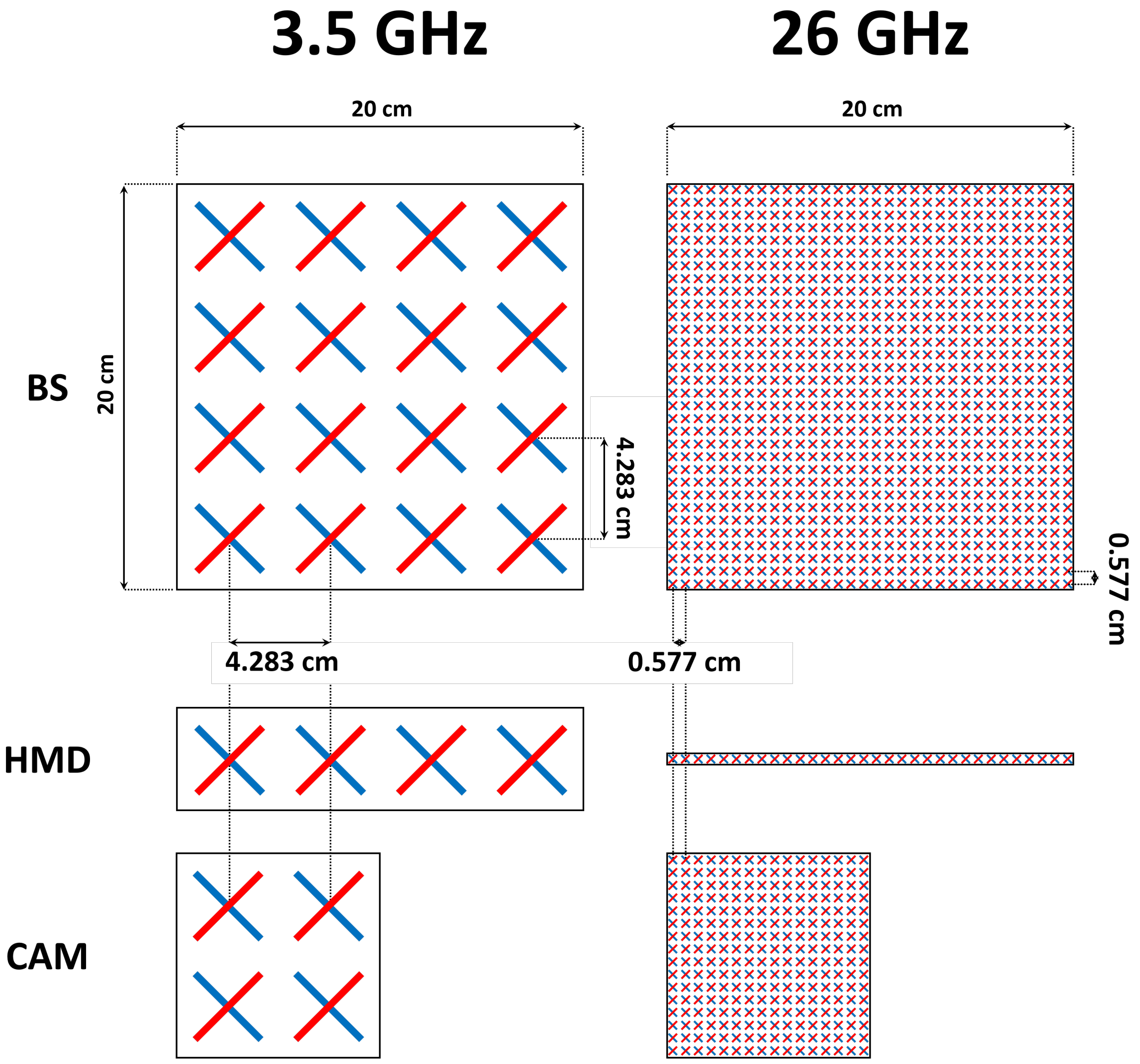


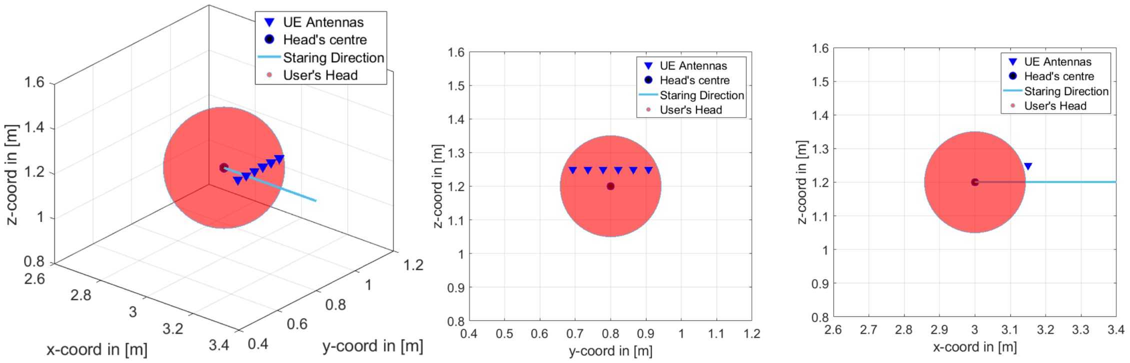
Figure 3.3: Antenna size and geometry modelling example.

We see in Figure 3.3 that while at 3.5 GHz only 16 antenna element fit in the 20cm square BS panel form factor, at 26 GHz the array has 1024 cross-polarised elements.

The respective half-wavelengths at those frequencies are also shown, as well as other realistic possibilities for antennas in the HMD and camera.

With regards to positioning of antennas in the user's HMD, the centre of the antenna array coincides with the instantaneous user position. However, the antenna elements have an offset such that variations in head orientation also cause change in the position of each element. These considerations make the antenna elements move with the user's head as if they were on top of glasses.

In detail, we set an offset distance outwards in direction of user orientation d_o, and an offset distance along the z-axis d_u, both measured from the centre of the user's head in the respective directions. An example with $d_o = 0.15$ metres and $d_u = 0.05$ metres is illustrated in 3.4, simulating antennas placed across the top of a minimalistic pair of AR glasses. Another interesting placement with potential would be on top of the user's head.



(a) Random View **(b)** Front view **(c)** Side View

Figure 3.4: HMD 6-element uniform linear array placement example for 3.5 GHz.

Let us now address orientations. Antenna orientations are vectors that determine the spatial direction of the antenna's normal. Since our default antenna belongs to the yOz, the default orientation is (1,0,0), correspondent to the positive x-axis.

A camera is static, their antenna arrays are simply oriented towards the ceiling, i.e. $O_{cam} = (0, 0, 1)$, and neither their position nor orientation changes. However, this is not the case with BSs and HMDs.

Concerning BSs antennas, Figure 3.5 shows the five positions considered, at a frequency of 600 MHz to facilitate the distinction of individual antenna elements. Each antenna panel is pointed at a the centre of mass of the room C_m - for the case of a 6 by 6 metre room with a ceiling at 3 metres tall $C_m = (3, 3, 1.5)$. In practice, the BS orientation O_{bs} comes from the difference between the BS position P_{bs} relative and the centre of mass of the room, i.e. $O_{bs} = P_{bs} - C_m$.

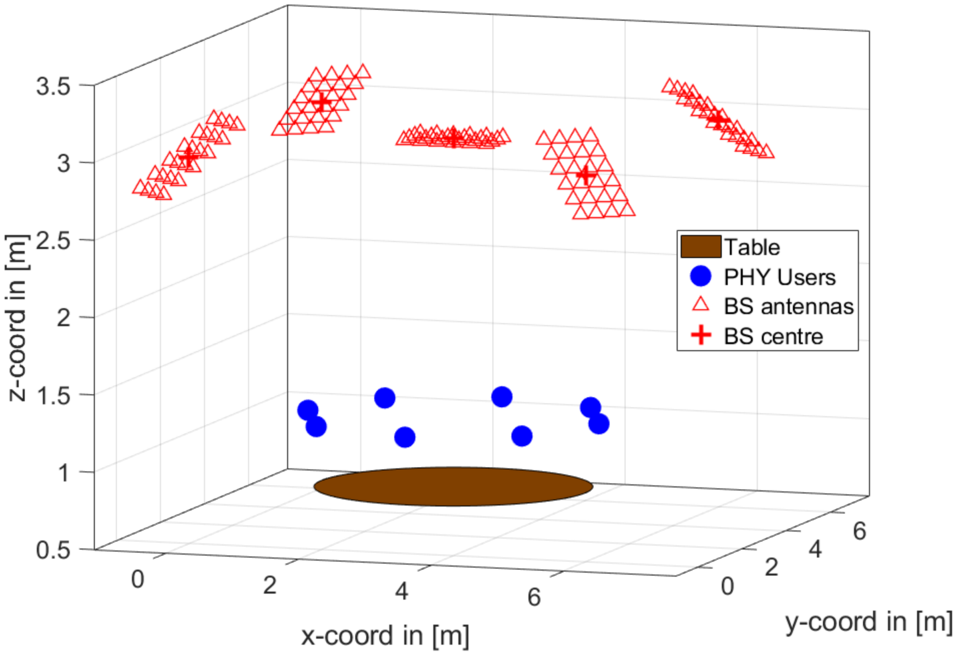


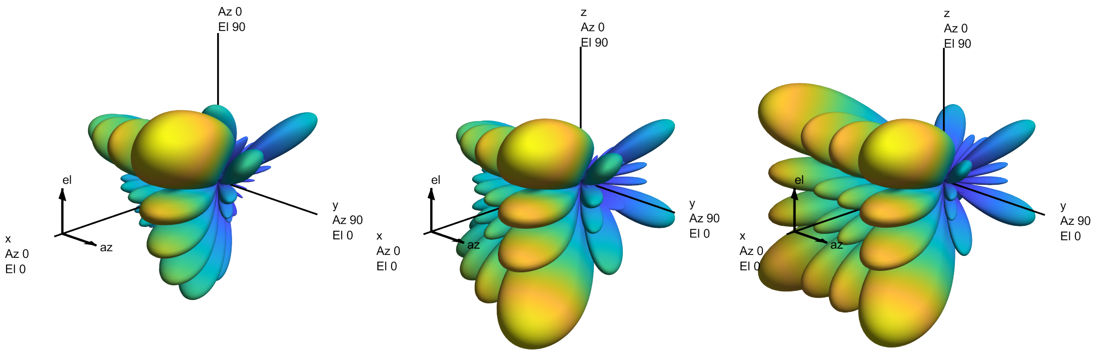
Figure 3.5: Position of multiple panels, each with 6×4 elements, for 600 MHz.

The orientation of the user's antenna O_u is the same as the user's orientation, which is defined in the next subsection, dedicated to user behaviour in terms of movement.

Finally, all antennas have a fully connected (or digital) architecture, i.e. each antenna element possesses its own radio frequency chain. Despite more expensive, this architecture is required in place of the more cost-efficient hybrid architecture due to radiation pattern symmetries.

Take the example of a panel placed at the centre of the ceiling, like represented in Figure 3.4a. From the perspective of that antenna, the users are in symmetric angular directions: if there is a user at relative azimuth and elevation $(\phi, \theta) = (0, 30)$, then there also is one user at $(0, -30)$. Therefore, when beamforming in either of those directions, the other direction should not have a significant lobe, or else there will be interference and the users cannot be co-scheduled.

Figure 3.6 shows the radiation patterns obtained from beam-steering an 8 by 8 rectangular array to 30° azimuth and ° elevation, using subarrays with different sizes. We see that with the digital architecture in 3.6a there are no major lobes in symmetric directions. However, in 3.6b and 3.6c which are hybrid architectures the lobes in symmetric directions are significant. Therefore the need to resort to a digital architecture for BS panels. For simplicity we consider digital architectures for all antennas.



(a) 1x1 subarray **(b)** 2x1 subarray **(c)** 2x2 subarray

Figure 3.6: Radiation pattern of 8 by 8 array steered to $(\phi, \theta) = (30°, 30°)$

3.1.3 User Behaviour

We consider all users seated around a table, thus user behaviour concerns solely head movement. We modelled user head position, which applies a translation to the HMD antennas equal to the change in head position. And we modelled user head rotation, that affects not only the orientation of the antennas but their positions as well, as explained previously.

The instantaneous head position is generated by sampling three normal distributions, for each cartesian component, and adding the outcome as an offset to the average position P_u. Hence, the normal distributions have mean zero and standard deviations $(\sigma_x, \sigma_y, \sigma_z)$, respectively for each cartesian component. There is no covariance between distributions. This way, if all standard deviations are equal to σ, roughly 99% of points will be inside the $\sigma/3$ metre radius sphere, as shown in Figure 3.7.

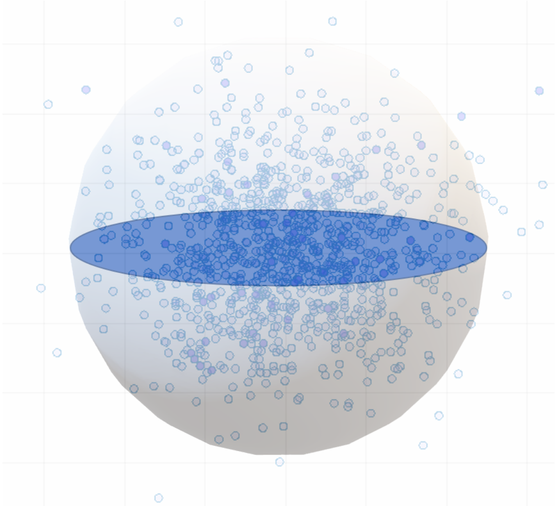
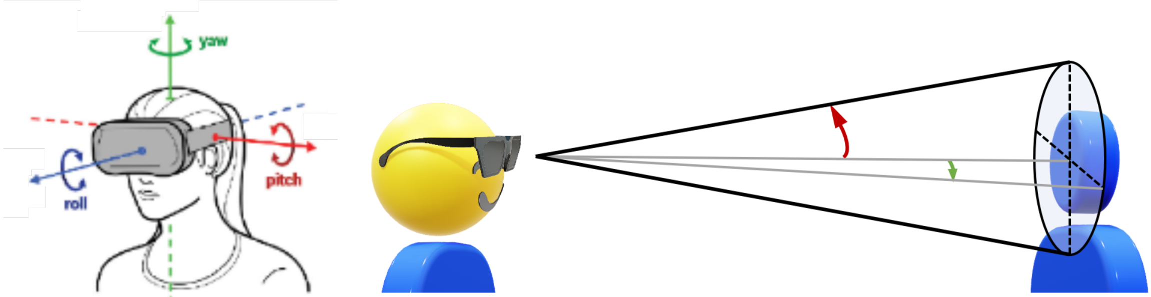


Figure 3.7: Sampling of three zero-mean, independent and identical normal distributions

Empirical observation indicates that height changes less than x and y components, therefore we should have $\sigma_z < \sigma_{x/y}$, rendering the actual volume of possible positions to an ellipsoid, not a sphere.

Change in orientation is quantified by three rotations around the cartesian axis. Happens according to two mechanisms:

- Wobbling while staring - while looking to a certain user, there are deviations from the centre. Limiting the angle of deviation create a cone of focus, as in Figure 3.8b. The figure shows pitch and roll rotations. The orientation of the antenna is constantly inside this cone. The cone is created from random sampling uniform distributions between $\pm\beta$, where β is the uniform distribution limit. The limits for roll, pitch and yaw, respectively, right-hand rotation around the x, the y and the z axis, are $(\beta_x, \beta_y, \beta_z)$.
- Change of focus - at the beginning of the simulation we define a speaker list, with users and the instants they start speaking. When a new user starts speaking, every other user turns its head to this new speaker, and this new speaker looks at the last user speaking, as if answering. This orientation change is more accentuated than head wobbling. The cone of focus is always centred at the user that is currently speaking. By default, each user looks at every other user in equal time intervals of T_{sim}/N_{users}, where T_{sim} is the simulation duration.



(a) Rotation reference

(b) Cone of focus

Figure 3.8: Head rotation model.

The integration of both rotation mechanisms happens naturally. When a new speaker starts speaking, all other users' cone of focus changes to the respective speaker and they rotate from their current orientation in their last cone, to a new orientation belonging to the new cone of focus.

Both head translation and rotation mechanisms are active simultaneously and across time to vary HMD antenna positions and orientations. To facilitate the quantification of movement, the movement index δ is introduced. The speed at which a user moves its head, both translation and head rotation, changes in accordance with the head movement Equations (3.3) and (3.4). The higher the movement index, the faster users move their head.

$$s = \delta \times 0.1 \quad [m/s] \tag{3.3}$$

$$T_{rot} = 1.5 - 0.2 \times \delta \tag{3.4}$$

The speed s in 3.3 tells us how quickly a user hops between positions. The rotation interval T_{rot} in 3.4 is the time the user takes to turn from one orientation to the other. A constant rotation interval means that the head rotation speed changes depending on the angle of rotation, i.e. rotating from one orientation to another takes always the same time, thus smaller changes are slower and vice-versa. We have found this to be more realistic than maintaining always the same speed since it was observed than 180 degree rotations had considerably higher rotation speed than than 30 degree rotations in a normal meeting.

Using Equations (3.3) and (3.4) with integer values of the movement index we obtain values for the head translation speed and head rotation speed, present in Table 3.1.

Table 3.1: Translation speed and rotation intervals for integer movement indices.

Movement Index	Speed [m/s]	Rotation Interval [s]
0	0	-
1	0.1	1.3
2	0.2	1.1
3	0.3	0.9
4	0.4	0.7
5	0.5	0.5
6	0.6	0.3
7	0.7	0.1

A more visually descriptive way of showing how this movement index affects the position is Figure 3.9. Further note that the standard deviation of the received power is positively correlated with the movement index.

Although Figure 3.9 is in 2-dimensions, position changes 3D. Moreover, a movement index of 3 appears to be a realistic value for users, as evidenced by an empirical, non-scientific and highly subjective assessment. Cameras have a movement index of 0 because they are static.

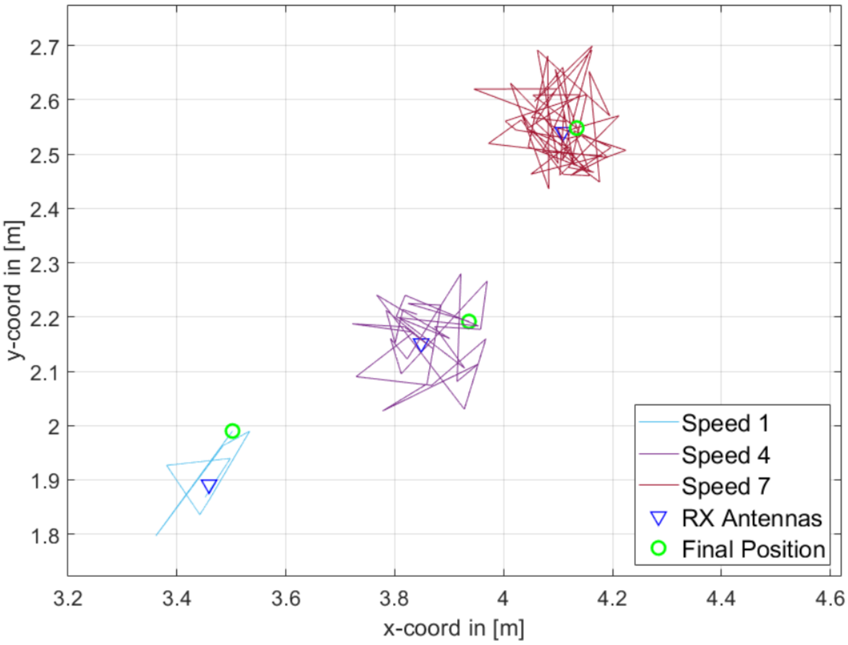


Figure 3.9: Head centre trajectory over 10s for different head translation speeds.

In essence, the translation and rotation mechanisms are similar. Both positions and an orientations can be represented by three coordinates, but one is a cartesian coordinate and the other is a set of rotation angles around the coordinate axis, respectively. The speed tells us how quickly to move from a position to the other, and the rotation interval tells us how quickly to change from an orientation to the next. The higher the speed, the more positions a user's head will visit. The lowest the rotation interval, the more orientations inside the cone the user's look will point. However, the head rotation interval plays an additional role in how fast a 'head turn' happens, when the speaker changes.

3.1.4 Traffic Model

The model that determines the time of arrival of packets resembles video streaming. We chose video streaming to shape our incoming traffic because nowadays there still is plenty of uncertainty regarding what format for 3D user representations performs the best in terms of throughput requirements, capture and encoding speeds, ease of stitching multiple streams together, among other. As such, we made a solid, relatively general model based on well-known traffic characteristics of video-streaming.

The packet generation process is frame-based. In modern video streaming, not all video frames are equal. There are I-frames (Intra-coded frames), that constitute essentially the complete picture.Then there are frames that considerably smaller: P-frames (previous-dependent), and B frames (bi-dependent, previous and forward dependent). P and B frames use less space because they can use information from adjacent frames to decompress, thus this is information that can be derived and does not need to be transmitted. For simplicity the models uses P-frames only, although the process described in this section only requires minor intuitive modifications to include B-frames as well.

Firstly, we compute the size of the I-frame S_I. Then, the size of the P-frame S_P is directly determined by the ratio between I and P frames ($r_{P/I}$), which depends on how many changes occur between frames, as these changes need to be encoded in the P-frame. For a meeting, a ratio of 20% between P and I-frame sizes is expected. Although it can vary across time, here it is considered constant. With the frame sizes, and time interval between frames (equal to the inverse of application the frame rate R_F, e.g. 30 frames per second), we can compute when frames are generated.

The size of an I-frame can be computed based on the average application throughput $\overline{R}$. It can be for UL $\overline{R}_{UL}$ or downlink $\overline{R}_{DL}$, and they are related $\overline{R}_{DL} = r_{DL/UL} \times \overline{R}_{UL}$, where $r_{DL/UL} = N_{phy}$ for AR and $r_{DL/UL} = N_u$ for VR. Computing traffic characteristics based on average throughputs is the safest way of proceeding since it allows minimal assumptions on the application layer parameters.

We need to choose a frame rate R_F, a $r_{P/I}$, and a Group of Pictures (GoP) size S_{GoP}. A GoP is a set of frames containing a single I-frame. Assuming usage of I and P-frames only, it further holds $S_{GoP} - 1$ P-frames. Figure 3.10 illustrates this relation for the common value of $S_{GoP} = 6$.

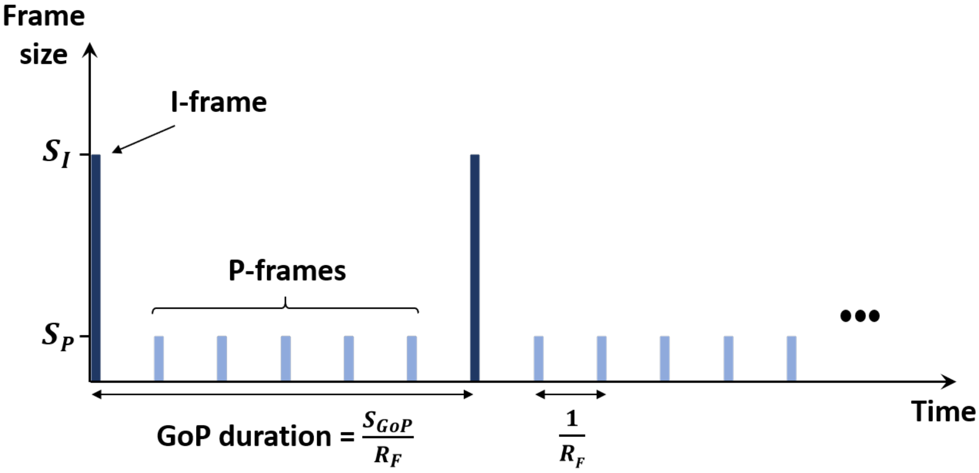


Figure 3.10: I and P-frames in time with a GoP size of 6.

If all frames were I-frames, we could relate S_I with $\overline{R}$ by multiplying the bits in a frame (which would be equal to S_I) with the frames in a second R_F, i.e. $S_I \times R_F$. When using P-frames, we need to compute an average frame size S_F, as in Equation (3.5), before multiplying the frame rate.

$$S_F = \frac{S_I + (S_{GoP} - 1)S_P}{S_{GoP}} \tag{3.5}$$

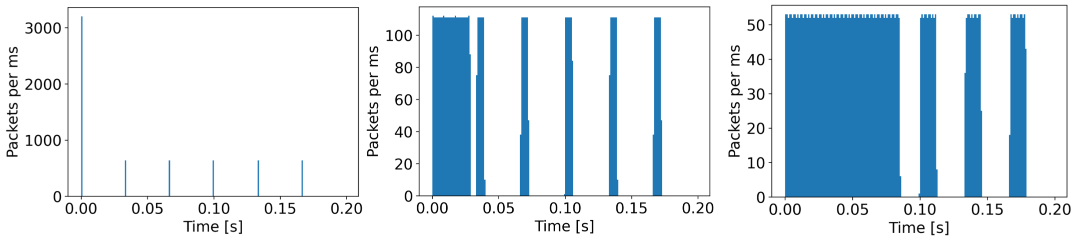
We can write S_P as $S_I \times r_{P/I}$, thus reaching Equation (3.6) which summarises the process of calculating the I-frame size.

$$S_I = \frac{\overline{R}\; S_{GoP}}{R_F\left(1 + (S_{GoP} - 1)r_{P/I}\right)} \tag{3.6}$$

Since this is kind of application has very demanding latency requirements, small GoP sizes are favoured, because an I-frame is sent more often decreasing the likelihood of image freezing. And due to the small GoP sizes, including B frames would not yield any significant effect.

Subsequently, each frame is turned into packets, assuming a constant packet size S_{packet} of 1500 bytes. And the packets are spread out in time, according with burstiness γ and overlap o parameters, such that the traffic characteristic can be adapted to be made realistic for uplink and downlink, respectively, camera and BS buffers.

In the uplink we assume all packets from a certain frame are instantaneously available since the actual frame packet-forming process speed would far surpass the uplink dispatching speed. The downlink packets have come from the MCU passing through many nodes across the internet. Therefore, due to throughput bottlenecks and queue delays, the packets may arrive more spaced. To mimic this effect we use a burstiness parameter. Figure 3.11 shows how the number of packets is spread out in time by changing γ.



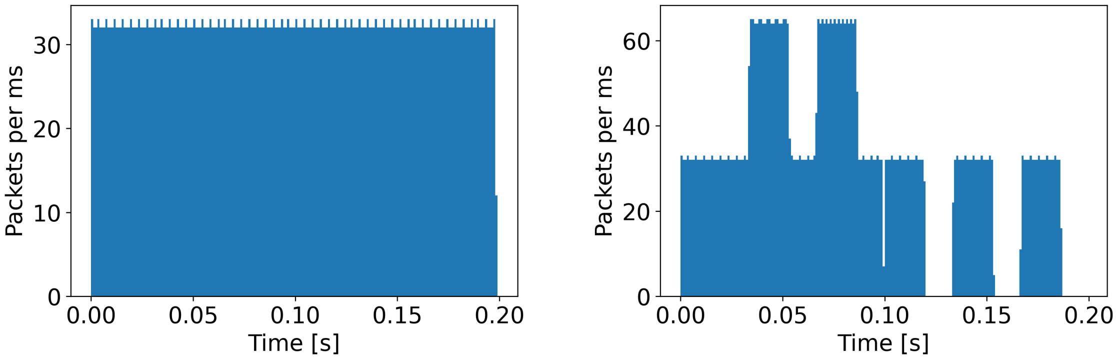
(a) $\gamma = 1$ **(b)** $\gamma = 0.7$ **(c)** $\gamma = 0.3$

Figure 3.11: Impact of burstiness parameter on the packet arrival for the duration of a GoP.

In detail, to obtain the packet arrival rate R_{packet} we convert the average throughput with the burstiness parameter according to Equation (3.7).

$$R_{packet} = \frac{\overline{R}}{1 - \gamma} \tag{3.7}$$

When the burstiness parameter is at its maximum of 1, the packets arrive to the buffers instantaneously, i.e. exactly when a frame is generated. For burstiness parameters smaller than 1, the information of a single frame arrives across time, and the overlap parameter plays a role. It is possible that packets from a given frame get so spread out in time they start to cross over with packets from other frames. Figure 3.12 illustrates the binary value of overlap parameter o. When the burstiness is minimum at 0 it leads to a constant packet arrival rate when the $o = 0$.



(a) $o = 0$ (overlap off)

(b) $o = 1$ (overlap on)

Figure 3.12: Packet overlap for minimum burstiness, with GoP size of 6 at 30FPS.

Finally, we need to coordinate packet arrival for the different users, in the UL and DL. In the DL, since all packets come from the MCU it makes sense that all all users have packets to be received simultaneously. In the UL however, there can be coordination in order to separate I frames as much as possible in order to put the network under less load.

As a final note, for generality purposes, we try to make as few assumptions as possible about the application layer. For an insight on how to map such an average application bit rate to actual application QoE, see Appendix B.

3.2 Propagation Environment

Channel modelling for simulating the propagation environment is a fundamental step when assessing performance over a radio channel. In this section we state the procedure for computation of the channel coefficients.

To generate a realistic propagation environment we use Quadriga as our channel generator. More specifically, the conference takes place in an indoor office environment and Quadriga implements a LoS office described by 3GPP in [51].

The channel traces consist of complex impulse responses for each pair of antennas between transmitter and receiver. Each coefficient corresponds to the electric field amplitude reduction and phase difference between antenna elements, taking into account the path loss, the antenna patterns, the LoS polarisation and the antenna orientations, as well as environment-specific fading. Noise is not considered and needs to be added a posteriori, as done in Equation 2.5.

In Figure 3.13 we show how channel gain varies across time for a user in a meeting, for two different frequencies with an antenna array with the same number of elements. Strictly speaking, Quadriga does not support time-evolution, but it supports space-evolution. As such, we make space evolve consistently, by proving new positions every $250\ \mu s$, which the slot duration for numerology two, for both frequencies. This is equivalent to sampling a channel evolving in time. Also represented are the zones where we deliberately caused a head-turning orientation change, i.e. who is speaking changed, and the user turned its head to the new speaker. Additionally, a LoS human blockage is introduced.

A few aspects are noticeable:

- There is a difference of 15 to 20 dB of average channel quality between 3.5 and 26 GHz, which checks out with the increased free-space attenuation by increasing frequency;
- The channel gain at higher frequency oscillates more. This makes sense due to its smaller wavelength, as introduced in Section 2.2;
- The drop in power due to the blockage is greater in 26 GHz than in 3.5 GHz. This can be explained by mmWaves having an higher percentage of the total received power in the LoS component, therefore if the LoS is blocked with a blocker that takes 20 dB off channel gain [84], the drop is more noticeable in cases where the LoS carries relatively more power, i.e. higher frequencies;
- The rise in channel gain after the blocker passes is smaller than the gain drop at the start of the blocked zone. This is due to variability in percentage of power in the LoS. Although we remove the 20 dB attenuation from the LoS, the percentage of power in the LoS is smaller than previously, therefore the change in channel gain is less noticeable.

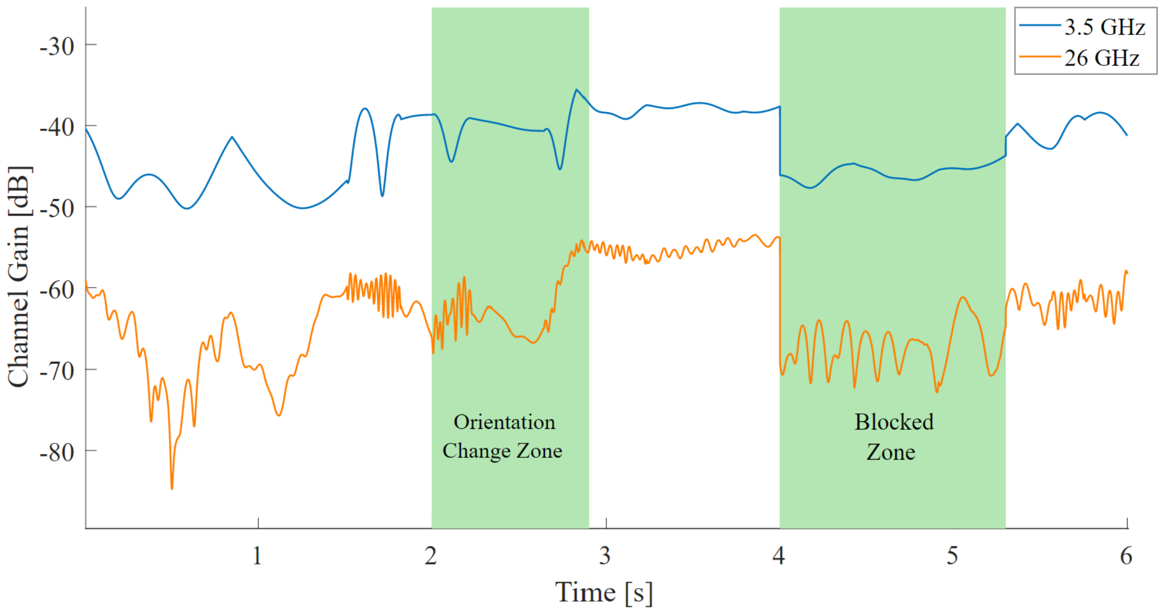


Figure 3.13: Radio channel gain variation across time, for 3.5 and 26 GHz.

Moreover, from correlating channel variability with head rotation we conclude that tilting the head vertically influences more than rotating the head in the horizontal plane. Antenna positions are the cause of such disparity - we used a square array pointing in the direction the user is looking, therefore up or down tilts result in higher variation relative to the position of the BS. This agrees with the plot: from 0 to 1.5 seconds there is a rotation with a significant vertical component and the channel gain oscillates considerably, and for the last second there are mainly horizontal rotations and the channel stays closer to constant.

Furthermore, changes in head orientation have a greater effect on channel gain than changes in head position. It is partly due to spatial consistency, i.e. similar propagation characteristics for similar positions. In essence, consecutive positions are similar enough and correlated that they cause only a small change in the channel, while orientation, even in small quantities, may completely misalign the main lobes of the radiation patterns from beamforming, henceforth leading to more salient changes.

Although the description is simple, the implementation process of such model is complex due to the sheer amount of data required (Terabytes) and computation time (days). We explore this engineering problem in Appendix F, where we look at the channel generation process from the implementation point-of-view and we introduce a parallelisation framework to reduce the required time.

3.3 Radio Access Network

In this section, we detail the functions executed by the network equipment to enable data transmission. The network equipment needs to acquire CSI and manage resources accordingly to cope with the incoming application traffic and fulfil service requirements. Firstly, we go over important considerations and assumptions, namely regarding multi-layer transmissions, concentrating on the DL, among other miscellaneous but relevant matters. Then we list the steps required to simulate a TTI and all processing associated with making the right choices when transmitting and receiving. We summarise these steps with a flowchart and proceed to detail each one.

Firstly, we opt for a GoB-based beamforming approach. With the growing number of antennas at the receivers, full channel knowledge is practically unobtainable, and we need to resort to more overhead-efficient approaches.

Secondly, we address the considerations regarding multi-layer transmission. To reiterate, the difference between single-layer and multi-layer operation is the number of independent streams transmitted per UE. And to transmit independent streams or layers, there must be some orthogonality mechanism that renders such layers independent. The orthogonality domain we are concerned with is orthogonality in space. However, by opting GoB-based beamforming although we save in overhead, we loose considerably in transmission flexibility. Free-format beamforming would allow us to send independent layers in the same direction, only focusing different antennas at the reception. But using a GoB we do not have enough beams to do that, instead we have to resort to completely different propagation paths, with paths beyond the first not being the LoS. This would not yield insignificant improvements.

Another option would be to resort to polarisation orthogonality. Instead of using all antenna elements to perform a transmission, we may use the antennas oriented in a given direction to send one layer and the elements oriented perpendicularly to send another. Note that the same beam in the GoB can be used for the different polarisations when they are to be sent over the same path. However, the moments in time where inter-polarisation interference is small, e.g. less than 20 dB, are rare. In other words, often antennas with a given orientation at the receiver get signal from both polarisations at the transmitter. Thus, it would require considerably more complicated interference estimations algorithms to do multi-layer transmissions polarisation-based. This is why we opt for single-layer transmission using all antennas both in the TX and in the RX. Nevertheless, the vast majority of modelling in this section is agnostic to the number of layers.

Thirdly, although in Section 3.1 we modelled the location of all UEs in the system and application traffic for UL and DL, for conciseness in this section we describe the model for DL transmission procedure and thus we do not consider cameras. In the DL, he number of UEs N_{ue} equal to the number of physical users N_{phy}. Moreover, we consider a single-BS with a one or more antenna panels.

In essence, we need to list all procedures that can happen in a TTI. Some may not happen every TTI and we need to state in what circumstances they do happen. Figure 3.14 shows a flowchart of the main steps required to simulate a (DL) TTI.

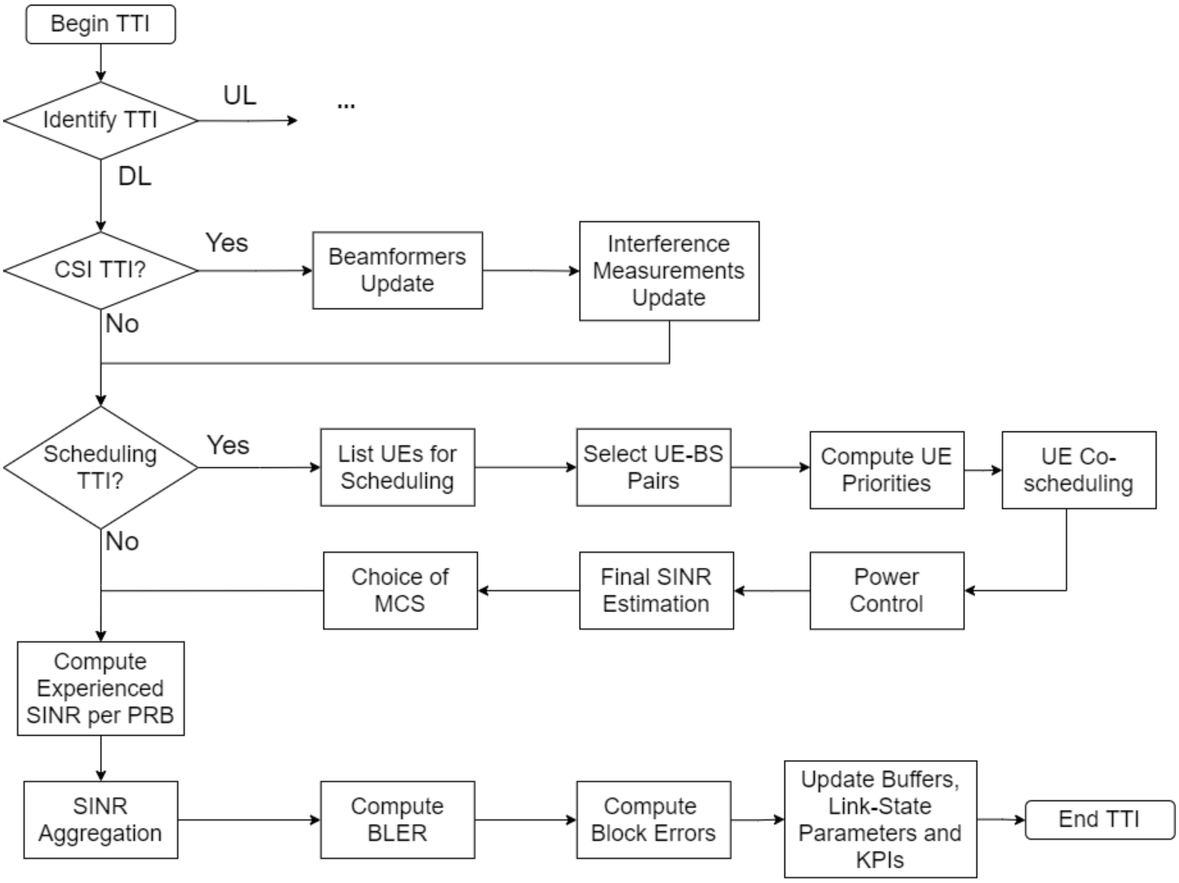


Figure 3.14: Flowchart for of simulation steps for each TTI.

Several verifications are made to decide whether some procedures should take place. The first is to identify the nature of the current TTI - UL and DL TTIs have different steps. The second is checking whether CSI should be updated. Thirdly, it is to verify whether the current user scheduling information for that TTI is to be updated. Only after those verifications and respective procedures, the transmissions scheduled for the present TTI are processed.

We start by assessing the nature of the TTI. It depends on the slot-structure and TDD split.

TDD Split and Slot Format

We recognise two options. The first is to use self-contained slots, at a cost of about $2/14 \approx 14\%$ lower bit rate since 2 out of 14 symbols are used for guard and control, but having the benefit of feedback about block errors in the same TTI, thus allowing triggering retransmissions of the lost information the next TTI. This way the likelihood of packet dropping due to transgressions of time constrains is reduced since latency is reduced, leading to more opportunities to transmit the data on time. The second is simpler and more throughput-efficient, at the cost of latency performance. It consists on using slots that only have DL/UL symbols, respectively, and we ignore the guard time in the transition slot.

Therefore our definition of UL-DL split, or TDD split depends on the option. We define s_{TDD} as the ratio between UL and DL slots. We represent this ratio as $N_{slots}^{DL} : N_{slots}^{UL}$, e.g. 4:1, meaning that for each UL slot there are 4 DL slots. The slot structure is fully defined by the number of slots in a transmission period N_{slots}^{TDD}.

In order to optionally change between both options, we introduce a transport block acknowledgement delay τ_{ACK} (in TTIs) and a slot efficiency η_{slot}. The acknowledgement delay is the number of TTIs before the transmitter receives the acknowledgement, thus $\tau_{ACK} + 1$ is the number of TTIs until the erroneous transport block can be transmitted again. With self-contained slots, $\tau_{ACK} = 0$. Without self-contained slots it depends on the s_{TDD}.

The slot efficiency $\eta_{slot} = 0.86$ in the example of self-contained slots where 14% of symbols are not used for data, and $\eta_{slot} = 0$ in the DL/UL heavy slots. It is applied to the instantaneous throughput R as $R_{modified} = R\eta_{slot}$

As mentioned, we solely present, and posteriorly evaluate, modelling for DL TTIs. Thus after making the distinction between TTIs, the next step is to update the CSI information based on our beamforming strategy. Therefore, let us first state how the GoB is created.

Grid of Beams

To create a GoB we need to know which directions to steer the beam. The beamsteering directions are all possible combinations of values in the azimuthal and elevation angular domains, relative to the antenna boresight (direction perpendicular to the plane the antenna array is inserted). And to create a beam grid in one such domain, one simple way is to use the resolution and the values of the extremes. We define in Equation (3.8) an interpolation function to perform the operation of creating a set of values from a to b, given $b > a$, with intervals of resolution r.

$$F_I(a,b,r) = \{a + i \times r \; \forall \; i \in \mathbb{N}_0 : i \times r \leq b - a\} \tag{3.8}$$

This way, we define in the azimuthal angular domain as $\mathcal{A}_\phi = F_I(a_\phi, b_\phi, r_\phi)$ and the elevation angular domain as $\mathcal{A}_\theta = F_I(a_\theta, b_\theta, r_\theta)$. For instance, if the antenna is positioned in the centre of the room, on the ceiling, pointing downwards, then the most logical approach is a symmetric approach because in that position the coverage of the room would be uniform since we consider our room with equal length and width - room and user behaviour is modelled in Section 3.1. More concretely, the GoB should cover all positions the UEs may potentially be. Thus, given the position and movement of the users in relation to the size of the room described in the example of Section 3.1.1, choosing the lower limits to $a_\phi = a_\theta = -60°$ and the upper limits to $b_\phi = b_\theta = 60°$ covers all possible UE positions.

The resolutions should depend on the array size. To create a pseudo-non-interfering GoB, where the maximum of the main lobe of one beam points at the a minimum of an adjacent beam, the resolution should be roughly half the First Null Beam Width (FNBW). It is 'pseudo-non-interfering' because the FNBW varies with the direction at which the beam is steered, which causes the maximums to not align perfectly with the nulls. This effect is unnoticeable in adjacent beams, and gets more noticeable the more far apart beams are from each other. So, this method is a simplistic yet effective approach to minimise the interference between beams, but it does not eliminate this interference.

Thus, the possible directions are defined as a cartesian product between the azimuthal and elevation domains, shown in Equation (3.9).

$$\mathcal{D} = \mathcal{A}_\phi \times \mathcal{A}_\theta = \{(\phi, \theta) : \; \phi \in \mathcal{A}_\phi, \; \theta \in \mathcal{A}_\theta\} \tag{3.9}$$

Having the directions, we need the precoder that will construct a beam pointing in that direction. In Equation (3.10) we define the M by N beamforming matrix $\boldsymbol{W}_{\phi,\theta}$ that contains the relative amplitudes and phases that are applied to the signal of each antenna element of an M by N planar array, obtaining as a result a beam directed to ϕ degrees on the horizontal plane and θ degrees on the vertical plane. Note that such planes depend on the orientation of the array and the angles ϕ and θ are null in the interception of both planes, corresponding to the direction orthogonal to the array plane. Appendix A holds a complete derivation.

$$\boldsymbol{W}_{\phi,\theta} = \begin{bmatrix} 1 & u_2 & \dots & u_2^{(N-1)} \\ u_1 & u_1 u_1 & \dots & u_1 u_2^{(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ u_1^{(M-1)} & u_1^{(M-1)} u_2 & \dots & u_1^{(M-1)} u_2^{(N-1)} \end{bmatrix}, \text{ with } \begin{cases} u_1 = e^{-j\pi \sin(\phi)\sin(\theta)} \\ u_2 = e^{-j\pi \cos(\phi)\sin(\theta)} \end{cases} \tag{3.10}$$

Subsequently, to obtain every precoder in the GoB we need to build a precoding matrix for each direction in $\mathcal{D}$. Let us define in Equation (3.11) the set $\mathcal{W}$ containing all precoders $\boldsymbol{W}_{\phi,\theta}$ in the GoB, formed for an M by N Uniform Rectangular Array (URA).

$$\mathcal{W}^{\text{GoB}} = \{\boldsymbol{W}_{\phi,\theta} : (\phi, \theta) \in \mathcal{D}\} \tag{3.11}$$

As a last step, we vectorise the matrix $\boldsymbol{W}_{\phi,\theta}$ into a vector $\boldsymbol{w}_{\phi,\theta}$, to facilitate its usage. This process is exactly the same as stacking the columns of $\boldsymbol{W}_{\phi,\theta}$.

Figure 3.15 illustrates the result of a cut at zero degrees elevation on beams of two grids. The two grids are built for square antenna arrays, with 16 and 1024 elements, respectively, left and right sides of the figure, hence the noticeably different directivity. Using the 3GPP-defined elements in [51], the maximum directivities are 20 dBi and 38 dBi, respectively, for the 16-element array and for the 1024-element array. Furthermore, since the resolutions were purposely set to match half of the FNBW, the grid on the left spans 120 ° of angular domain, from -60 ° to 60 °, with steps of 30 °, while the grid on the right does so with a resolution of 4 °. In total, this equates to 25 distinct beams of the small array and 961 beams in the larger array.

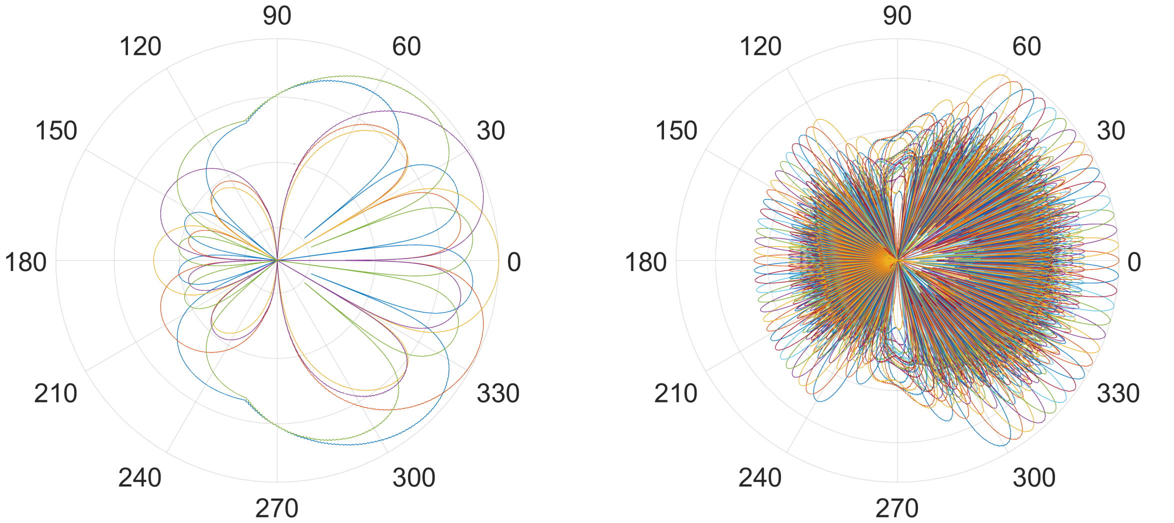


Figure 3.15: GoB azimuth cuts for 4 by 4 (left) and a 32 by 32 (right) antenna element array.

3.3.1 Channel State Information

CSI updates happen every N_{slots}^{CSI} slots or TTIs. Not all TTIs have CSI updates because the channel does not change enough to be worth updating that frequently, and due to prohibitive overheads since reference signals are sent in place of data.

The overheads associated with different CSI feedback schemes is not modelled. The overhead would depend on the type and quality of said measurements, thus we simply define a CSI-slot efficiency η_{CSI} meant to reduce the bit rate of CSI slots.

CSI is required for operation of two important mechanisms. First, to direct and receive signals optimally, in accordance with the paths where attenuation is lower. As such, it is used to update matching beamformers at the transmitter and receiver, or beam pairs. Second, to assess received power and interference, which are crucial to estimate channel quality, which is then used to, e.g. determine which MCS to use.

Let us address the beam pair establishment first. Our formulation holds beam correspondence [61], this means the beam computed for the transmitting are used for receiving as well. Therefore we refer to weights vectors as beamformers, instead of the direction-specific nomenclatures like precoder or combiner.

Beam pairs Update

To update the best beam pairs between UEs and BS panels, the BS should transmit N_{CSI} CSI-RSs precoded in GoB beams and the UE reports how well it received each RS. However, this would require a mechanism for the BS to identify, based on previous channel measurements, which beams are more likely to best serve the UE. As such, instead we check all beams in the GoB to assess which best suit the channel. Furthermore, we keep received power information about N_{CSI} of them, which is useful for future SINR estimations.

The best beam pairs are chosen to maximise the channel gain achieved from performing a transmission with a given GoB beam, with a best effort reception using MRC. Therefore, for a link between UE u and BS panel b, the beamformer on the BS side $\boldsymbol{w}_{bu}^{BS}$ is always a $N_{ant}^{BS} \times 1$ beam-steering vector from the GoB, i.e. $\boldsymbol{w}_{bu}^{BS} \in \mathcal{W}_b^{GoB}$. The UE-side beamformer $\boldsymbol{w}_{bu}^{UE}$ is always the MR beamformer that fits the BS beamformer used over the $N_{ant}^{UE} \times N_{ant}^{BS}$ channel $\boldsymbol{H}_{bu}$. As such, the received signal in each of the N_{ant}^{UE} UE antennas is $\boldsymbol{H}_{bu} \cdot \boldsymbol{w}_{bu}^{BS}$. Here, N_{ant} refers to the number of single-polarised antenna elements. The computation of the UE-side beamformer is given in Equation 3.12, from using Equation 2.7.

$$\boldsymbol{w}_{bu}^{UE} = \frac{\left(\boldsymbol{H}_{bu} \cdot \boldsymbol{w}_{bu}^{BS}\right)^{H}}{\left|\boldsymbol{H}_{bu} \cdot \boldsymbol{w}_{bu}^{BS}\right|} \tag{3.12}$$

When $\boldsymbol{w}^{UE}$ is a MR beamformer, the channel gain under transmit and receive beamforming is a real number. Therefore, to choose the $\boldsymbol{w}$ that achieves the highest gain, we simply have to choose $\boldsymbol{w}$ that results in highest norm of its internal product with the channel. This shortcut is represented in Equation (3.13). In essence, this means that because we are computing the UE-side beamformer already taking into account the transmit-side beamformer, to maximise the norm of the received signal it is sufficient to choose the appropriately the transmit-side beamformer.

$$\boldsymbol{w}_{bu}^{BS} = \underset{\boldsymbol{w} \,\in\, \mathcal{W}_{b}^{GoB}}{\operatorname{argmax}} \left|\boldsymbol{w}_{bu}^{UE} \cdot \boldsymbol{H}_{bu} \cdot \boldsymbol{w}\right| = \underset{\boldsymbol{w} \,\in\, \mathcal{W}_{b}^{GoB}}{\operatorname{argmax}} \left|\boldsymbol{H}_{bu} \cdot \boldsymbol{w}\right| \tag{3.13}$$

When $N_{CSI} > 1$, instead of the best beamformer, we save the N_{CSI} best GoB beamformers. For sake of practicality, let us assume $N_{CSI} = 1$ for now on. Furthermore, beam pairs computed in this way profit from beam-reciprocity, i.e. the beams used for receiving can be used for transmitting as well. And doing this way, the received power is already present from Equation (3.14), thus we only need to update the interference now.

$$P_{r,bu}^{UE} = P_{t,bu}^{BS} \left|\boldsymbol{w}_{bu}^{UE} \cdot \boldsymbol{H}_{bu} \cdot \boldsymbol{w}_{bu}^{BS}\right|^{2} \tag{3.14}$$

Interference Measurements Update

To measure interference, the BS should schedule an empty UE-specific RS for interference measurements. It should result in measuring the power received by the interfering sources. The main drawback is the outdatedness of the measurement. It takes around 4 TTIs until the information is available since it needs to be sent, received, processed and fed back. Therefore, when the interference measurement is available, it refers to τ_{TTI} TTIs back, e.g. 4 TTIs ago.

A major disadvantage of estimating the interference in this manner comes from the fact that the experienced interference is extremely dependent on current scheduling. If the scheduled UEs or beamformers in use change, then it is expected that a major change in the experienced interference takes place, thus possibly rendering the measurement completely invalid. We foresee precise interference estimation algorithms, perhaps driven by learning mechanisms, to be a future direction of work. We discuss this matter further in Section 5.

3.3.2 User Scheduling

Analogous to the CSI update procedure, the scheduling information is only updated every N_{slots}^{SCH} TTIs. In a first stage, we renovate the scheduling information on which UEs are considered for scheduling and which BS panels are used for each UE. In essence, only UEs with non-empty buffers are examined to potentially be part of the scheduled list; and each UE is served by a single BS panel with the best beam pair to that UE. This constitutes the simplest panel selection scheme.

The scheduling process then continues to estimate SINRs and achievable throughputs for each UE, to compute UE priorities, to make user co-scheduling decisions, assign powers for each transmission and derive MCS to be used. The SINR estimation step is presented first.

SINR Estimation

The received powers for the best N_{CSI} beams have been reported in the CSI acquisition step, as well as the interference levels computed from experienced interference from τ_{CSI} TTIs ago. Also, the channel gain can be derived directly knowing the transmit power that was used. Thus, we assume an equal distribution of the maximum transmit power at the BS $P_{t,max}^{BS}$ over the number of scheduled UEs with non-empty buffers. And the only missing piece in the SINR expression is the noise.

We use wideband scheduling, i.e. allocating all available spectrum to every transmission, relying on spatial separation to prevent excessive interference. Therefore, assuming B to be the system bandwidth, using thermal noise we get a noise power P_N given by Equation (3.15), with the Boltzmann constant $k_B = 1.380649 \times 10^{-23}$ J/K, the noise temperature T and an upscaling with the receivers' noise figure NF_r. All hardware imperfections are abstracted by considering noise figures in the BS and in the UEs, respectively, NF_{BS} and NF_{UE}, in dB.

$$P_N = k_B T B \times 10^{\frac{NF_r}{10}} \tag{3.15}$$

To summarise, the expression used for SINR estimation uses the received power P_s and total interference I information from τ_{CSI} TTIs ago. Such information is used as shown in Equation 3.16 to compute an estimate of the SINR $SI\hat{N}R_{eff}$.

$$SI\hat{N}R_{eff} = \frac{\hat{P}_s}{\hat{I} + P_N} \tag{3.16}$$

Instantaneous Throughput

To compute the instantaneous throughput, we need to quantify the value of serving each user. Then we can weigh transmission options regarding fairness, maximum aggregated throughput, or likelihood of fulfilling latency constraints. It is also needed to calculate the estimated and realised bit rates.

The SINR is used to choose which CQI should be reported from the BLER curves represented in Figure 3.16, with equations in Appendix C. The point at which each curve intercepts the BLER probability of 10% is marked. The corresponding MCS choice consists on selecting the highest MCS that achieves a lower percentage of block errors than the Block Error Rate (BLER) target $BLER_0$.

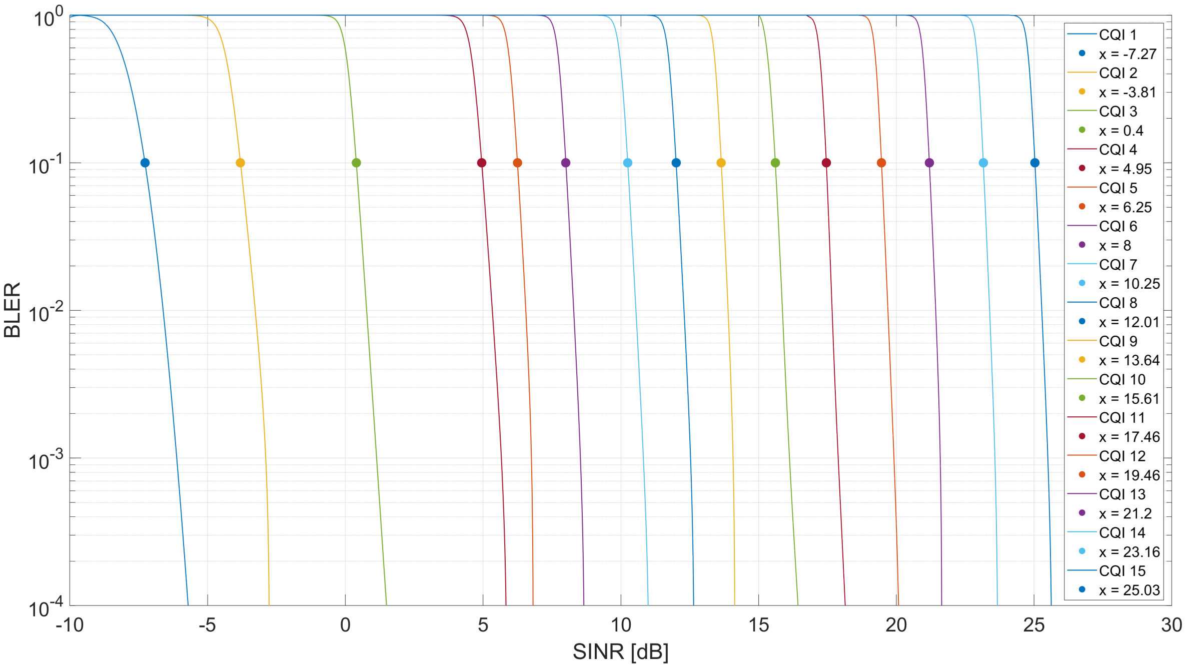


Figure 3.16: BLER curves for all MCSs. Simulated with Vienna Link-Level Simulator [10].

The selected MCS is then adjusted with the OLLA parameter as described ahead. The resultant MCS tells us the number of bits in a symbol N_{bits}^{symb}, which can be computed from the modulation order M, through $\log_2(M)$. To compute the bits per PRB we assume initially that all REs in a PBR are used for data, i.e. $N_{symb}^{PRB} = 168$, we multiply by N_{bits}^{symb} and take into account the code rate R_c. In essence, Equation (3.17) computes the bit rate by dividing the number of bits transmitted in a PRB by the duration of that PRB, which corresponds to a slot duration $T_{slot,\mu}$, that depends on the numerology μ.

$$R_b = \frac{N_{bits}^{symb} \times N_{symb}^{PRB} \times R_c}{T_{slot,\mu}} \tag{3.17}$$

To balance the excessively optimistic assumptions, like assuming all symbols are used for data, we adjust to the bit rate, namely due to signalling overheads and self-contained slots, respectively, by multiplying the efficiencies η_{OH} and η_{slot}. Equation (3.18) has the final bit rate efficiency η. An estimation for the instantaneous throughput per TTI R is $R = R_b \times \eta$.

$$\eta = \eta_{OH} \times \eta_{slot} \tag{3.18}$$

Compute UE Priorities with Scheduler

A scheduler task is to compute UE priorities p according to a trade-off of resource sharing fairness, achieving the maximum instantaneous aggregated throughput, or attain the lower average latencies, to name a few. These priorities allow us to select UEs by order of importance according to the weighted trade-off relation we choose.

The most common and widely used scheduler is the Proportional Fair (PF), presented in Equation (3.19). PF takes the ratio between the estimated instantaneously attainable throughput $\hat{R}$ and average attained throughput $\overline{R}$ to balance immediate reward and fairness across users, for each TTI t. The average $\overline{R}$ is computed using exponential smoothing with a parameter t_w, according to Equation (3.20).

$$p(t) = \frac{\hat{R}(t)}{\overline{R(t)}} \tag{3.19}$$

$$\overline{R}(t) = \left(1 - \frac{1}{t_w}\right)\overline{R}(t-1) + \frac{1}{t_w}R(t-1) \tag{3.20}$$

As seen, PF does not consider latencies. Yet, for our case where each user has the same amount of data to receive (and each camera the same amount of data to transmit), the PF also levels latencies by weighting fairness, not leaving any user waiting for long. However, it may not perform as well as latency-aware alternatives.

Two latency-aware alternatives are Exponential/Proportional Fair (EXP/PF) [85] and Maximum-Largest Weighted Delay First (M-LWDF) [86]. The latter is almost as simple as the PF, only weighting the Head Of Line (HOL) latency as well. EXP/PF is more complex and considers a maximum delay and increases priorities exponentially as latencies approach the limit. Both use the PF ratio described in Equation (3.19). M-LWDF outperforms EXP/PF in practically every scenario, besides when the load is very high [86] [87]. Therefore, both M-LWDF and EXP/PF seem worthwhile alternatives, but we choose the PF for this work.

Co-schedule users

This step lists the users to be scheduled together until the next update to the schedule. The co-scheduling rule for a single-BS-panel operation is to add one UE layer at a time to the list, by order of UE priority (computed in the previous step), if the best beams used for those layers are compatible with the previously added UE layers. And we define as compatible beams when the BS-side beam, belonging to the GoB is at least κ beams apart, with $\kappa \in \mathbb{N}_0$. If κ is 0, then all layers are accepted. If $\kappa = 1$, then the beams must be different - adjacent beams have a distance of 1, so are still used together. Beams located diagonally adjacent of the GoB are considered to have a distance of 2, hence they may be co-scheduled when $\kappa \leq 2$. Figure 3.17 illustrates the beams that cannot be co-scheduled with certain values of κ, representing in filled blue circles as incompatible beams with respect to the orange one, and empty circles as compatible beams with the central orange beam. More generally, the beam distance is defined by the sum of absolute differences of the beam indices in the grid. Mathematically, the beamformers $\boldsymbol{w}_{i,j}$ and $\boldsymbol{w}'_{i',j'}$, having (i, j) and (i', j') as the GoB indices, respectively, are compatible if $|i - i'| + |j - j'| \geq \kappa$.

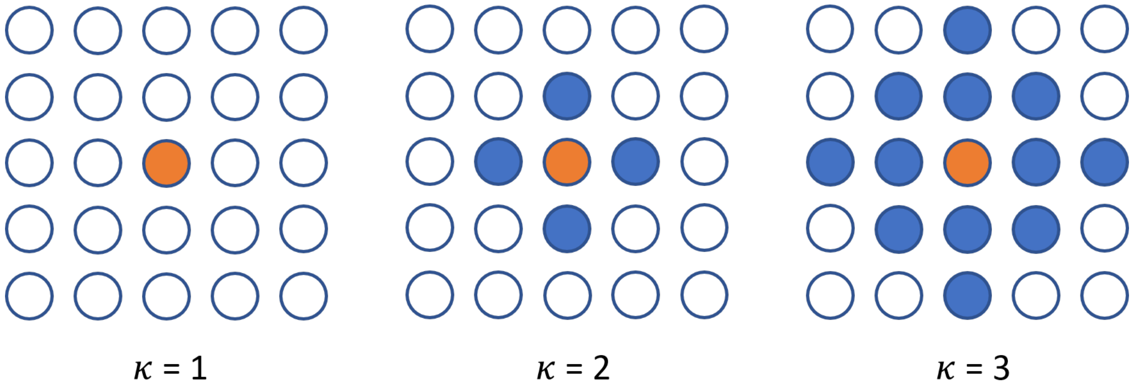


Figure 3.17: Beam co-scheduling incompatibility distance.

In this step there is space for more elaborate algorithms that attempt to choose different combinations of the best N_{CSI} beams of each user, in an attempt to maximise the metrics we care about. Of course, if scheduling one more user considerably reduces the quality of the channel to many others, it is likely not worth doing.

Power Control

Depending on the beamforming strategy, it may be necessary to scale down all precoders due to excessive power per antenna constraints. This, however, does not apply to our case because beam-steering beamformers always have uniform amplitude. Power control in the downlink is as simple as distributing the maximum total transmit power equally amongst the scheduled UEs and assigned PRBs.

MCS selection

By now the transmit power to each user and the beam pairs are fixed. Therefore, we obtain an estimation of the SINR for each user, as done previously but with the new information on powers and beam pairs. And we obtain a MCS to be used for each transmission, according to the same process used in the throughput estimation.

The choice of which MCS to use for transmission has major impact in performance. Since choosing one MCS above what the channel quality allows can lead to excessive errors, while one lower than ideal will wastes resources. Factors such as the outdatedness of measurements complicate the process of choosing the correct MCS. We present a mechanism that adapts the MCS choice according with block errors. This procedure takes place every time a MCS is used, including in instantaneous throughput calculations.

It is a Outer Loop Link Adaptation (OLLA) mechanism [88] and it is UE-specific. When a MCS is estimated, it is subsequently adjusted with the OLLA parameter. The OLLA parameter Δ_{OLLA} is initialised at zero and is updated in every TTI the given UE is scheduled. When a TB is successfully transmitted, the OLLA parameter is updated with a step-size of γ_{OLLA} according to Equation (3.21) such that the BLER long-term average converged to the target $BLER_0$. If the block is erroneous, Equation (3.22) is used instead.

$$\Delta_{OLLA} = \Delta_{OLLA} + BLER_0 \times \gamma_{OLLA} \tag{3.21}$$

$$\Delta_{OLLA} = \Delta_{OLLA} - (1 - BLER_0) \times \gamma_{OLLA} \tag{3.22}$$

Observe the subtlety of the asymmetry in update. The term that multiplies the step size γ_{OLLA} is much bigger in Equation (3.22) than in (3.21), since $BLER_0$ is usually 0.1 or smaller, depending on the QoS reliability requirements. It is a defensive approach, to take bigger steps towards more conservative MCSs when there are errors because it is always better to have some throughput than none. Contrarily, the progression to increasing the MCS is slower.

The OLLA parameter adjusts the MCS choice by flipping an appropriately biased coin and adding either $\lfloor \Delta_{OLLA} \rfloor$ or $\lceil \Delta_{OLLA} \rceil$ to the MCS index estimated in the previous step. An appropriately biased coin in this situation is a coin that selects to round down the OLLA parameter with a probability of $\lceil \Delta_{OLLA} \rceil - \Delta_{OLLA}$. This makes sense because Δ_{OLLA} is decreased when a block has errors, thus making more likely that the MCS is reduced when the link has worse quality than expected. When the block does not have errors, it makes it more likely to increase the MCS estimate, such that a good link condition can be taken advantage of to increase the bit rate. Note that this formulation still works as supposed for negative values, i.e. the OLLA mechanism works for increasing and decreasing the MCS.

3.3.3 Transmission

Here we obtain the outcomes of the realised transmissions. Firstly, we calculate the number of TBs in which the data to be transmitted in a given TTI is segmented. Secondly, the SINRs each UE experiences in each PRB are computed and then we present how these SINRs can be aggregated in something more easily useable to conclude on overall channel quality, an effective SINR. Finally, effective SINRs are probabilistically used to determine the success or failure of the transmitted transport blocks according with the MCS used for transmission and then the link quality adaptation mechanism is updated appropriately, as well as buffers and PF ratios, to be used in upcoming transmissions to assure a balanced operation of the system in line with the result of the transmission in the present TTI. As usual, the steps follow.

Transport Block Size Calculation

To obtain the Transport Block Size (TBS), essentially two ways have been modelled. The first is to consider the same number of TBs on every transmission, N_{TB}. Therefore the numbers of bits to be transmitted $N_{bits,bul}$ is divided equally over TBs and the size of each TB is the same, as shown in Equation (3.23).

$$S_{TB} = \lceil N_{bits,bul}/N_{TB} \rceil \tag{3.23}$$

The second is to consider a maximum TBS $S_{TB,max}$, obtain N_{TB} from Equation (3.24) and then use Equation (3.23).

$$N_{TB} = \lceil N_{bits,bul}/S_{TB,max} \rceil \tag{3.24}$$

This such manner, N_{TB} TBs are sent and the experienced bit rates depend on how many of them are delivered with no errors. If there are no errors, the bit rate computed in Equation (3.17) is achieved, otherwise only a fraction of that bit rate is achieved, corresponding to the successfully transmitted TBs over total TBs. One of the modelled methods is chosen by fixing either N_{TB} or S_{TB}, respectively, for the first and second methods.

Another alternative way is to follow an extensive list of steps described in [12], making the Transport Block Size depend on the number of layers $\#\mathcal{L}_{bu}$ carrying the same

QoS flow, modulation order M, code rate R_c, number of allocated PRBs $N_{PRB,bul}$ and transmission duration, which we assume to be always T_{slot}.

Compute Realised SINR: A Multi-layer SINR Framework with Beamforming

Although the rest of this chapter assumes simplifications for downlink single-layer transmission, for future purposes we derive a general multi-layer framework that works for uplink as well.

To accurately compute the SINR experienced during a transmission, we need to know the power received from each transmitter, for any scheduled UE, taking into account the different channel responses in each PRB of the assigned bandwidth.

Let l be the layer that links a set of antennas in BS b to a set of antennas in a UE u, with $P_{t,l}$ the total transmit power and $P_{r,l}$ is the received power in that layer, after combining the contributions of each receive antenna. Then, let $P_{r,ll'}$ be the power received by layer l receiver using combiner $\boldsymbol{w}_{r,l}$, transmitted by layer l' transmitter using precoder $\boldsymbol{w}_{t,l'}$, with $\boldsymbol{H}_{ll'}$ the channel matrix that connects the receiver and the transmitter. Equation (3.25) shows how these quantities relate.

$$P_{r,ll'} = P_{t,l'} \left|\boldsymbol{w}_{r,l} \cdot \boldsymbol{H}_{ll'} \cdot \boldsymbol{w}_{t,l'}\right|^2 \tag{3.25}$$

The powers are scalars, $\boldsymbol{w}_{r,l}$ is a $1 \times N_r$ vector, $\boldsymbol{w}_{t,l'}$ is $N_t \times 1$ vector and $\boldsymbol{H}_{ll'}$ is a $N_r \times N_t$ matrix, where N_t and N_r are the number of antenna elements at the transmitter and receiver antenna arrays, respectively.

Knowing how to calculate this quantity we can compute the powers of all parts of the SINR expression on a PRB basis: the signal P_s, the intra-cell interference P_{IaCI}, the inter-cell interference P_{IeCI}, the inter-layer interference P_{ILI} and the noise P_N. Here, the term cell refers to a panel. Using all these powers in Equation (3.26) we obtain the SINR of a specific layer l in a given PRB. Subsequently we present equations for each quantity in the SINR expression, along with the rationale behind them.

$$SINR = \frac{P_s}{P_{ILI} + P_{IaCI} + P_{IeCI} + P_N} \tag{3.26}$$

Moreover, and to reiterate, all quantities mentioned in this section are time (TTI) and frequency (PRB) specific. These SINRs need to be posteriorly aggregated in an effective SINR for each transmission in the given TTI. We choose to omit the i index to simplify notation, as we did with the TTI since this chapter is TTI-specific.

The received signal power P_s is presented in Equation (3.27).

$$P_s = P_{r,ll} \tag{3.27}$$

In case of multi-layer transmission, other layers scheduled to/from the same UE may interfere among themselves. The power of inter-layer interference P_{ILI} takes into account this interference by summing the interferences caused in layer l by every other layer l' scheduled between BS b and UE u. This rational is condensed in Equation (3.28), where $\mathcal{L}_{bu}$ is the set of layers scheduled between BS b and UE u.

$$P_{ILI} = \sum_{\substack{l' \in \mathcal{L}_{bu} \\ l' \neq l}} P_{r,ll'} \tag{3.28}$$

Interference power contributions from the same cell/BS come from every transmission that takes place to other UEs in the same cell/served by the same BS. We take into account all those transmission in Equation (3.29), where $\mathcal{U}_b$ is the set of users served by BS b.

$$P_{IaLI} = \sum_{\substack{u' \in \mathcal{U}_b \\ u' \neq u}} \sum_{l' \in \mathcal{L}_{bu'}} P_{r,ll'} \tag{3.29}$$

Interference contributions from outside the cell come from all non-serving BSs, all UEs and in all layers. Equation (3.30) represents this relation, where $\mathcal{B}$ is the set of all BS (or BS panels) in the system.

$$P_{IeCI} = \sum_{\substack{b' \in \mathcal{B} \\ b' \neq b}} \sum_{u' \in \mathcal{U}_{b'}} \sum_{l' \in \mathcal{L}_{b'u'}} P_{r,ll'} \tag{3.30}$$

The noise power P_N is computed according to Equation 3.15, using the bandwidth of a single PRB, which depends on the numerology as evidenced in Table 2.2.

This framework is also applicable when several BS are jointly serving one user, or when one user is transmitting to several BS simultaneously, i.e. Distributed-MIMO (D-MIMO). This is true because we simply account for power contributions, abstracting from the content of the spatial streams.

Aggregate SINRs: Mutual Information Effective SINR Mapping

MI-ESM is an SINR aggregation technique that allows us to attribute one SINR to a transmission where the quality of the channel varies across the transmission band, namely across PRBs. We choose this SINR mapping strategy because [89–93] show that it unquestionably achieves very good results without the need of calibration for different MCSs. Equation (3.31) sums how it works.

$$SINR_{eff} = I_k^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} I_k\left(SINR_i\right)\right) \tag{3.31}$$

Above, I_k is the mutual information function that for a given SINR and MCS (with k bits per symbol) gives the bits of information that are conceivably extracted for a transmission with that SINR. For low SINRs, the mutual information is practically zero. As the SINR grows, the quantity of information bits extracted approaches k. Appendix D goes into further detail on the mutual information function works.

Therefore, Equation (3.31) obtains the mutual information achievable in each PRB, averages it and computes the SINR that would achieve that average information. Thus, the effective SINR is determined as the SINR that would yield this average mutual information if it were applied on all PRBs.

Compute Block Errors

Subsequently, with the effective SINR $SINR_{eff}$ and the MCS used for the transmission, we get the resultant $BLER$ from the correspondent MCS curve in Figure 3.16. Then we flip a BLER-biased coin to determine whether each block was received well.

Update Link Adaptation, Buffers and Performance Indicators

Firstly, the link adaptation mechanism is updated based on the block errors in accordance with Section 3.3.2.

Then, the information that was successfully transmitted needs to be removed from the buffers. We model an ordered buffer where the information in one transport block has a direct mapping to IP packets. Therefore if that TB gets lost, those packets with information carried in the lost TB stay in the buffer.

This means that block errors may cause packets to arrive out of order. This phenomenon is represented in Figure 3.18 where the size of a TB is set to the same size as a packet for illustration purposes. We see the bits in the transport blocks that did not arrive successfully are kept in the transmission buffer. Thus, if those bits are eventually successfully sent in the future, they would be out of order. Note that this is something common in packet networks. Successfully transmitted TBs get their share of packets removed from the buffers.

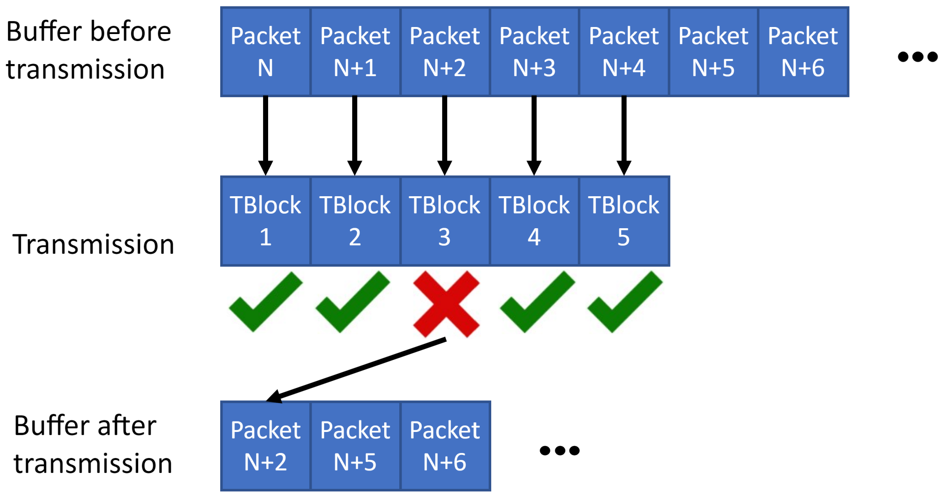


Figure 3.18: Buffer state before and after a transmission, with $S_{TB} = S_{packet}$.

These modelling considerations make the system considerably more realistic compared to a pool-of-bits formulation where no packet has a specific latency budget. Since a packet is dropped when the time it has passed since it arrived in the buffer exceeds the latency budget for the radio link, such realistic modelling considerations increase the likelihood that a packet is discarded due to excessive delay, so the latencies supported in a given scenario are higher.

Finally, the throughputs are used in Equation (3.20) to update the PF ratio.

Conclusion

In this section we presented the required steps to perform data transmission. We started with TTI identification and slot format. Then, we modelled CSI acquisition and presented an intuitive way of creating a GoB. Afterwards, in the user scheduling steps, we addressed SINR estimation, instantaneous throughput calculation from an SINR, user priority calculation, user co-scheduling, power control and finally the selection of MCS for the transmission, adjusted with a link adaptation algorithm.

Finally, we simulate a data transmission, where we introduced calculations on the number and size of TBs. Then we presented a flexible SINR framework and a broadly accepted SINR aggregation algorithm, respectively, to compute the SINR each UE experienced in a given transmission in each PRB, and to aggregate those SINRs in one effective SINR that describes the quality of the transmission. Lastly, we used that SINR to determine the block errors, which are in turn used to adapt the link adaptation parameter, to remove the TB contents from the buffer in case it is successfully transmitted and to update the schedulers PF ratios.

In the next section we simulate single and multi-user scenarios and assess the relations between several parameters described in this section.

4

Results

Contents

4.1 Scenarios

This chapter presents the results that validate the modelling framework. In particular, this section defines the scenarios concretely, providing values to the parameters defined in the (previous) modelling chapter.

Firstly, we analyse the case with only one user. In this case, there is no interference thus the SINR is very high. Still with one user, we obtain the maximum achievable bit rate, which requires us to change the average DL application-layer bit rate to a much higher value to emulate an always-full buffer.

Posteriorly, we simulate four physically present users. In doing so, we see how the received signal strength varies across time non-identically for different users, how the link adaptation parameter OLLA changes depending on block errors, and correlate different metrics with one another, proving the consistency of the model.

Parameters

With the exception of the number of physical users present in the meeting room N_{phy} and the average downlink application bit rate $\overline{R}_{DL}$, there are no changes in the parameters throughout the chapter. See the values of the all necessary parameter below: Table 4.1 contains parameters for the application layer and Table 4.2 contains parameters from the radio layer.

The parameter N_{phy} changes from the first section to the second, respectively, from $N_{phy} = 1$ to $N_{phy} = 4$. The parameter $\overline{R}_{DL}$ is always 80 Mbps with the exception of one plot, Figure 4.2, where it is put to 500 Mbps.

Most parameters have the values from the examples provided when introducing the parameter in the previous Chapter. Some parameters without examples were, e.g. the number of slots between CSI anc scheduling updates, respectively, N_{slots}^{CSI} N_{slots}^{SCH}. We attempt to make the case-study as simple as possible, so we make both unitary. For the same reason, we set the acknowledgement delay τ_{ACK} to zero. This means the data that has errors in one slot can be sent again in the next.

Like in the modelling, UL is disabled. Therefore, there are no UL slots, resulting in $s_{TDD} = 0$. Furthermore, a heavy slot format is considered, where all symbols are of the same type as the slot, DL in this case. However, we do not take into account signalling and slot-format overheads, i.e. $\eta_{OH} = \eta_{slot} = 1$. Thus, we compute a radio-layer bit rate, not an application-layer throughput.

We perform a simulation for the duration T_{sim} of 1 second, such that we can see more precisely the variations and accurately make correlations between variables. We use numerology two because it is the only numerology present in sub-6 GHz and in mmWAves. With $\mu = 2$, one second corresponds to 4000 TTIs. Therefore whenever a plot shows time on the horizontal axis, each second holds 4000 values.

Some final considerations and clarifications regarding the choice of parameters:

- N_{vir} must change with N_{phy} to keep eight users around the table such that each present user has the same head movement patterns in both scenarios;
- $\mathcal{F}$ represents the set of simulated frequencies. We simulate both 3.5 and 26 GHz. Since every conclusion and relation between parameters apply to any frequency, the results for 26 GHz are in Appendix E, due to space constraints;
- We use $P_{t,max}^{BS} = 0.1$ Watt [94], which is the same value used in Wi-Fi access points in the 2.4 GHz band, thus it is a safe and conservative approach.

Table 4.1: Application layer parameters.

Variable	Value	Variable	Value	Variable	Value
N_{phy}	1 or 4	S_{packet}	1500 B	r_t	1 m
N_{vir}	7 or 4	δ	3	r_u	1.2 m
N_{cam}	0	σ_x	0.667 m	R_{DL}	80 Mbps
S_{room}	[6, 6, 3] m	σ_y	0.667 m	S_{GoP}	6
h_{user}	1.4 m	σ_z	0.223 m	$r_{P/I}$	0.2
h_{table}	1 m	β_x	$\pi/9$ rad	R_F	30
d_o	0.15 m	β_y	$\pi/6$ rad	γ	0.5
d_u	0.6 m	β_z	$\pi/9$ rad	o	0

Table 4.2: Radio layer parameters.

Variable	Value	Variable	Value	Variable	Value
T_{sim}	1 s	τ_{ACK}	0	t_w	8000
$\mathcal{F}$	$\{3.5, 26\}$ GHz	τ_{CSI}	4	N_{TB}	5
μ	2	$P_{t,max}^{BS}$	0.1 W	NF_{UE}	8 dB
B	40 MHz	N_{slots}^{CSI}	1	T	290 K
N_{PBR}	50	N_{slots}^{SCH}	1	$[a_\phi, b_\phi]$	[-60° , 60°]
N_{BS}	1	N_{CSI}	1	$[a_\theta, b_\theta]$	[-60° , 60°]
N_{ant}^{UE}	3.5 GHz: 2x2 26 GHz: 8x8	κ	1	r_ϕ	3.5 GHz: 30° 26 GHz: 4°
N_{ant}^{BS}	3.5 GHz: 4x4 26 GHz: 16x16	$BLER_0$	0.1	r_θ	3.5 GHz: 30° 26 GHz: 4°
s_{TDD}	0	γ_{OLLA}	0.1	η	0

4.2 Single user

Considering a single user in the conference room, the SINR of that user will be very high compared to a scenario with more than one physically present user, as a result of experiencing no interference. Since intra-cell interference is the only interference term we may have in a single-panel single-layer transmission, the total interference is zero when a single-user is considered and the SINR degenerates in an SNR, reaching values well above the 30 dB mark.

From the analysis of the BLER curves, previously seen in Figure 3.16, when operating with such high SINRs the probability of occurring block errors is practically zero. Hence the bit rate is either constrained by the available bits in the buffer or by the maximum possible bit rate a user can have, and we must identify which situation it is. Figure 4.1 shows the instantaneous bit rate in each TTI and the average bit rate of the past 800 TTIs, or 200 ms, the duration of a GoP.

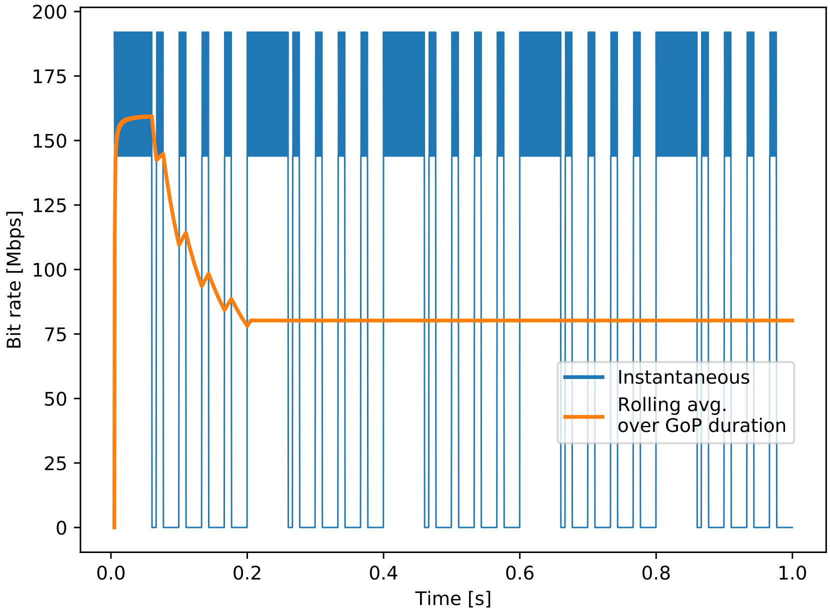


Figure 4.1: Instantaneous and rolling average bit rate with $\overline{R}_{DL} = 80$ Mbps.

Note the instantaneous bit rate should resemble the packet arrival rates of Figure 3.11, since when there are no packets, there cannot be transmissions. The thin blue lines happen when there is no bit rate oscillations. The thick blue blocks are quick bit rate oscillations outside of the imposed by the packet arrival mechanism, e.g. right after the 0.2 and 0.4 second instants.

We can see that these quick instantaneous bit rate oscillations happen between 145 and 190 Mbps, approximately. They occur because the buffer gets empty every TTI, and only achieve the bitrate correspondent to the number of packets available to send. Since the available packets each TTI oscillates, the bit rate does too.

Additionally, if we perform the computation for the highest achievable bit rate under ideal conditions, i.e. no block errors on the highest MCS, with all 50 PRBs, we obtain that the system should support 250 Mbit/s. So, to see this number we must guarantee there are enough packets in the buffer. So, we set $\overline{R}_{DL}$ from 80 Mbps to 500 Mbps exclusively to plot the maximum achievable bit rate in Figure 4.2.

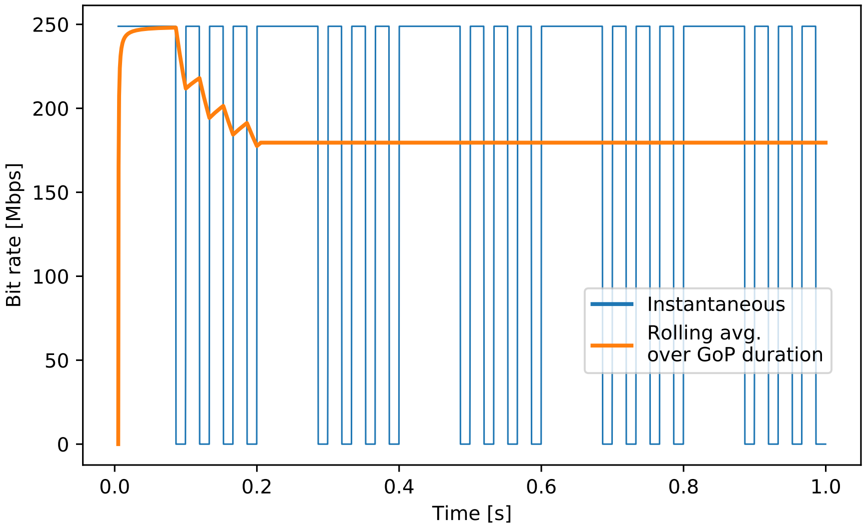


Figure 4.2: Instantaneous and rolling average bit rate with $\overline{R}_{DL} = 500$ Mbps.

Contrary to Figure 4.1, there are no quick instantaneous bit rate oscillations in Figure 4.2 because the number of bits in the buffer always exceeds the bits to be sent in that TTI. The zero-bit rate TTIs are due to an empty buffer - packets get discarded if the scheduler realises they are going to exceed the latency constraints and excess packets come from setting the average arrival rate so high.

Thus we may conclude that maximum average achievable bit rate for a user under the described conditions is roughly 175 Mbps, reaching 250 Mbps in instantaneous bit rate. And observe that the latency plays an important role. The higher the latency, the closer to the maximum instantaneous bit rate the average bit rate gets.

From Figure 4.1 we conclude the deployment and network configurations support an average bit rate of 80 Mbps in the DL for one user. In Figure 4.2 we set the threshold for what is the maximum bit rate a user can achieve. To further expand this quantity, other strategies need to be used, like using an higher MCS or multi-layer transmission.

Finally, is important to note that a conference use case with only one user is a scenario where the resource distribution is trivial and very high bit rates are expected. In the next section we consider a more demanding case of having multiple users to serve simultaneously.

4.3 Multiple users

Let us now consider a more demanding scenario with four physical users, each with a downlink average application bit rate of 80 Mbps. Analogously to Figure 4.2, Figure 4.3 represents the instantaneous bit rate as well as the average bit rate of the last 200 ms, for each present user. Here we see the a more chaotic and realistic scenario.

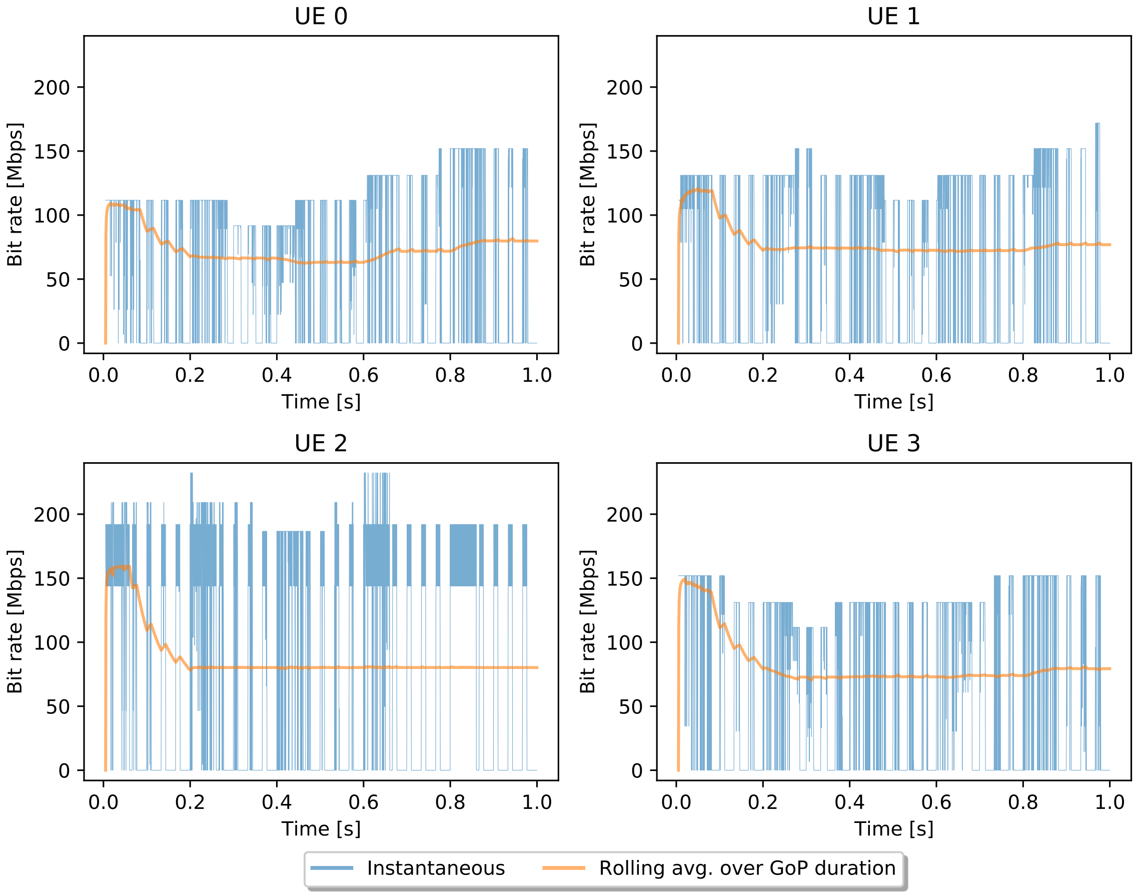


Figure 4.3: UE bit rates, instantaneous and averaged over the last 200 ms (GoP duration).

The average quality of connections varies across users although they are placed at the same distance to the BS, uniformly distributed around the table, as described in Section 3.1.1. This is due to their random head movements, and such movements change HMD antennas' orientations, thus influencing radiations patterns, and consequently the quality of the connection and achievable bit rate.

In Figure 4.3, bit rate oscillations have two causes: the same as in the previous section, i.e. variability of packet arrival rate on a TTI-basis, and variability of channel conditions. In this section we carefully dissect this occurrence

A good indicator of the channel quality variability is the SINR each user experiences. Figure 4.4 shows the SINR estimated from channel measurements and the SINR experienced from the actual transmission. The time-varying experienced received signal power and interference power are presented in Figure 4.5.

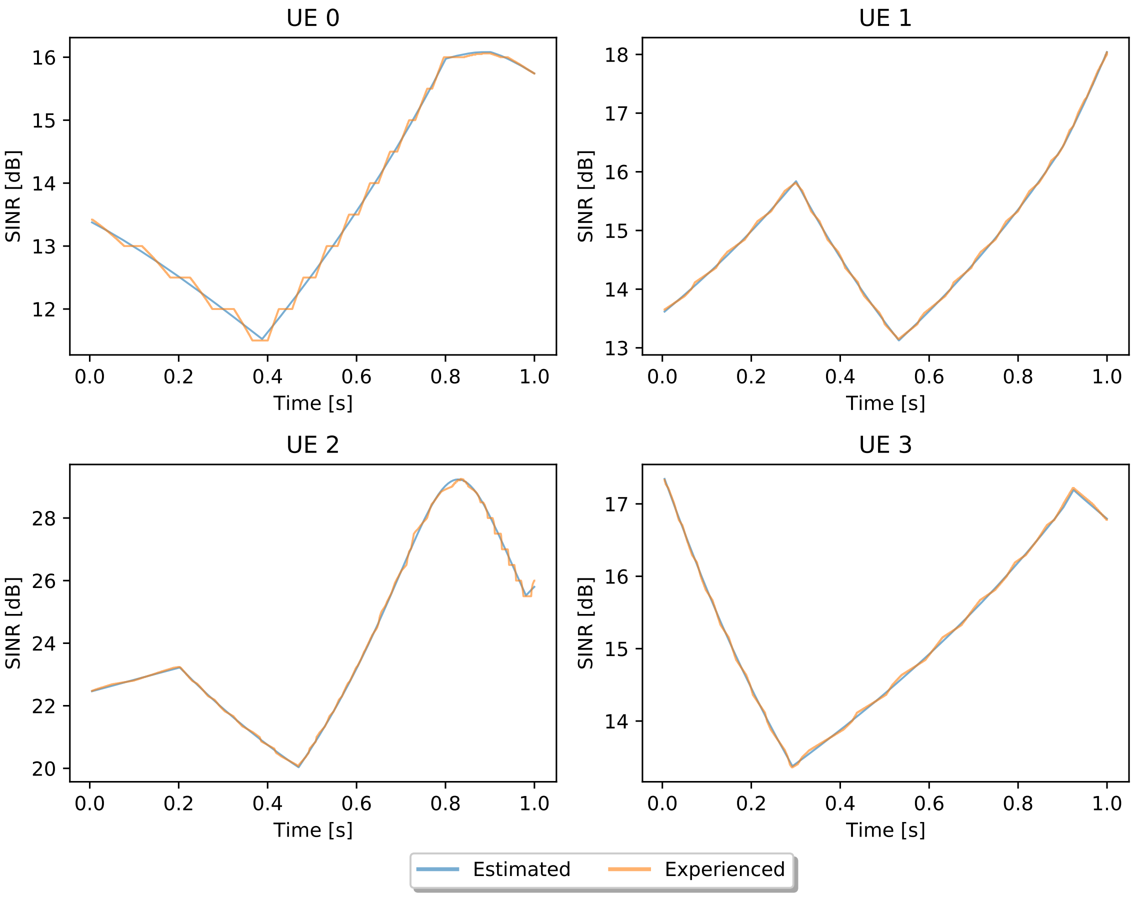


Figure 4.4: SINR, estimated before transmission and experienced during transmission.

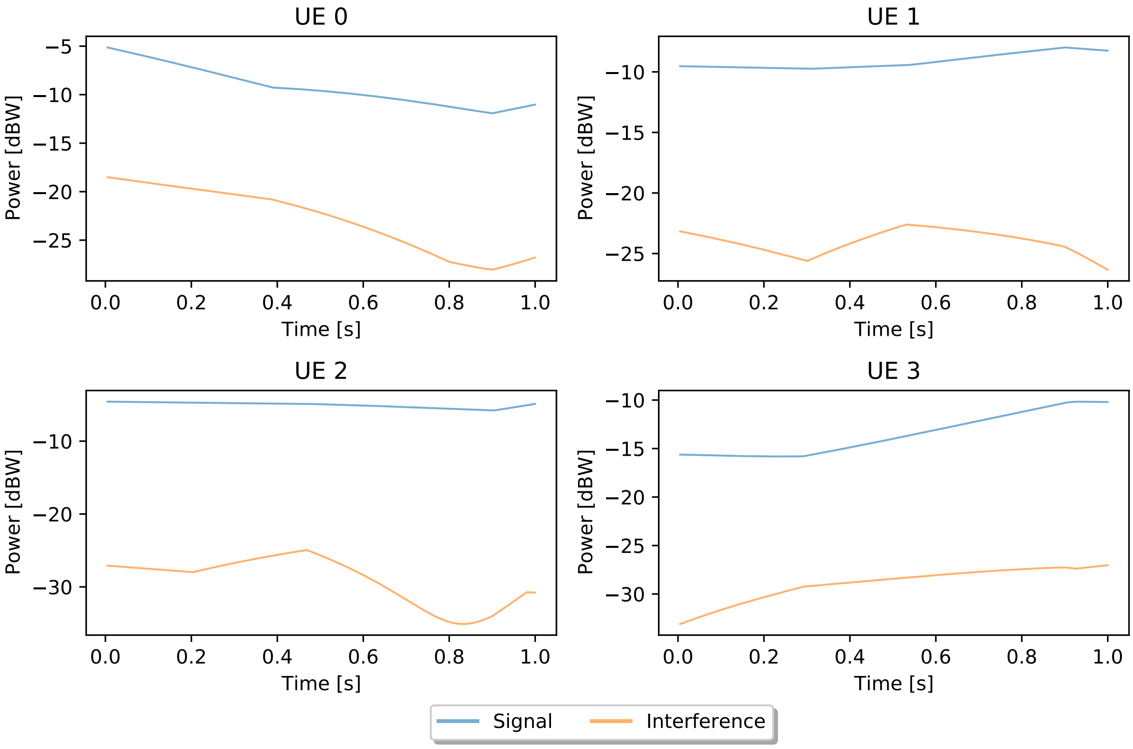


Figure 4.5: Received Signal and Interference Powers.

In Figure 4.4 we see the estimative closely following the experienced. Note that the experienced SINR is only updated when there are transmissions, and as such it is constant for some short periods. Moreover, it agrees perfectly with Figure 4.5. Assuming noise plays no significant role, the SINR is practically the ratio between received power and interference, and we see the SINR changing according to the signal power and interference curves. Actually, since they are in logarithmic units, the SINR is the different between the signal and interference powers.

However, Figure 4.5 also shows unexpected behaviour. It shows abnormally high received signal powers. Taking into account the BS transmit power is 100 mW, we can see that some UEs are receiving more than that, which is certainly wrong. However, it seems to be a problem merely with the scale because all graphs are consistent and lead to realistic SINRs.

Nonetheless, the previous two figures are consistent. Furthermore, we see significant and unpredictable channel variations for each user. With such accentuated channel variability, the choice of MCS varies as well, which justifies variable bit rates.

Figure 4.6 shows the MCS index used for the transmissions to each UE. Assuming the same degree of block errors, the higher the average MCS index, the more likely a user is to get an higher bit rate. We see this relation when comparing the achieved bit rates in Figure 4.3 and the MCS used for each transmission, below.

As expected, sufficient channel variability causes changes in the experienced bit rate. Furthermore, channel variability may lead to block errors, since predicting the future state of the channel is no trivial task and even the slightest drifts between SINR estimation and realisation can lead to using the incorrect MCS.

Figure 4.7 shows the running average BLER across time. And one clearly testifies that all users experience blocks with errors.

This figure agrees with what we have seen so far. For instance, in case of UE 2, we can identify that around the 0.7 second mark the BLER monotonically decreases. This happens the SINR is so high that the highest MCS is always chosen with an estimated BLER smaller than 10%. Indeed, by analysing the MCS curves (Figure 3.16), we see that SINRs above 26 dB, approximately, results in virtually no block errors, therefore driving the average BLER down. Figures 4.4 proves that UE 2 passes this 26 dB threshold at that time.

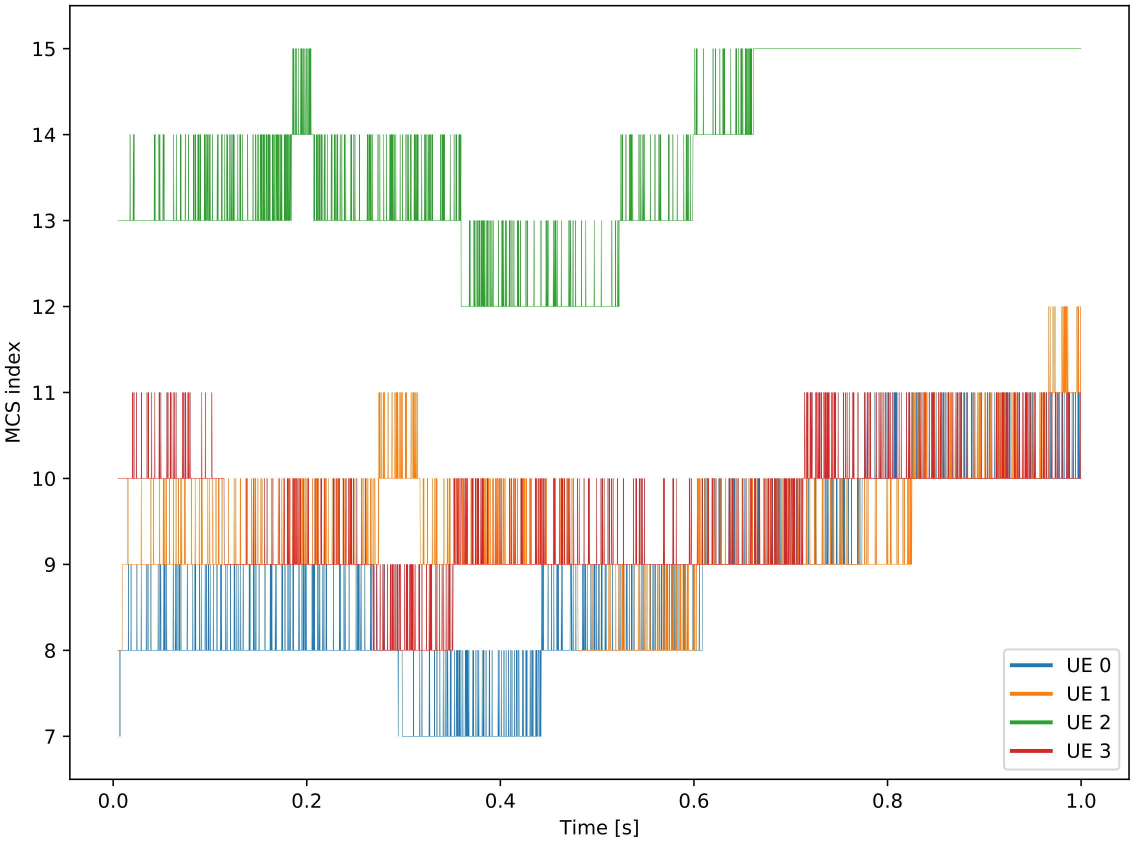


Figure 4.6: MCS index used by each UE in every transmission.

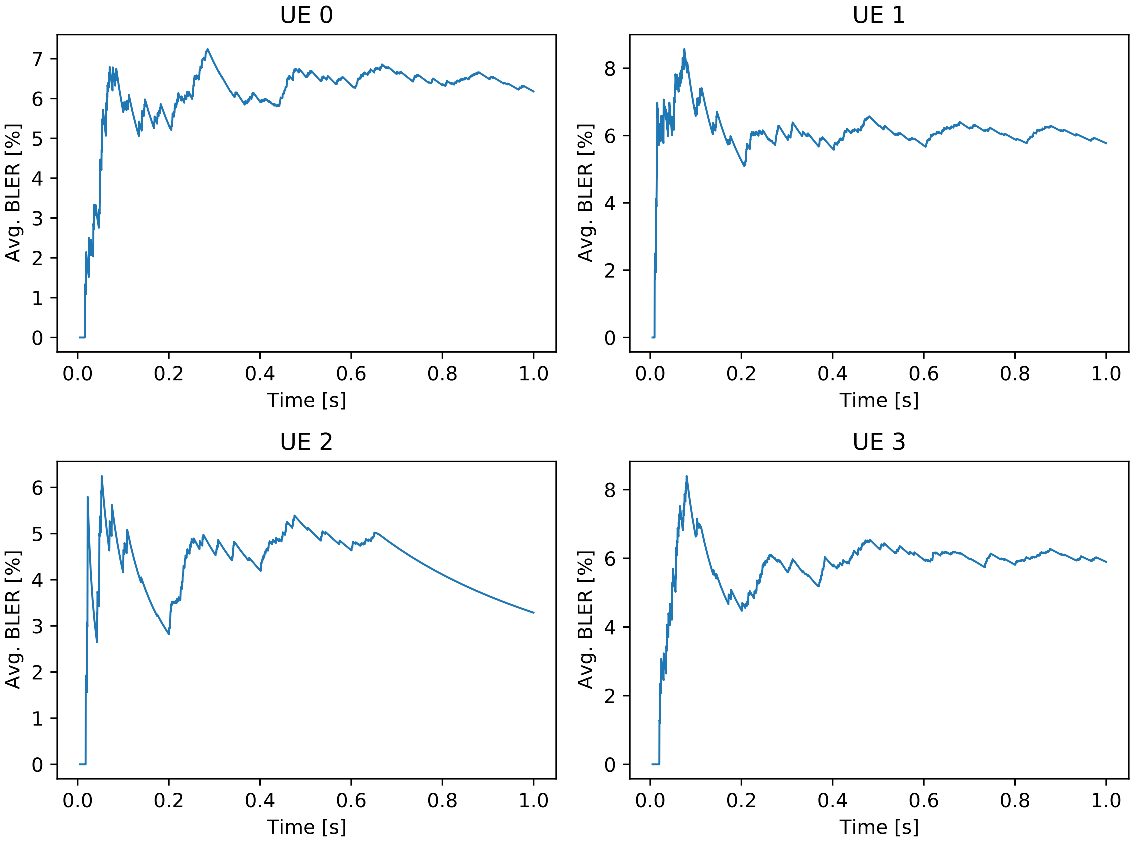


Figure 4.7: All time BLER average in a multi-user scenario.

Figure 4.8 shows the instantaneous BLER for UE 2 and how the OLLA parameter varies accordingly. On close inspection, Δ_{OLLA} rises when there are transmissions without errors and decreases when there are transmissions where blocks had errors. This behaviour leads to choosing a lower MCS when there are errors and slowly opting for an higher MCS when the channel is better than expected. Only when there are transmissions (zones marked in grey) there are blocks with or without errors. Naturally therefore, the OLLA parameter is not updated outside of such zones.

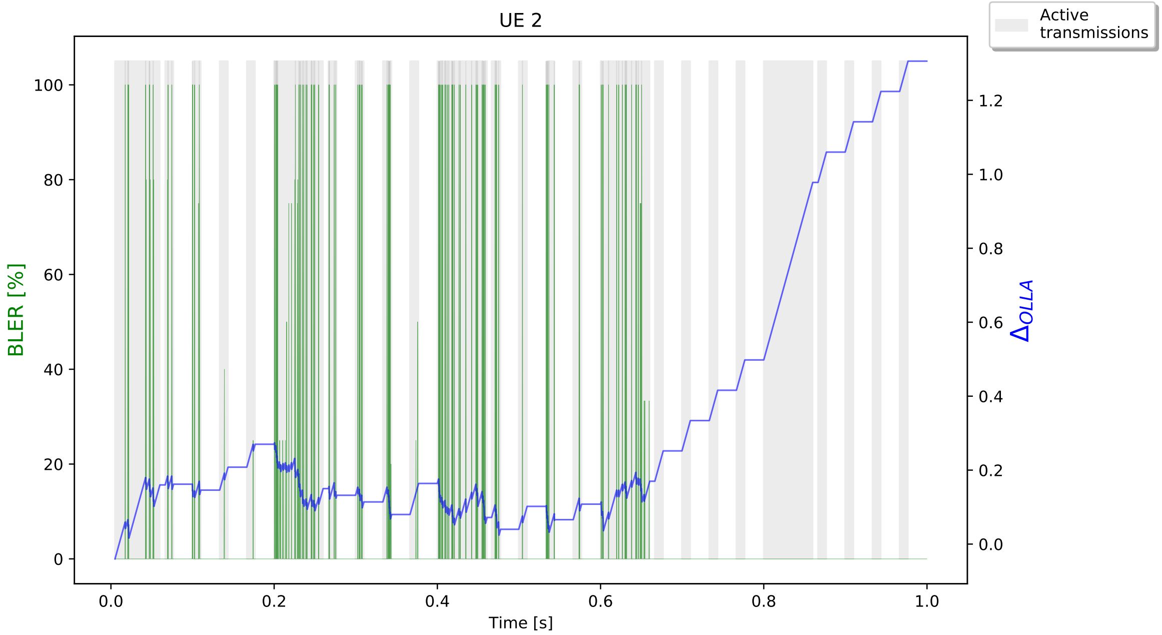


Figure 4.8: Link adaptation parameter variation with the instantaneous BLER. Grey zones mark when the UE has active transmissions.

The use of this link adaptation scheme should make the BLER converge to the BLER target of 10%. It still is uncertain why that does not happen. A possibility is the excessively short simulation. There are artifacts from before the convergence of the link adaptation mechanism which can drive the BLER down, and in short simulations they may still have a significant influence on the average BLER.

Finally, we can conclude regarding how well the deployment and configurations cope with the requirements. However, it should not be taken as a final statement because of the modelling simplifications listed in the beginning of this chapter, some results are yet to be understood, and this analysis lacks statistical significance: we cannot derive such conclusions by looking at one second of one use-case. But we can preliminarily conclude something about that one second despite all limiting factors.

Nevertheless, to conclude something relevant we cannot resort exclusively to bit rate plots since those say nothing about performance with respect to latencies. From the application perspective, we want to answer questions such as: “what is maximum application throughput that achieves less than X % packet error rate?”

A sizeable step towards answering it, and also towards improving the answer is understanding under what circumstances high packet loss occurs.

In Figure 4.9 we see the average packet latency, i.e. average time a packet takes to be successfully sent after arriving to the buffer, and the packet drop rate on a per-frame basis. This means the horizontal axis has application frame indices - five GoP are sent per second, and each I-frame is marked in red. As a general rule, both the average packet latency and drop rate rise right after an I-frame, because those are the times where the system is under the most load. This, however, does not apply for UE2 since its channel quality is good enough to sustain the load. Nonetheless, we correlate the moment of most stress of UE 2, marked by the peak of average packet latency, with the poorest channel quality of Figure 4.4.

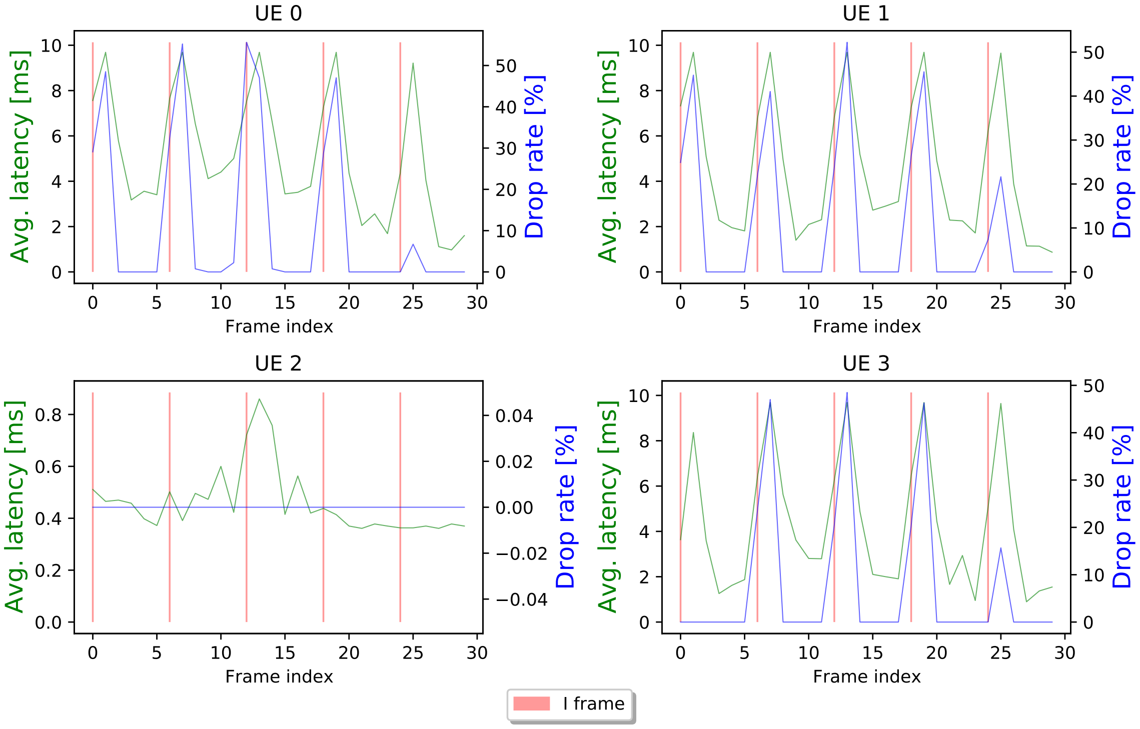


Figure 4.9: Packet latencies and drop-rates for all UEs with I-frame marking.

If packets are dropped after the I frame, it means the link had enough quality to cope with the I frame, but having a P frame following with no pause led to the delay of some packets past the latency requirement. This is the case when the drop rate peak comes after the I-frame, which is the most common case. And the lower the drop rate peak is, the closer the user was to handling all packets within the required time. For instance, UE 1 has severe difficulties handling all data from the third I-frame, but is only second in performance to UE 2 in the last frame. As looking at the SINR plots in Figure 4.4 again, we see that UE 1 has the lowest SINR at 0.4 second mark (time of third I-frame), but has the highest SINR with the exception of UE2 at the 0.8 second mark (time of the last I-frame).

Conclusions

This chapter objective was to validate the modelling framework.

In this chapter we started by defining the scenario as clearly as possible, as well as any simplifications that took place while simulating such scenario. We provided values to all variables we had previously defined in the Methodology Chapter.

Then, we assessed two reasons for rapid bit rate oscillations. One reason is when the number of packets arriving each TTI oscillates and the achieved bit rate is capped by the number of packets in the buffer. This happens when the channel is very good, which is the situation in Section 4.2. And we have naturally concluded that we can only measure accurately what bit rate the link can achieve when the incoming packet bit rate surpasses the achievable bit rate.

The other possibility is when the channel is not optimal, to the point of causing some blocks to have errors, resulting in oscillations in the achieved bit rate. A mix case of phases with bit errors and phases with stable conditions is also possible, and often the case, which makes the situation harder to dissect and demands resorting to other metrics as well. Section 4.3 was dedicated to exposing some of the metrics we can extract from simulations in order to assess these more complex situations. Also, the most important relations between metrics were put forth.

Although conclusions on radio layer configurations and deployment need to be done carefully given the light statistical evidence, we proved how they can be derived. For the scenario with multiple users, the requirements of 80 Mbps and 10 ms were only possible for UE 2, under the current setting. Actually, UE 2 supported such bit rate with sub-millisecond latencies. Next steps include expanding the quantity of acquired data from each simulation, longer simulations to enable correlation between events on a larger time scale, such as the head rotation, and altogether simulations on different channel realisations.

Moreover, we have identified places that require further attention. Certainly some adjustments to the modelling and implementation are required to solve the few observed inconsistencies. Nonetheless, we have shown working simulations with predominantly coherent results.

5

Conclusion

Conclusion

This section brings this thesis to a closure. Here we make a summary and present the most prominent future work directions.

In this thesis, we investigated and validated modelling methodologies to simulate the radio access for a XR conference meeting application. Our main goal was to create a complete framework that allows us to perform sensitivity analysis and that way enable assessments on how deployment and configurations impact application performance. With such insights we can derive deployment guidelines, like the number of antennas, the position and number of the base stations, and spectrum assignments. Additionally, we can create autonomously managed systems that automatically adjust radio-layer configurations, such as scheduling parameters and beamforming algorithms, in order to enhance the service provision given certain available resources and a given channel state.

Naturally, such task requires an extensive background and literature review since the majority of the components we combine into a framework have had plenty of research.

We do this in Chapter 2. Firstly, social virtual reality meeting applications are seen in lights of network requirements. Then we probe what aspects in the application have influence in the radio channel, thus we investigate where cameras, headsets and base stations are placed, and how users move in a meeting. Then, we review traffic models and read on different approaches to optimise the radio layer for this demanding real-time application.

We proceed to survey the key technologies of today. Massive MIMO and mmWaves are the answer, and in the same section we present what they entail. Then we examine how these technologies are standardised and used in today's emerging 5G networks. We also inspect relevant physical layer specifications and network equipment. We conclude the background by selecting the radio channel generator that creates the propagation environment on top of which we simulate transmissions.

Chapter 3 is where we show how everything comes together in our simulation framework. We present all modelling considerations, starting with the application. We describe the physical setting in the room, what antennas are used and how they are placed. We model the head movement of a user and we detail a flexible application traffic model based on video streaming.

Subsequently, we show how the propagation environment changes with the previous application considerations, like antenna placement and user behaviour. The radio channel is measured by the network to make resource management decisions. We present and model the tasks carried by the network. These include channel state information acquisition, user scheduling and the actual transmission.

In Chapter 4 is assessed the result of all modelling considerations. In the previous chapter we defined the parameters that dictate all modelled aspects and in this chapter we give them values to precisely describe the simulation scenarios. We see how the channel quality changed from a conference with a single user to a multi-user conference. More importantly, we verified that our modelling produces realistic results that relate coherently among themselves. We can measure throughputs and latencies and quantify the QoS from a set of application, propagation environment and network settings. This tells us this simulator has been successfully designed.

This work forms a basis and provides tools for further research.

Future Work

We foresee an autonomously managed Social XR network slice. One clear direction of work is to expand modelling to the remaining of the network, beyond the radio access. However, there are numerous challenges on the radio layer will likely constitute bottlenecks as the application requirements increase. Therefore we focus our future work analysis on radio access.

Towards achieving this vision, we identify the following viable future work directions:

- Perform extensive sensitivity analysis - parameters in the simulator can be changed and performance measured in order to derive insights about how settings impact performance, both in combinations and individually. Some examples of settings to change are: numbers of meeting participants, user behaviour, antenna arrays' placement, size, architecture, and geometry, bandwidths, frequency bands, numerologies, settings of the latency-aware packet scheduler, multi-user scheduling strategies, TDD splits, number of base stations and their location, multi-BS operation algorithms (D-MIMO), acknowledgement delays, channel state information and scheduling periodicities, target BLER and link adaptation parameter.

- Improvements to make the simulator more realistic - some modelling considerations can be considered simplistic. More realism can be achieved by improving modelling. Moreover, such improvements often lead to thoughts on how something can be done differently, and generate more work directions;

- Flexible slots and mini-slots - we mention in the background that one of the most important advancements of 5G New Radio is a flexible slot-based transmission structure. The standards also allow for symbol-based transmissions (mini-slots) [95], although with more constraints. Nonetheless, it provides unseen granularity in the time domain allowing lower latencies;

- Reciprocity-based beamforming - the performance difference between GoB and reciprocity-based beamforming has not been studied in mmWaves [64]. This may require to quantifying overheads of CSI-RS and SRS;
- Development of new resource management mechanisms, possibly AI-based;
- Human blocking - mmWaves are more susceptible to blockages than lower frequencies. This direction includes human blockage modelling and the development of measures and procedures to attenuate the impact of a blockage event. More precisely, examples of possibles solutions may be smart multi-BS operation algorithms, or the usage of additional hardware like intelligent reflective surfaces [96] or relays. It may be the case where ray tracing simulations are required to have an accurate representation of the reflections;
- Slice interaction and management - when there are conflicts between slices, how to solve them? The process should require quantifying how much each slice needs a given modification to the network, and what priority does that slice have and use that information to make slices interact seemingly.

Solely for sake of conciseness, this selection is nowhere near exhaustive. During development of each component of the framework, plenty of more detailed research directions have been identified. So much so, that we intend to continue working in this exciting project beyond this thesis.

Bibliography

[1] ITU-R, *Rec. M.2083*, September 2015. [Online]. Available: https://www.itu.int/rec/R-REC-M.2083-0-201509-I/en

[2] ETSI, *Why do we need 5G?* [Online]. Available: https://www.etsi.org/technologies/5g

[3] M. Vaughn (directors); M. Vaughn, D. Reid and A. Bohling (producers), "Kingsman: The Secret Service," *20th Century Fox*, 2014.

[4] I. W. User, *Wikimedia Commons: Fourier Unit Pulse.*, at commons.wikimedia.org/wiki/File:Fourier_unit_pulse.svg, 2006.

[5] V. Milosevic, B. Jokanovic, O. Boric-Lubecke, and V. Lubecke, "Key Microwave and Millimeter Wave Technologies for 5G Radio," *Powering the Internet of Things With 5G Networks*, 2017.

[6] Sploty: Telecom Explained, "LTE E-UTRAN open loop spatial multiplexing - TM3," *from sploty.com/en/lte-e-utran-open-loop-spatial-multiplexing-tm3.html*, 2014.

[7] George N. Gibson, "Physics Notes Section 5.2 - Constructive and Destructive Interference," *from phys.uconn.edu/ gibson/Notes/Section5_2/Sec5_2.htm*.

[8] Keysight, *Concepts of Orthogonal Frequency Division Multiplexing (OFDM) and 802.11 WLAN.*, RF & Microwave web help files from rfmw.em.keysight.com, 2020.

[9] S. W. User, *Gray coding Constellation Diagrams.*, from user wikipedia.org/wiki/User:Splash, 2006.

[10] S. Pratschner, B. Tahir, L. Marijanovic, M. Mussbah, K. Kirev, R. Nissel, S. Schwarz, and M. Rupp, "Versatile mobile communications simulation: the Vienna 5G Link Level Simulator," *EURASIP Journal on Wireless Communications and Networking*, 2018.

[11] C. A. Balanis, *Antenna Theory: Analysis and Design.*, 4th Edition, Wiley, 2016.

[12] 3GPP, *TS 38.214 - 5G NR; Physical layer procedures for data.*, v15.3.0, Rel. 15, 2020.

[13] Harri Holma and Antti Toskala and Takehiro Nakamura, *5G Technology: 3GPP New Radio.*, Wiley, 2019.

[14] Ericsson, *5 key facts about 5G radio access networks.*, White Paper, 2020.

[15] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wireless Communications*, 2017.

[16] A. Kaloxylos, "A Survey and an Analysis of Network Slicing in 5G Networks," *IEEE Communications Standards Magazine*, 2018.

[17] S. Spielberg (directors); D. De Line, K. M. Krieger, S. Spielberg and D. Farah (producers), "Ready Player One," *Warner Bros. Pictures*, 2018.

[18] F. Hu, Y. Deng, W. Saad, M. Bennis, and A. H. Aghvami, "Cellular-Connected Wireless Virtual Reality: Requirements, Challenges, and Solutions," *IEEE Communications Magazine*, 2020.

[19] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward Low-Latency and Ultra-Reliable Virtual Reality," *IEEE Network*, 2018.

[20] E. Bastug, M. Bennis, M. Medard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Communications Magazine*, 2017.

[21] M. Hosseini, "View-aware tile-based adaptations in 360 virtual reality video streaming," *2017 IEEE Virtual Reality (VR)*, 2017.

[22] Vive, "Vive Cosmos Elite VR headset specifications," *from vive.com*, 2020.

[23] PiMax, "Vision 8k X VR headset specifications," *from pimax.com*, 2020.

[24] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, "VR is on the Edge: How to Deliver 360° Videos in Mobile Networks," *Association for Computing Machinery*, 2017.

[25] 3GPP, *TR 26.928 - Extended Reality (XR) in 5G.*, v16.0.0, Release 16, 2020.

[26] S. Dijkstra-Soudarissanane, K. E. Assal, S. Gunkel, F. t. Haar, R. Hindriks, J. W. Kleinrouweler, and O. Niamut, "Multi-Sensor Capture and Network Processing for Virtual Reality Conferencing," *Association for Computing Machinery*, 2019.

[27] S. N. B. Gunkel, H. M. Stokking, M. J. Prins, N. van der Stap, F. B. t. Haar, and O. A. Niamut, “Virtual Reality Conferencing: Multi-User Immersive VR Experiences on the Web,” *Association for Computing Machinery*, 2018.

[28] S. Fremerey, A. Singla, K. Meseberg, and A. Raake, “AVtrack360: An Open Dataset and Software Recording People’s Head Rotations Watching 360° Videos on an HMD,” *Association for Computing Machinery*, 2018.

[29] E. J. David, J. Gutiérrez, A. Coutrot, M. P. Da Silva, and P. L. Callet, “A Dataset of Head and Eye Movements for 360° Videos,” *Association for Computing Machinery*, 2018.

[30] M. Chen, W. Saad, and C. Yin, “Virtual Reality Over Wireless Networks: Quality-of-Service Model and Learning-Based Resource Management,” *IEEE Transactions on Communications*, 2018.

[31] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, “Cutting the Cord in Virtual Reality,” *Association for Computing Machinery*, 2016.

[32] X. Ge, L. Pan, Q. Li, G. Mao, and S. Tu, “Multipath Cooperative Communications Networks for Augmented and Virtual Reality Transmission,” *IEEE Transactions on Multimedia*, 2017.

[33] X. Yang, Z. Chen, K. Li, Y. Sun, N. Liu, W. Xie, and Y. Zhao, “Communication-Constrained Mobile Edge Computing Systems for Wireless Virtual Reality: Scheduling and Tradeoff,” *IEEE Access*, 2018.

[34] Global Mobilie Suppliers Association, “Spectrum Pricing April 2020 update,” *from gsacom.com/paper/spectrum-pricing-april-2020/*, 2020.

[35] S. Rangan, T. S. Rappaport, and E. Erkip, “Millimeter-Wave Cellular Wireless Networks: Potentials and Challenges,” *Proceedings of the IEEE*, 2014.

[36] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, “Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!” *IEEE Access*, 2013.

[37] Theodore S. Rappaport, *Wireless Communications: Principles and Practice.*, 2nd ed., Prentice Hall, 2002.

[38] H. Zhao, R. Mayzus, S. Sun, M. Samimi, J. K. Schulz, Y. Azar, K. Wang, G. N. Wong, F. Gutierrez, and T. S. Rappaport, “28 GHz millimeter wave cellular communication measurements for reflection and penetration loss in and around buildings in New York city,” *IEEE International Conference on Communications (ICC)*, 2013.

[39] L. Sanguinetti, E. Björnson, and J. Hoydis, “Toward Massive MIMO 2.0: Understanding Spatial Correlation, Interference Suppression, and Pilot Contamination,” *IEEE Transactions on Communications*, 2020.

[40] E. Björnson, E. G. Larsson, and T. L. Marzetta, “Massive mimo: ten myths and one critical question,” *IEEE Communications Magazine*, 2016.

[41] E. Björnson, E. G. Larsson, and M. Debbah, “Massive MIMO for Maximal Spectral Efficiency: How Many Users and Pilots Should Be Allocated?” *IEEE Transactions on Wireless Communications*, 2016.

[42] T. L. Marzetta, “Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas,” *IEEE Transactions on Wireless Communications*, 2010.

[43] B. M. Hochwald, T. L. Marzetta, and V. Tarokh, “Multiple-antenna channel hardening and its implications for rate feedback and scheduling,” *IEEE Transactions on Information Theory*, 2004.

[44] X. Li, E. Björnson, S. Zhou, and J. Wang, “Massive MIMO with multi-antenna users: When are additional user antennas beneficial?” *2016 23rd International Conference on Telecommunications (ICT)*, 2016.

[45] Andrea Goldsmith, *Wireless Communications.*, Cambridge University Press, 2005.

[46] David Tse and Pramod Viswanath, *Fundamentals of Wireless Communication.*, Cambridge University Press, 2005.

[47] Andrea Goldsmith, *Wireless Communications.*, Cambridge University Press, 2005.

[48] Y. Huang, Y. Li, H. Ren, J. Lu, and W. Zhang, “Multi-Panel MIMO in 5G,” *IEEE Communications Magazine*, 2018.

[49] U. Madhow, D. R. Brown, S. Dasgupta, and R. Mudumbai, “Distributed massive MIMO: Algorithms, architectures and concept systems,” *Information Theory and Applications Workshop (ITA)*, 2014.

[50] S. Braam, R. Litjens, P. Smulders, and W. IJntema, “Assessment of Distributed Multi-User MIMO Transmission in 5G Networks,” *Proceedings of the 18th ACM Symposium on Mobility Management and Wireless Access, Association for Computing Machinery*, 2020.

[51] 3GPP, *TS 38.901 - 5G; Study on channel model for frequencies from 0.5 to 100 GHz.*, v16.1.0, Rel. 16, 2020.

[52] S. Ghosh and D. Sen, “An Inclusive Survey on Array Antenna Design for Millimeter-Wave Communications,” *IEEE Access*, 2019.

[53] E. F. W. Alexanderson, “Transatlantic radio communication,” *Proceedings of the American Institute of Electrical Engineers*, 1919.

[54] O. E. Ayach, R. W. Heath, S. Abu-Surra, S. Rajagopal, and Z. Pi, “The capacity optimality of beam steering in large millimeter wave MIMO systems,” *IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2012.

[55] T. K. Y. Lo, “Maximum ratio transmission,” *IEEE Transactions on Communications*, 1999.

[56] Qualcomm, *Making 5G NR a Commercial Reality: A unified, more capable 5G air interface.*, White Paper, 2020.

[57] 3GPP, *TS 38.331 - 5G NR; Radio Resource Control (RRC); Protocol specification.*, v16.2.0, Rel. 16, 2020.

[58] Qualcomm, *The 3GPP Release-15 5G NR Design.*, Presentation, 2018.

[59] N. Bhushan, T. Ji, O. Koymen, J. Smee, J. Soriaga, S. Subramanian, and Y. Wei, “5G Air Interface System Design Principles,” *IEEE Xplore*, 2017.

[60] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, “Initial access frameworks for 3GPP NR at mmWave frequencies,” *17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2018.

[61] Y. R. Li, B. Gao, X. Zhang, and K. Huang, “Beam Management in Millimeter-Wave Communications for 5G and Beyond,” *IEEE Access*, 2020.

[62] E. Dahlman, S. Parkvall, and J. Sköld, *5G NR: the Next Generation Wireless Access Technology.*, 4th Edition, Academic Press, 2018.

[63] E. Dahlman, S. Parkvall, and J. Sköld, “4G, LTE-Advanced Pro and The Road to 5G - Channel-State Information and Full-Dimension MIMO,” *Academic Press,3rd ed.*, 2016.

[64] J. Flordelis, F. Rusek, F. Tufvesson, E. G. Larsson, and O. Edfors, “Massive MIMO Performance—TDD Versus FDD: What Do Measurements Say?” *IEEE Transactions on Wireless Communications*, 2018.

[65] Qualcomm, *QTM527 mmWave antenna module - Feature sheet.*, at qualcomm.com/products/qtm527-mmwave-antenna-modules, 2020.

[66] Y. Ren, X. Su, C. Qi, and Y. Wang, “Channel Reconstruction for SVD-ZF Precoding in Massive 3D-MIMO Systems: Low-Complexity Algorithm,” *IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016.

[67] E. Björnson, J. Hoydis, and L. Sanguinetti, “Massive MIMO Has Unlimited Capacity,” *IEEE Transactions on Wireless Communications*, 2018.

[68] E. Björnson, J. Hoydis, and L. Sanguinetti, “Pilot contamination is not a fundamental asymptotic limitation in massive MIMO,” *IEEE International Conference on Communications (ICC)*, 2017.

[69] J. Flordelis, F. Rusek, F. Tufvesson, E. G. Larsson, and O. Edfors, “Massive MIMO Performance—TDD Versus FDD: What Do Measurements Say?” *IEEE Transactions on Wireless Communications*, 2018.

[70] 3GPP, *TS 38.104 - 5G NR; Base Station (BS) radio transmission and reception.*, v16.5.0, Rel. 16, 2020.

[71] 3GPP, *TS 38.331 - 5G NR; User Equipment (UE) radio access capabilities.*, v16.3.0, Rel. 16, 2020.

[72] T. K. Sarkar, Zhong Ji, Kyungjung Kim, A. Medouri, and M. Salazar-Palma, “A survey of various propagation models for mobile communication,” *IEEE Antennas and Propagation Magazine*, 2003.

[73] F. Bohagen, P. Orten, and G. E. Oien, “On spherical vs. plane wave modeling of line-of-sight MIMO channels,” *IEEE Transactions on Communications*, 2009.

[74] I. Carton, Wei Fan, P. Kyösti, and G. F. Pedersen, “Validation of 5G METIS map-based channel model at mmwave bands in indoor scenarios,” *10th European Conference on Antennas and Propagation (EuCAP)*, 2016.

[75] C. Wang, J. Bian, J. Sun, W. Zhang, and M. Zhang, “A Survey of 5G Channel Measurements and Models,” *IEEE Communications Surveys Tutorials*, 2018.

[76] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, “QuaDRiGa: A 3-D Multi-Cell Channel Model With Time Evolution for Enabling Virtual Field Trials,” *IEEE Transactions on Antennas and Propagation*, 2014.

[77] Remcom, “Wireless InSite - 3D Wireless Prediction Software,” *remcom.com/wireless-insite*.

[78] S. Sun, G. R. MacCartney, and T. S. Rappaport, “A novel millimeter-wave channel simulator and applications for 5G wireless communications,” *2017 IEEE International Conference on Communications (ICC)*, 2017.

[79] M. K. Müller, F. Ademaj, T. Dittrich, A. Fastenbauer, B. R. Elbal, A. Nabavi, L. Nagel, S. Schwarz, and M. Rupp, "Flexible multi-node simulation of cellular mobile communications: the Vienna 5G System Level Simulator," *EURASIP Journal on Wireless Communications and Networking*, 2018.

[80] MathWorks, "Matlab 5G Toolbox," *at mathworks.com/products/5g.html*, 2018.

[81] D. He, B. Ai, K. Guan, L. Wang, Z. Zhong, and T. Kürner, "The Design and Applications of High-Performance Ray-Tracing Simulation Platform for 5G and Beyond Wireless Communications: A Tutorial," *IEEE Communications Surveys and Tutorials*, 2019.

[82] C. Jao, C. Wang, T. Yeh, C. Tsai, L. Lo, J. Chen, W. Pao, and W. Sheen, "WiSE: A System-Level Simulator for 5G Mobile Networks," *IEEE Wireless Communications*, 2018.

[83] Y. Kim, J. Bae, J. Lim, E. Park, J. Baek, S. I. Han, C. Chu, and Y. Han, "5G K-Simulator: 5G System Simulator for Performance Evaluation," *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2018.

[84] J. Lu, D. Steinbach, P. Cabrol, and P. Pietraski, "Modeling human blockers in millimeter wave radio links," *ZTE Communications Magazine*, 2012.

[85] Jong-Hun Rhee, J. M. Holtzman, and Dong-Ku Kim, "Scheduling of real/non-real time services: adaptive EXP/PF algorithm," *The 57th IEEE Semiannual Vehicular Technology Conference, Spring*, 2003.

[86] F. Afroz, K. Sandrasegaran, and P. Ghosal, "Performance analysis of PF, M-LWDF and EXP/PF packet scheduling algorithms in 3GPP LTE downlink," *2014 Australasian Telecommunication Networks and Applications Conference (ATNAC)*, 2014.

[87] R. Basukala, H. A. M. Ramli, and K. Sandrasegaran, "Performance analysis of EXP/PF and M-LWDF in downlink 3GPP LTE system," *First Asian Himalayas International Conference on Internet*, 2009.

[88] K. I. Pedersen, G. Monghal, I. Z. Kovacs, T. E. Kolding, A. Pokhariyal, F. Frederiksen, and P. Mogensen, "Frequency Domain Scheduling for OFDMA with Limited and Noisy Channel Feedback," *IEEE 66th Vehicular Technology Conference*, 2007.

[89] E. Tuomaala and Haiming Wang, "Effective SINR approach of link to system mapping in OFDM/multi-carrier mobile network," 2005.

[90] A. M. Cipriano, R. Visoz, and T. Salzer, "Calibration Issues of PHY Layer Abstractions for Wireless Broadband Systems," *2008 IEEE 68th Vehicular Technology Conference*, 2008.

[91] Z. Hanzaz and H. D. Schotten, "Performance evaluation of Link to system interface for Long Term Evolution system," *7th International Wireless Communications and Mobile Computing Conference*, 2011.

[92] X. Li, Q. Fang, and L. Shi, "A effective SINR link to system mapping method for CQI feedback in TD-LTE system," *IEEE 2nd International Conference on Computing, Control and Industrial Engineering*, 2011.

[93] J. C. Ikuno, "System Level Modeling and Optimization of the LTE Downlink," *Ph.D. dissertation, TU Wien*, 2013.

[94] ETSI, *EN 300 328: Wideband transmission systems; Data transmission equipment operating in the 2,4 GHz band; Harmonised Standard for access to radio spectrum.*, v2.2.1, 2019.

[95] 3GPP, *TR 38.912- 5G; Study on New Radio (NR) access technology.*, v16.0.0, Release 16, 2020.

[96] O. Özdogan, E. Björnson, and E. G. Larsson, "Intelligent Reflecting Surfaces: Physics, Propagation, and Pathloss Modeling," *IEEE Wireless Communications Letters*, 2020.

[97] Xingbin He, Zhi Zhang, and Wenjie Wang, "Doa estimation with uniform rectangular array in the presence of mutual coupling," *2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016.

[98] K. Brueninghaus, D. Astely, T. Salzer, S. Visuri, A. Alexiou, S. Karger, and G. . Seraji, "Link performance models for system level simulations of broadband radio access systems," *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2005.

[99] Wikipedia, "Wikipedia: Hyper-threading," https://en.wikipedia.org/wiki/Hyper-threading.

[100] IEEE, *IEEE 754-2019 - IEEE Standard for Floating-Point Arithmetic.*, Active Status, published on the 22nd of June, 2019.

A

Deriving a Beam Steering Vector

This appendix shows how to obtain a conventional beam steering vector, i.e. the vector of complex weights with unitary amplitudes and varying phases to apply to each antenna element such that the beam has the desired direction. The derivation starts assuming the reader is familiar with antenna theory, and is knowledgeable about the principles behind beam steering. For a complete introduction antenna theory and beam-steering, consult [11]. Additionally, we adapt the formulas with conventional references on the angles to better angular references that considerably facilitate computations with planar arrays.

Let us start by recalling that the radiation/antenna pattern of an antenna array is equal to its directivity scaled by the total radiation power. To facilitate comparisons, we will handle always the normalised version of the antenna pattern, i.e. its directivity. Although one is more used to hear about gains, remember that the gain is nothing more than the directivity multiplied to the antenna radiation efficiency. And they are equal if we consider a perfect radiator. Thus, let us consider the directivity to avoid assuming values that will not be used anywhere else, but note that thinking about gain or directivity makes no difference in the conclusions of this section.

The directivity of the array D_{array} for an uniform antenna array (same antenna elements, uniformly spaced) comes from the product of its current-normalised Array Factor AF_n with the directivity of each antenna element D_{ae}. Equation (A.1) summarises this fact.

$$D_{array}(\phi, \theta) = |AF_n(\phi, \theta)|^2 \times D_{ae}(\phi, \theta) \tag{A.1}$$

Furthermore, recall that conventional beam steering is nothing more than changing the array factor such that the resultant antenna pattern has a maximum along the intended direction. The array factor AF with uniform element excitation and a setting illustrated in Figure A.1 is given in Equation (A.2). Note that for the directivity we need the current-normalised version given by $AF_n = AF/I_0$.

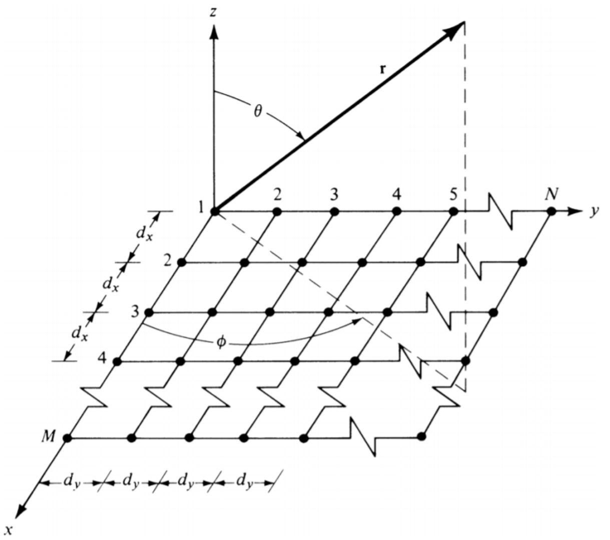


Figure A.1: Planar array geometry. From [11].

$$AF(\phi, \theta) = I_0 \sum_{m=1}^{M} e^{j(m-1)(kd_x \cos(\phi)\sin(\theta)+\beta_x)} \sum_{n=1}^{N} e^{j(n-1)(kd_x \sin(\phi)\sin(\theta)+\beta_x)} \tag{A.2}$$

To maximise the power in a given direction we just have to change the phases differences β_x and β_y such that the exponentials equate 1, for any element/index of the sum. One does so by having the exponent be 0, by making the progressive phase shifts β_x and β_y be the symmetric of the other term in the same brackets. Consequently, creating a beam to (ϕ_0, θ_0) implies having the progressive phase shifts like the left side of Equations (A.3a) and (A.3b).

$$\beta_x = -kd_x \cos(\phi_0)\sin(\theta_0) \tag{A.3a}$$

$$\beta_y = -kd_y \sin(\phi_0)\sin(\theta_0) \tag{A.3b}$$

Finally, to obtain the steering vector we follow a standard procedure, well explained in [97]. We define incremental phase steps, along the x-axis denoted by u_x and along the y-axis, denoted by u_y, and apply them to the antenna elements in given positions to coherently sum their contribution, thus having a weight per element w_{n_x,n_y} as shown in Equation (A.4), where n_x and n_y are the indices of the antenna element in the array.

Regarding inter-element spacing, half-wavelength is the most common distance between elements. Two of the main reasons are: i) half-wavelength is minimum antenna spacing for obtaining a fully formed the main lobe, and the maximum for not obtaining multiple copies of the said main lobe, called grating lobes; and ii) with distances that are even multiples of half-wavelength, the probability that there's a null at both antennas is much lower, enhancing diversity gain.

For the an inter-element spacing of half-wavelength in both orientations ($d_x = d_y = \lambda/2$), one may simplify the β_x and β_y as in Equations (A.5) and (A.6).

$$w_{m,n} = u_x^{n_x-1} u_y^{n_y-1},\ n_x = 1, \ldots, N_x, \\ n_y = 1, \ldots, N_y \tag{A.4}$$

$$u_x = e^{j\beta_x} = e^{-j\pi \sin(\phi_0) \sin(\theta_0)} \tag{A.5}$$

$$u_y = e^{j\beta_y} = e^{-j\pi \cos(\phi_0) \sin(\theta_0)} \tag{A.6}$$

And using (A.4), we obtain the beamforming matrix $\boldsymbol{W}$ in (A.7), that maximises the signal transmission/reception in (ϕ_{r0}, θ_{r0}) direction.

$$\boldsymbol{W} = \begin{bmatrix} 1 & \ldots & u_y^{(N_y-1)} \\ \vdots & \ddots & \vdots \\ u_x^{(N_x-1)} & & u_x^{(N_x-1)} u_y^{(N_y-1)} \end{bmatrix} \tag{A.7}$$

B

Bridge to Application QoE

Previously we defined a method to compute the size of an I-frame based on the average throughput. Here we relate application parameters such as resolutions and pixel colour depths to the computation of an I-frame. This way, we can say that if we support a throughput which corresponds to a given I-frame size, we also support any application layer settings that get the same frame size.

To compute the uncompressed I-frame size $S_{I,uncomp.}$ from application parameters we use (B.1) with the following parameters:

- Resolution or number of pixels per frame N_{pixels}^{frame} - e.g. 4K is 3840 horizontal pixels by 2160 pixel on the vertical;
- Pixel format or number of channels per pixel N_{ch}^{pixel} - channels refer to the components of that pixel, e.g. RGB has 3 channels, one for red, one for green and one for blue;
- Pixel depth or average number of bits per channel $\overline{N_{bits}^{ch}}$ - e.g. if each channel uses 8 bits to specify its value, the average will be 8 bits per channel, and in RGB this would mean 24 bits per pixel, from the multiplication with N_{ch}^{pixel};
- Depth Resolution D_{res} - in RGBD (RGB + depth) cameras there is an extra channel for depth and this parameter holds how many bits it involves.

$$S_{I,uncomp.} = N_{pixel}^{frame} \times \left(N_{ch}^{pixel} \times \overline{N_{bits}^{ch}} + D_{res} \right) \tag{B.1}$$

Now we need to bridge this uncompressed I-frame size to the actual I-frame size to be sent, thus pos-compression. Equation (B.2) shows how the two relate, and (B.3) shows how to compute the ratio between the two. The remaining compression ratio RCR is the ratio between compressed and . Therefore, if the encoder is expected to compress the stream to 1/300, and the compression from using IP frames is 3 times, then RCR is 1/100 because we do not want to compress the I frame. This step is necessary because the encoder is also responsible for generating I and P frames. Therefore, the total compression ratio C_T attributed to the encoder must be separated in the RCR and the ratio that we implicitly apply by considering I and P frames C_{IP}.

$$I_{size} = I_{\text{size}_{\text{uncomp.}}} \times \text{RCR} \tag{B.2}$$

$$RCR = \frac{C_T}{C_{IP}} \tag{B.3}$$

And to obtain C_{IP}, the compression from using I and P frames instead of only I frames, see Equation B.4.

$$
\begin{aligned}
C_{IP} &= \frac{\text{Avg. Frame Size after IP compression}}{\text{Avg. Frame Size before IP compression}} \\
&= \frac{S_I \frac{1+r_{P/I}(S_{GoP}-1)}{S_{GoP}}}{S_I} = \frac{1+r_{P/I}(S_{GoP}-1)}{S_{GoP}} \qquad \text{(B.4)}
\end{aligned}
$$

To summarise, besides the application parameters listed previously in the $S_{I,uncomp.}$ computation, we require an application total compression ratio C_T to compute S_I.

C

BLER Curves Fitted Equations

The curves plotted in Figure 3.16, Section 3.3, have been fitted to simulations by the currently PhD student Maria Raftopoulou in the Technical University of Delft, using the Vienna link-level simulator [10]. The equations for each MCS curve are below.

$$0.8942e^{-(x+10.05)/1.28^2} + 0.5795e^{-(x+8.602)/0.9784^2} \tag{C.1}$$

$$1 - \frac{9.182}{e^{-4.293x-16.31} + 9.171} \tag{C.2}$$

$$1 - \frac{0.7106}{e^{-6.388x+0.7106}} \tag{C.3}$$

$$1 - \frac{1}{e^{-6.138x+28.19} + 0.9996} \tag{C.4}$$

$$1 - \frac{1}{e^{-7.502x+44.68} + 0.9985} \tag{C.5}$$

$$1 - \frac{1}{e^{-8.279x+64.07} + 0.9996} \tag{C.6}$$

$$1 - \frac{1}{e^{-7.981x+79.61} + 0.9998} \tag{C.7}$$

$$1 - \frac{1}{e^{-8.217x+96.46} + 0.9995} \tag{C.8}$$

$$1 - \frac{1}{e^{-9.292x+124.6} + 0.9989} \tag{C.9}$$

$$0.6046e^{-(x-15.1)/0.3454^2} + 0.9940e^{-(x-13.47)/0.969^2} + 0.6544e^{-(x-14.56)/0.5685^2} \tag{C.10}$$

$$\begin{aligned} &0.6768e^{-(x-16.3)/0.6109^2} + 0.9575e^{-(x-15.24)/0.895^2} + 0.5245e^{-(x-17.08)/0.2716^2} + \\ &\qquad + 0.4280e^{-(x-16.73)/0.3344^2} \end{aligned} \tag{C.11}$$

$$1 - \frac{1}{e^{-10.22x+196.7} + 0.9999} \tag{C.12}$$

$$1 - \frac{1}{e^{-9.939x+208.5} + 0.9988} \tag{C.13}$$

$$1 - \frac{1}{e^{-10.23x+234.7} + 0.9995} \tag{C.14}$$

$$1 - \frac{1}{e^{-9.504x+235.7} + 0.9997} \tag{C.15}$$

D

Mutual Information Effective SINR Mapping

Disclaimer: practically all content in this appendix was not authored by the author of this thesis. Instead, it was taken from a non-public report on ‘Radio channel modelling for LTE system-level simulations’ authored by Onno Mantel, Yohan Toh, Manolis Chrysallos and Remco Litjens, from TNO, a Dutch research company.

Physical-layer abstraction models are an important building block of system-level simulators for mobile networks. In wideband systems, the SINR varies over the system bandwidth and the link is typically characterised by a set of SINR values for the different PRBs in the band. This is schematically illustrated in Figure 2-1. Physical-layer abstraction models translate the set of PRB-specific SINR values into one effective SINR, denoted $SINR_{eff}$, over the assigned bandwidth. This process is also referred to as Effective SINR Mapping (ESM). Different ESM functions have been proposed in the literature [98].

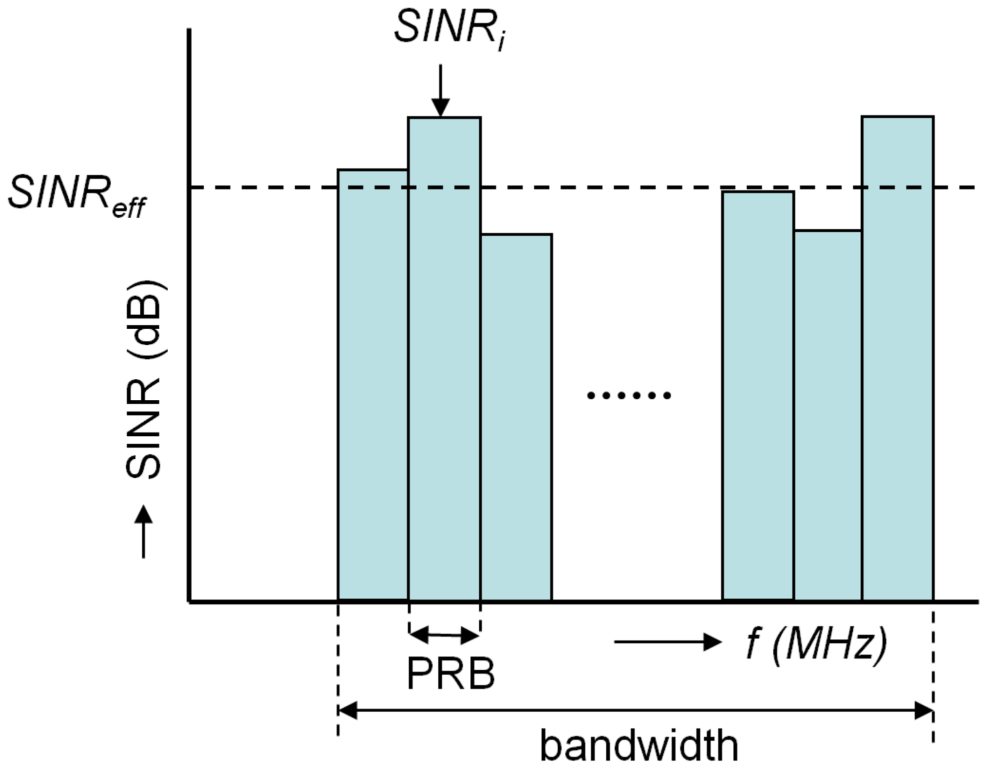


Figure D.1: Effective SINR mapping result.

The effective SINR mapping using MI-ESM is given by:

$$SINR_{eff} = I_k^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} I_k\left(SINR_i\right)\right) \tag{D.1}$$

where $SINR_i$ is the SINR of PRB i and N is the number of assigned PRBs in the considered bandwidth. I_k is the Bit-Interleaved Coded Modulation (BICM) capacity for the considered modulation order k. It is given by:

$$I_k(SINR_i) = k - \mathbb{E}\left\{ \frac{1}{2^k} \sum_{p=1}^{k} \sum_{b=0}^{1} \sum_{z \in \chi_b^p} \log_2 \frac{\sum_{\hat{x} \in \chi} e^{-\left| Y - \sqrt{SINR_i/\beta_k}(\hat{x}-z) \right|^2}}{\sum_{\hat{x} \in \chi_b^p} e^{-\left| Y - \sqrt{SINR_i/\beta_k}(\hat{x}-z) \right|^2}} \right\} \tag{D.2}$$

where:

- χ is the set of $2k$ constellation symbols for the modulation order k. For instance, for modulation order $k = 2$ the corresponding constellation symbols are $1/\sqrt{2}(-1-j)$, $1/\sqrt{2}(-1+j)$, $1/\sqrt{2}(1-j)$, and $1/\sqrt{2}(1+j)$.

- χ_b^p is the subset of constellation symbols for which bit b equals p. For instance, χ_{-1}^1 is is $\{1/\sqrt{2}(-1-j), 1/\sqrt{2}(-1+j)\}$. Note that b is the (zero-indexed) index of the bit, and p is the value of the bit.

- Y is a complex stochastic variable which is normally distributed with zero mean and unit variance.

- β_k is a calibration parameter which is MCS-dependent. The parameter is not always included in the equation (see e.g. [93]), suggesting that is also possible to apply MI-ESM without calibration parameter.

The quantity I_k is also referred to as the 'mutual information' and is commonly used in information theory to describe the relationship between two variables that are sampled simultaneously. In particular, it tells how much information is communicated in one variable about the other. In the context of MI-ESM, this should be seen as to what extent a received symbol provides information about the transmitted symbol. For high SINR (ideal channel) the mutual information is maximum and equal to the number of bits per symbol. For low SINR (no data transfer possible) the mutual information reduces to zero, i.e., the received symbol provides no information about the transmitted symbol.

Figure D.2 shows the BICM capacity I_k for the three modulation orders k ($k = 2$ for QPSK, $k = 4$ for 16QAM, $k = 6$ for 64QAM) used in LTE. This figure should be read as follows:

- For low SINR (left side of the plot), the mutual information is zero. The received bit sequence per symbol is statistically independent of the transmitted bit sequence, and thus no information is sent over the link.

- For high SINR (right side of the plot), the mutual information is equal to the number of bits per symbol - 2 in the case of QPSK, 4 in the case of 16QAM, and 6 in the case of 64QAM. The received bit sequence is identical to the transmitted sequence.

- For intermediate SINR, the mutual information value lies somewhere in between these two extreme values. Since 64QAM applies more bits per symbol than 16QAM (or QPSK), it is more vulnerable to channel distortions. Therefore a higher SINR is required to achieve the maximum mutual information value.

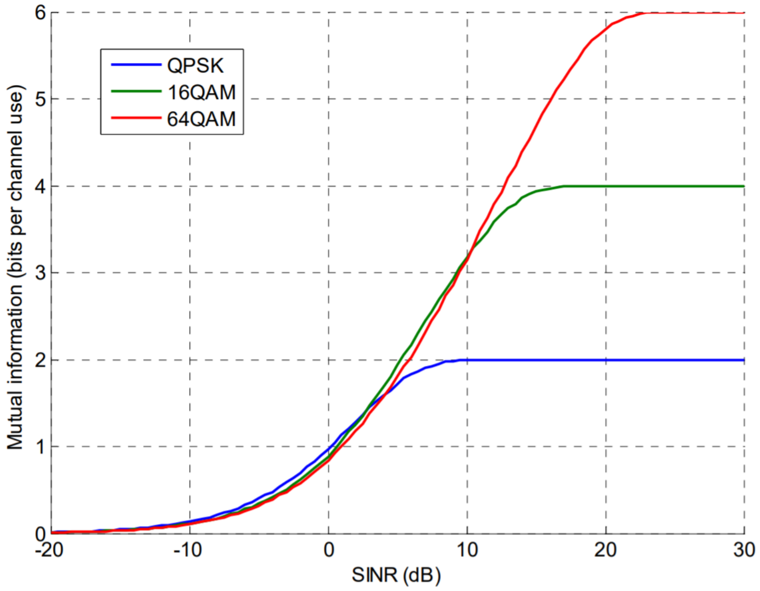


Figure D.2: Mutual Information versus SINR for the three modulations orders in LTE.

Thus, we understand that MI-ESM works as follows:

- Per PRB the mutual information is computed, depending on the SINR for that PRB. For low SINR, the mutual information is zero, meaning that no useful information can be extracted for that PRB. For high SINR, the mutual information is equal to the number of bits per symbol for the given modulation order. This means that all the information in the symbol can be successfully transferred.
- Then the average is determined of these mutual information values per PRB.
- Subsequently, the effective SINR is determined as the SINR that would yield this average mutual information if it were applied on all PRBs.

E

mmWave Simulation Plots

All plots in this appendix refer to mmWaves.

The throughput is higher, which is expected if the SINRs are higher as a result of more directive transmissions (more gain, less interference).

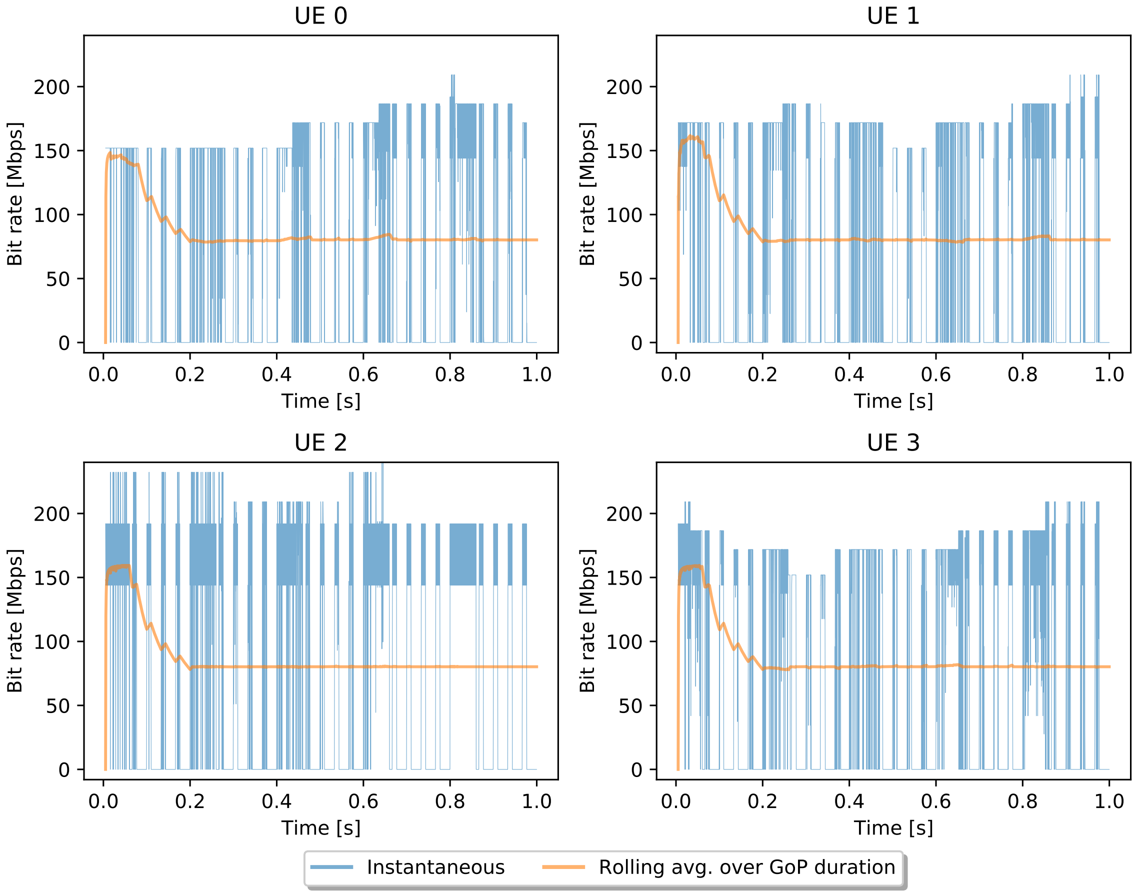


Figure E.1: UE bit rates, instantaneous and averaged over the last 200 ms (GoP duration).

The graphs of the SINR, in Figure E.2, and the signal and interference powers, in Figure E.3, are consistent. However, UE 0 and UE 2 have different shapes than the same UEs in lower frequency, while UE 1 and UE 3 have the same. This may be due to different beam shapes.

The MCSs in Figure E.4, much like the SINRs, are consistently higher in mmWaves than in lower frequencies.

The BLERs in Figure E.5 are very similar to lower frequencies. Figure E.6 that shows the OLLA parameter correlated with the BLER is also consistent.

Figure E.7 shows a considerable better performance with respect to latency than 3.5 GHz. More concretely, now 3 out of 4 users have 0% drop rate. This makes sense since all SINRs are higher. Given a better channel quality, the performance in terms of throughput and latency expectedly improves.

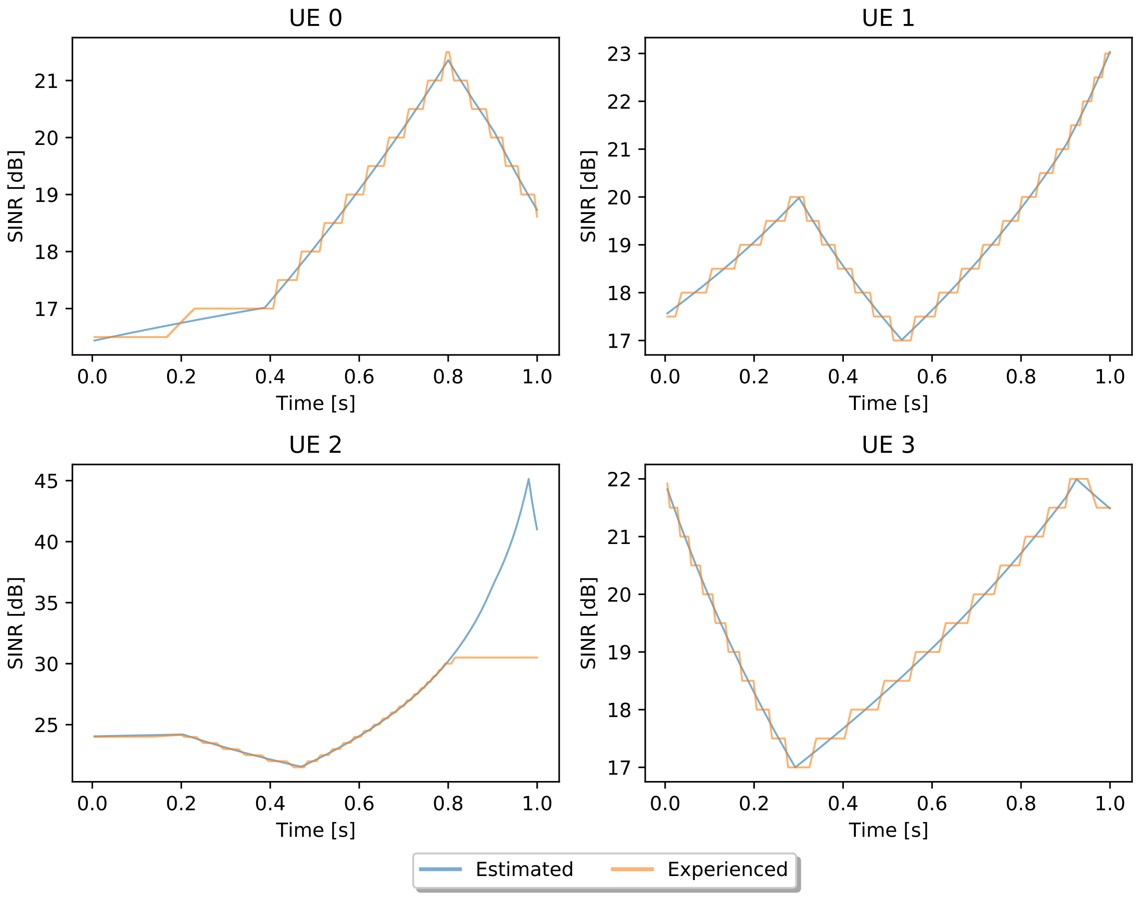


Figure E.2: SINR, estimated before transmission and experienced during transmission.

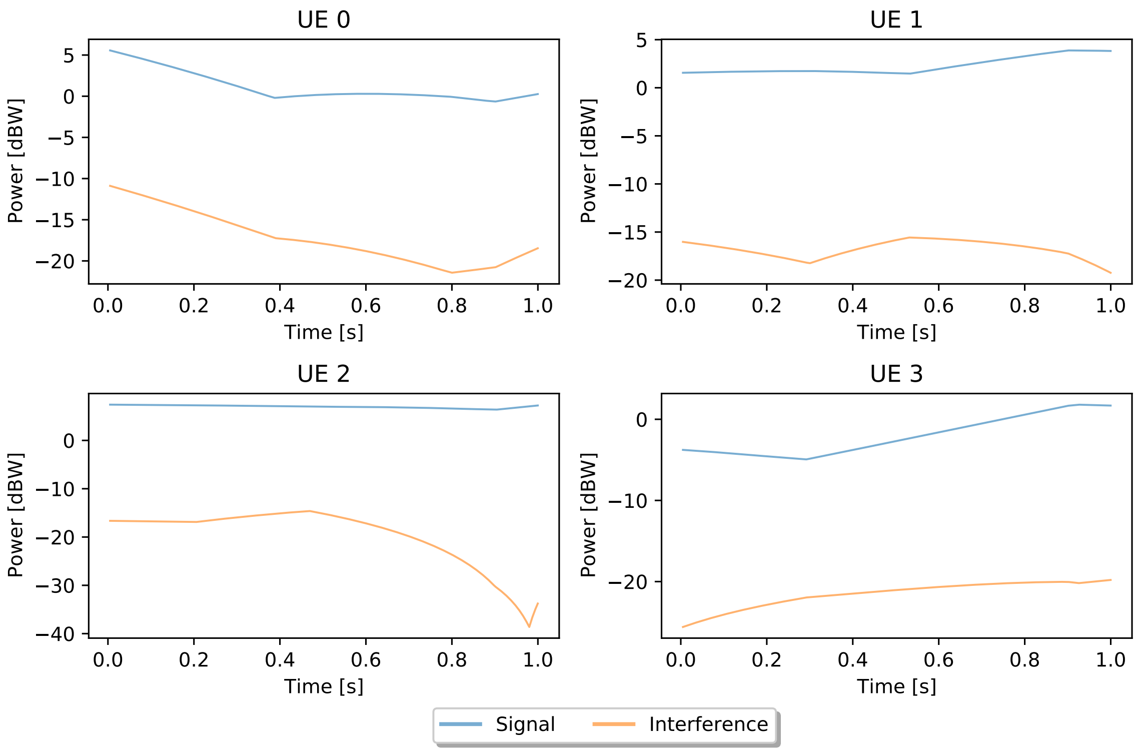


Figure E.3: Received Signal and Interference Powers.

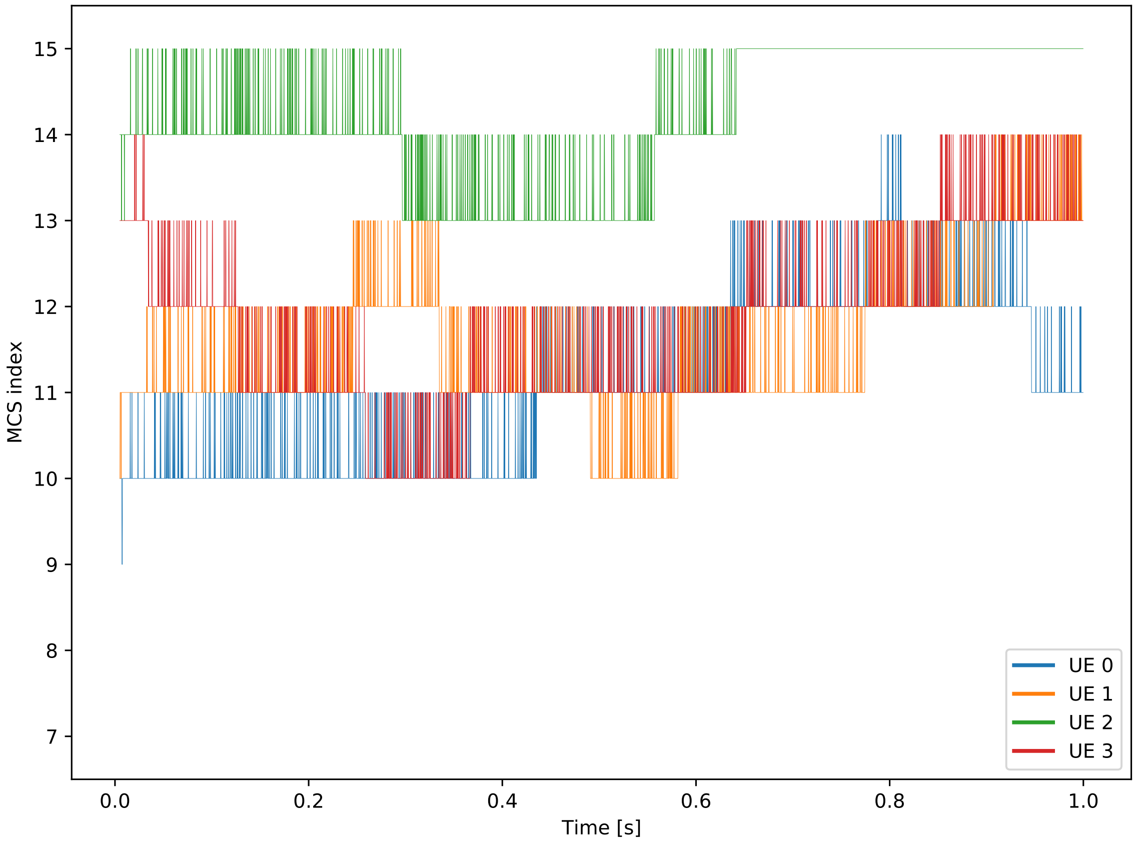


Figure E.4: MCS index used by each UE in every transmission.

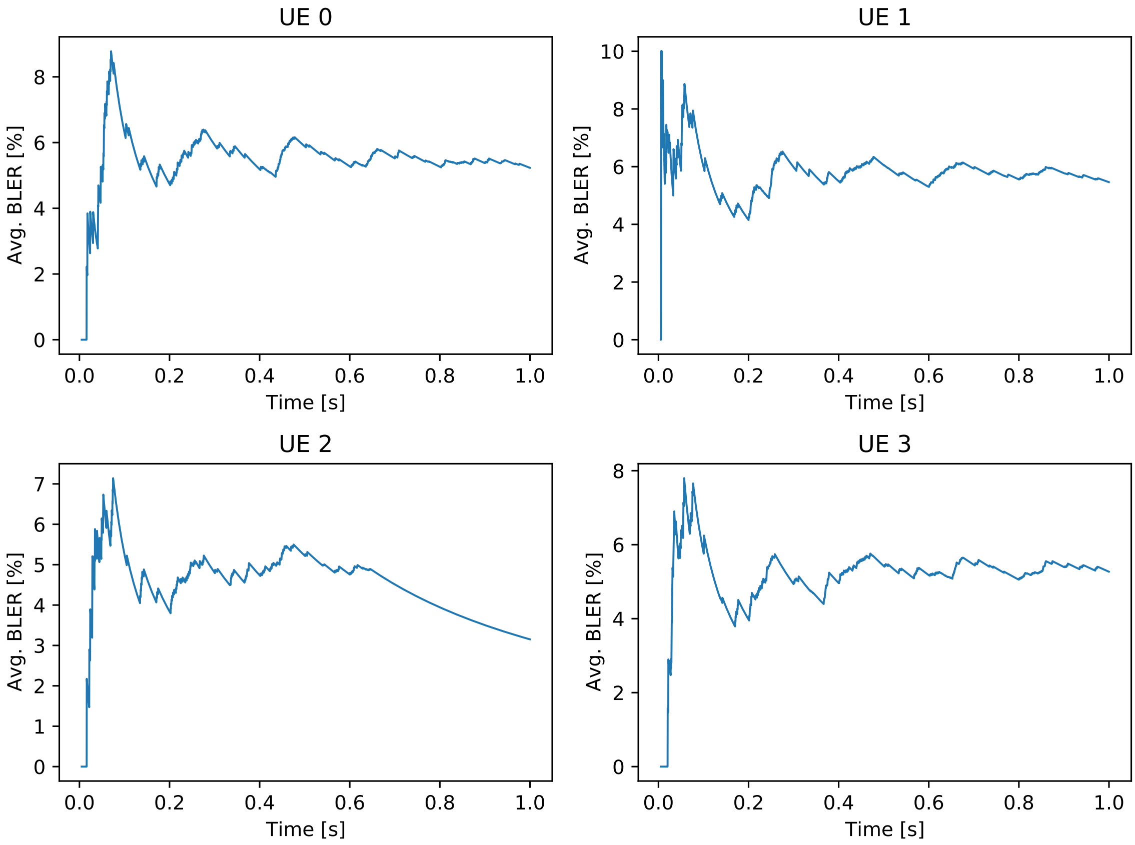


Figure E.5: All time BLER average in a multi-user scenario.

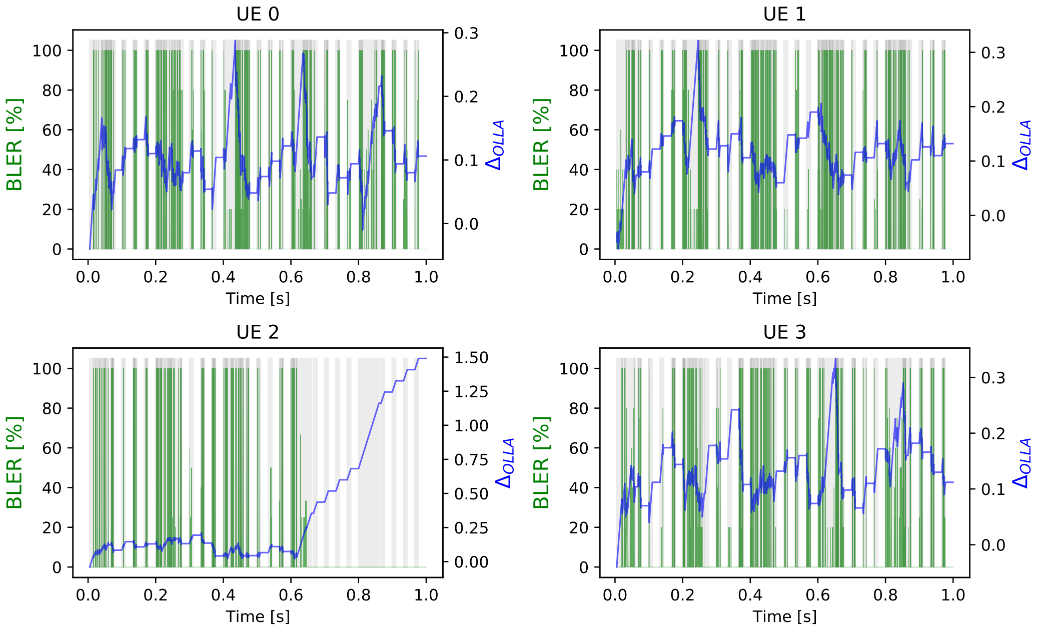


Figure E.6: Link adaptation parameter variation with the instantaneous BLER. Grey zones mark when the UE has active transmissions.

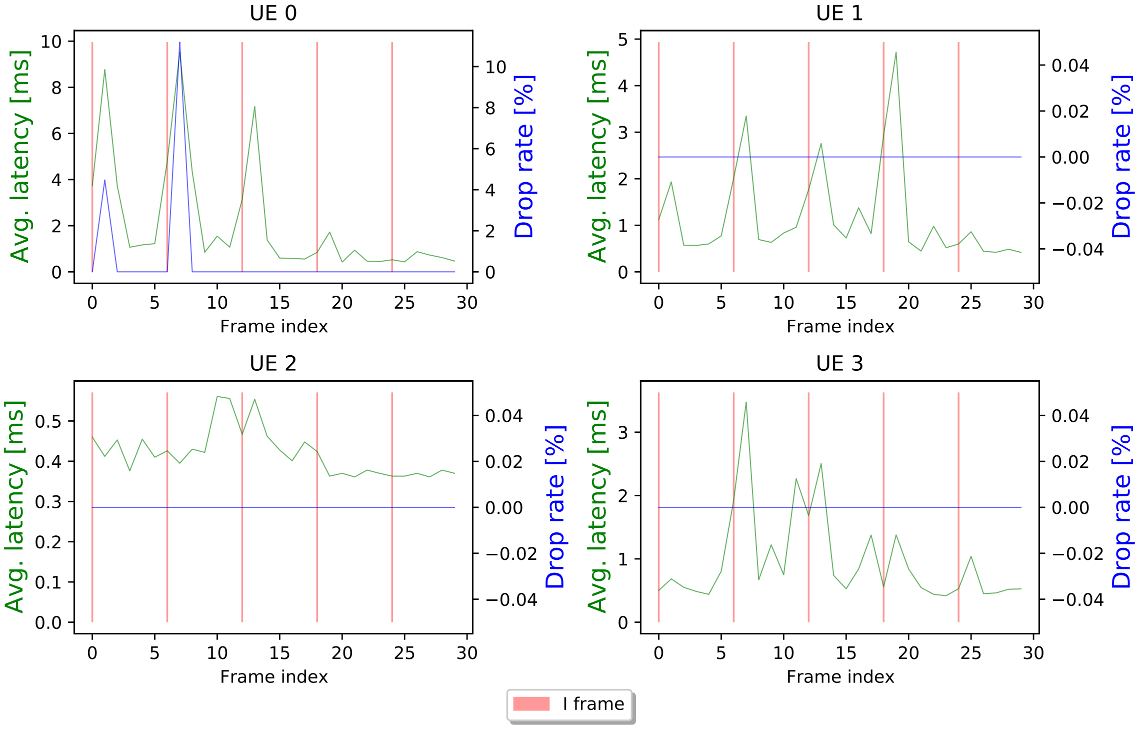


Figure E.7: Packet latencies and drop-rates for all UEs with I-frame marking.

F

On the Implementation of a System-level Simulator

Contents

F.1 Overview

This appendix concerns implementation aspects. As expected, there are many more implementation parameters than modelling parameters. For instance, we can model a set of frequencies $\mathcal{F}$ and a bandwidth B, but it is quite reasonable to use different bandwidths for different frequency bands. Likewise, we can use different antennas across users, place users around rectangular instead of circular tables, or just in non-standard places, use hybrid architectures for each antenna, have users with different heights, put a UE in each camera instead of aggregating them, among plenty of other things. The total number of setting variables exceeds 200 while the modelling parameters presented in this thesis are around 50.

When organised and properly filtered to the essential files, the simulator codebase should consist of around 12000 lines. As of now, the simulator counts with more than 18000 lines of code, taking into account empty lines, comments, plots, auxiliary scripts, and all Matlab and Python functions.

In essence, the simulator is made of two phases, channel generation and simulation. We provide an overview on each phase, how they interact and then we carefully address the generation process, since it has represented a major engineering challenge in the course of this work. Additionally, with the modelling of the application, in Section 3.1, and radio access network, in Section 3.3, it should be simple to understand and modify the heavily commented code on the simulation phase. The channel generation process is complex and was not addressed before, hence doing it here.

Everything in the simulation phase is programmed in Python 3, and runs perfectly in Python version up to and including 3.8.7. In the channel coefficient generation phase is programmed in Matlab and compiled to an executable which is then parallelised with Python. The Matlab script does not require external toolboxes or packages to be compiled, only Quadriga, version 2.4. Python versions are very compatible between themselves, but that is less true with Quadriga so it may require adjustments for subsequent versions as it was the case when upgrading from 2.2 to 2.4.

As mentioned, the executable is parallelised in Python, and we use an external library for that. All libraries used in Python:

- NumPy for optimised mathematics and multi-dimensional arrays management;
- JobLib distributes tasks among workers in different virtual cores, allowing to run in different subprocesses/threads the channel generation workload;
- Pathlib to facilitate directory management;
- IO module from SciPy library to save/load matlab files;

- Datetime for obtaining current time and managing time instants;
- OS for getting the directory where the Python script has been executed and create directories for new simulations and assessing system specifications like the number of available cores;
- SYS for input argument parsing;
- Matplotlib for plotting data;
- Pickle to save and load Python's environment variables, useful to save simulations and interface them with the data analysis scripts.

We first present overviews for both the simulation and generation phase, and then we address the generation phase carefully. We start with the simulation phase, to contextualise the use of the channel coefficients.

Simulation Overview

The simulation phase integrates the application traffic and the channel coefficients in a process that returns performance results and a plethora of other useful metrics that allow us to dissect radio access network behaviour.

The core of the system-level simulator has had its modelling detailed before so we will not present implementation details here. Regarding the application, there are two places where the system level simulator interacts with the application traffic.

The first is at the beginning of the TTI, where the queue times are updated. This update consists of adding the packets that came in between the previous and the current TTI, updating the head of queue delay, which may be needed for making scheduling decisions,, and discarding packets that will not be served within the radio latency budget. When the head of queue delay plus the average time a transport block takes to be sent exceeds the latency budget, the packet at the front of the buffer is discarded. The head of queue delay is then updated for the next packet which is discarded as long as the condition verifies.

The second is at the end of the transmission where the bits carried by the successful transport blocks are removed from the packets they belong to. If a transport block has errors, no modification to the buffers is necessary .

See in Figure F.1 a flowchart that provides a short summary on how channel coefficients, application traffic and the core of the system-level simulator interact. Basically, application traffic, channel coefficient and radio access parameters are inputs for the core of the simulator.

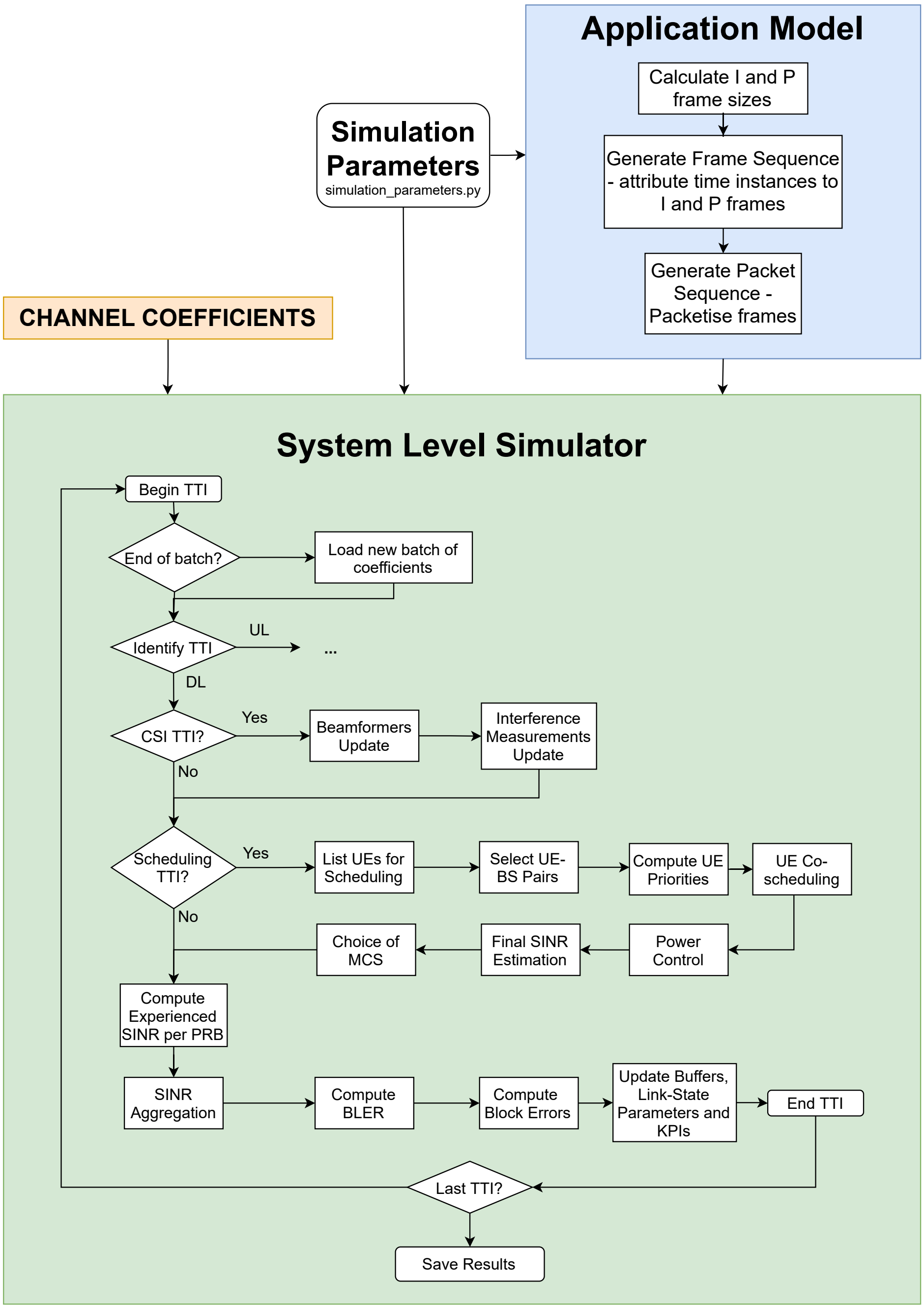


Figure F.1: Simulation phase overview flowchart.

Channel Generation Overview

The channel generation component has to do with generating the channel impulse responses or coefficients. As mentioned previously, such coefficients describe our propagation environment. Each coefficient is a complex number and can represent a time or frequency response, depending on the domain we generate our channel. Using Quadriga it is only possible to compute time-domain responses and then convert them to frequency. This is relevant because we need frequency-domain coefficients. not only for the required steps to obtain

To justify why generating coefficients has represented such a challenge we need to address how many coefficients a casual simulation needs and how long they take to compute. Quadriga separates all BSs in a group called TXs and all UEs in a group called RXs, therefore, for now on, we will address a BS as transmitter, although it is capable of receiving as well, and a UE as a receiver.

For each combination of transmitters and receivers, in each frequency band, there is one four-dimensional (4D) tensor of coefficients. In time domain, we call that tensor $\boldsymbol{C}$ and $\boldsymbol{C} \in \mathbb{C}^{N_{ant}^{UE} \times N_{ant}^{BS} \times N_{path} \times N_{TTI}}$. In frequency domain, that tensor has slightly different dimensions and we call it $\boldsymbol{H}$, with $\boldsymbol{H} \in \mathbb{C}^{N_{ant}^{UE} \times N_{ant}^{BS} \times N_{PRB} \times N_{TTI}}$, where N_{ant}^{UE} is the number of single-polarised antenna elements per UE.

We can compute N_{TTI} from Equation (F.1). Furthermore, defining N_{domain} as the only parameter that in the tensor dimensions between time and frequency domains, according to Equation (F.2), then we can compute the total number of coefficients using Equation (F.4).

$$N_{TTI} = 1000 \times 2^{\mu} \times T_{sim} \tag{F.1}$$

$$N_{domain=time} = N_{path} \tag{F.2}$$

$$N_{domain=freq} = N_{PRB} \tag{F.3}$$

$$N_{coeff} = N_{UE} \cdot N_{BS} \cdot N_{FREQ} \cdot N_{ant}^{UE} \cdot N_{ant}^{BS} \cdot N_{domain} \cdot N_{TTI} \tag{F.4}$$

Let us see quite a conservative example to show the computation challenges. For a single frequency, we take the following values:

- $N_{UE} = 4$
- $N_{BS} = 1$
- $N_{ant}^{UE} = 8$
- $N_{ant}^{BS} = 64$
- $T_{sim} = 20$ s
- $\mu = 2$
- $N_{path} = 16$
- $N_{PRB} = 100$

The values above result in roughly 2.6×10^9 coefficients in the time domain and 16.4×10^9 coefficients in the frequency domain, for the given frequency band. We use single-precision floating-point format to save time and memory - see Section F.3 for why single-precision is sufficient. Therefore, with 4 bytes per real number, each complex number takes up 8 bytes. For the values above, together the time and frequency coefficients would require around 160 GB.

Note that we intend to simulate up to 8 UEs, up to 5 BSs, 64 antennas per UE, 512 antennas per BS and 2 frequencies, which represents a multiplier of 1280, or 200

TB worth of data. Such numbers are not for this work, although we have discovered methods of reducing the required data for more than hundredfold. The simulations presented in the results Section 4 required 800 GB.

Regarding computation time, it takes around 4 minutes to generate 1 GB worth of coefficients in an Intel Core i5-7300U @ 2.60 GHz, which has a performance per virtual core on par with the average server-grade CPU core that is not running at very high frequencies due to heat dissipation issues and hardware longevity. Therefore, slightly more than 2 days per Terabyte.

The number of instances we are able to execute in parallel is how much faster we can compute coefficients. Therefore, with 12 instances, a 48-hour TB comes down to 4 hours, assuming we do not run into bottlenecks such as slow writing speeds. Of course, what is fast and slow is relative and depends on our parallelisation settings. That is why we first present what steps are required for channel generation, and then the parallelisation engine and its settings.

Not only regarding time, but memory wise, parallelisation also allows us to leverage multiple physically separated storages, without having to setup a transmission mechanism between them. This means the memory problems can be more easily dealt with. Finally, still in the memory department, Matlab has a limit matrix size for efficient handling, which is 75 GB in most systems although it is system-specific. Therefore, some segmentation is necessary to guarantee the computation time does not scale further.

F.2 Channel Generation

We have shown how the problem can represent prohibitive time and memory requirements. In this work we did not use substantial memory reduction methods, but the parallelisation is a significant way of reducing the time needed. The point of this section is to expose the main parts of the channel generation process, including the parallelisation engine and the settings used to get a setup an efficient channel generation.

Let us first see the structure of the generation workflow.

F.2.1 Structure of Generation workflow

Figure F.2 shows a flowchart of the steps implemented to obtain channel coefficients. Yellow boxes mean the steps are executed in Matlab, namely in a compiled MATLAB executable. Green boxes symbolise when the channel generator Quadriga is involved. Orange boxes limit major generation phases.

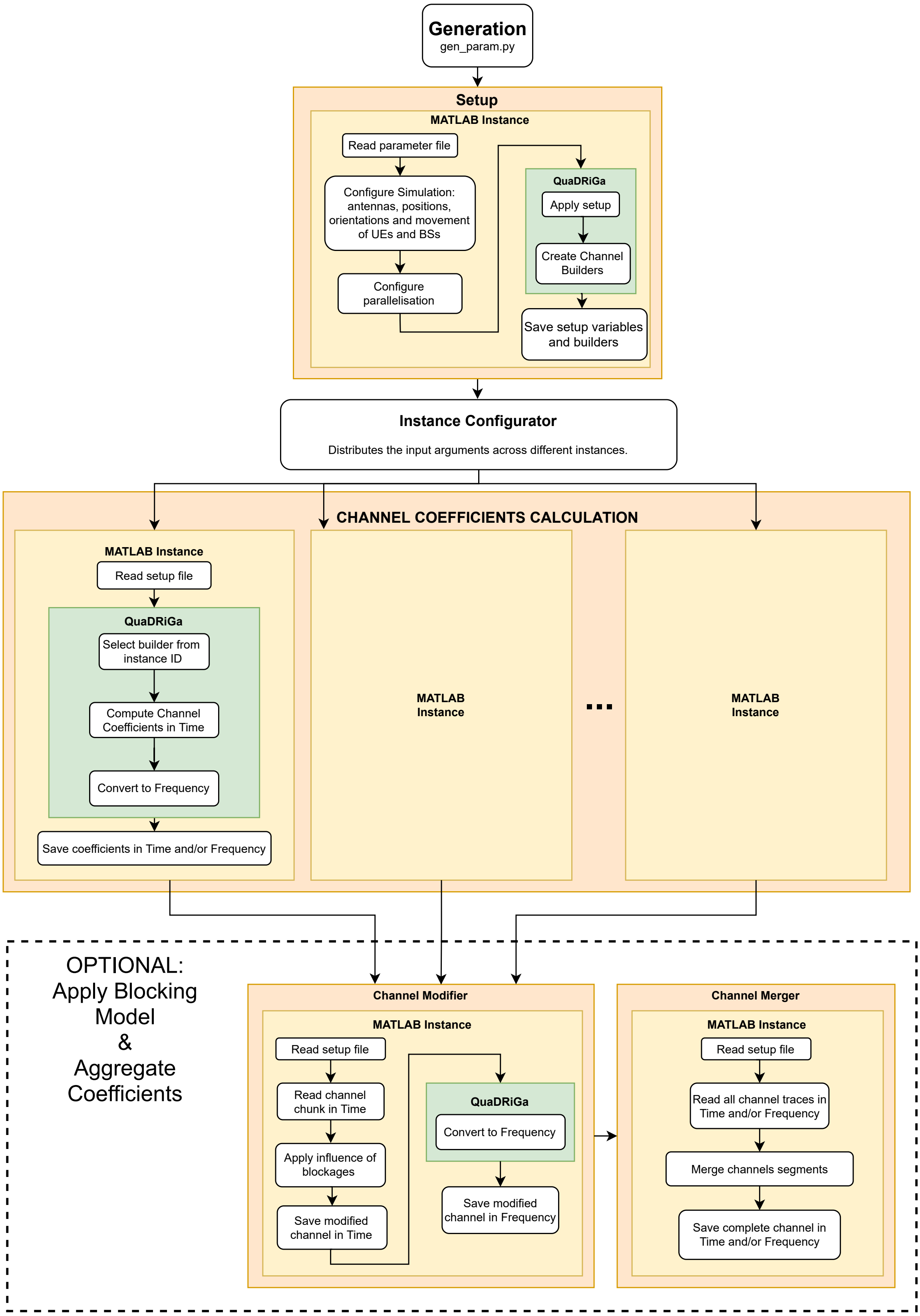

Generation
gen_param.py
Setup
MATLAB Instance
Read parameter file
Configure Simulation: antennas, positions, orientations and movement of UEs and BSs
Configure parallelisation
QuaDRiGa
Apply setup
Create Channel Builders
Save setup variables and builders
Instance Configurator
Distributes the input arguments across different instances.
CHANNEL COEFFICIENTS CALCULATION
MATLAB Instance
Read setup file
QuaDRiGa
Select builder from instance ID
Compute Channel Coefficients in Time
Convert to Frequency
Save coefficients in Time and/or Frequency
MATLAB Instance
...
MATLAB Instance
OPTIONAL: Apply Blocking Model & Aggregate Coefficients
Channel Modifier
MATLAB Instance
Read setup file
Read channel chunk in Time
Apply influence of blockages
Save modified channel in Time
QuaDRiGa
Convert to Frequency
Save modified channel in Frequency
Channel Merger
MATLAB Instance
Read setup file
Read all channel traces in Time and/or Frequency
Merge channels segments
Save complete channel in Time and/or Frequency

Figure F.2: Generation phase flowchart.

Before addressing the generation phases, we need to distinguish between builders and instances: we call builder to a Quadriga channel builder object and we use it with Quadriga functions to generate the channel coefficients. An instance is a physical program that runs on a virtual/logical core. More specifically, the program of an instance is the compiled MATLAB executable. There's two logical cores per physical core since Hyper-Threading was first introduced by Intel [99]. An instance can be responsible for building channel coefficients with one or more builders. The number of builders per instance is configurable to some extent, as we see next.

The generation phases are:

1. Setup - All variables for the whole simulations are set ahead of time and saved, to guarantee every instance in parallel is in conformity with a simulation that would not be parallelised. Also in this step, the builders are created and saved. The builder is the set of parameters required to generate a set of channel coefficients. Each builder represents a fraction of the workload required to generate the complete channel trace. This process is sequential and happens in a single instance.

2. Calculation - Load the variables from the setup and the set of builders assigned to that instance. Many instances can be executed simultaneously. Each instance uses its builders to generate the respective channel coefficients. These coefficients are saved in chunks, to be loaded as needed.

3. Modification (optional) - Modifies the channel in time domain, e.g. to study how the system would respond to certain conditions like human blockages. The time-domain channel coefficients are changed at the time instants where blockages happen. We need an additional step for it because we need MATLAB to convert these time-domain coefficients with information on each propagation path to frequency-domain.

4. Aggregation/Merge (optional) - In some cases it can be more useful to have only a single trace instead of many separated traces. This step allows glueing/merging all traces into a single file.

The same executable is used for all phases. The executable takes an argument called *flow_control*. The argument tells the executable which components should run. This, however, is not required to utilise the simulator. And will be left aside.

Each builder generates the channel between a BS and UE at a given frequency. Since the number of antenna elements normally changes with frequency, not all builders compute the same amount of data. For this reason, we use a soft batching parallelisation strategy where a new instance is started right when the previous finishes, and thus there are always as many instances running as supported.

The number of builders per instance, the number of instances, the total workload attributed to a builder and others, are important aspects to take into account when setting up a generation. Next we present exactly the variables used to control the parallelisation.

F.2.2 Parallelisation Settings

We aim to minimise the time it takes to execute the generation T_{total}. Doing so consists on controlling the average time each batch of instances running in parallel takes to run $\overline{T}_{batch}$ and how many batches need to be executed sequentially $N_{batches}$.

$$T_{total} = \overline{T}_{batch} \times N_{batches} \tag{F.5}$$

To minimise the time per channel generation is to maximise the utilisation of a pool of resources (RAM and CPU). For that, we use three parameters:

- **Number of Time Divisions** : $N_{time\ divs}$ dictates how many segments the workload has, by further dividing each segment in time. Each segment needs a builder, so the more time divisions, the more builders there are to run, but each builder takes less time and memory resources;
- **Number of Builders per Instance** : $N_{builders}^{instance}$ commands how many individual builders each instance executes. More builders make the instance run slower, but less instances are required.
- **Instances running in parallel** : $N_{inst\ parallel}$ is, as the name suggests, how many instances run in parallel. The total amount of instances in parallel depends on the size of each instance, which in turn depends on the number of builders per instance $N_{builders}^{instance}$ and the size of each builder (indirectly related to $N_{time\ divs}$). The heavier the instances, the less instances can run simultaneously, and the longer each batch of instances takes to finish, i.e. $\overline{T}_{batch}$ is larger. When instances are lighter, more of them can run simultaneously, but the number of sequential runs, or the number of batches $N_{batches}$ required is going to be more.

Let us demystify the parameters and conclude on how they can be used to tune generations.

Number of Time Divisions

The complete workload is divided across builders. A builder computes the coefficients for a BS-UE pair in a certain frequency band, and for a specific period of time. Additionally, each builder has an index which makes possible to address a specific portion of the workload with ease, both for generation or reading.

Normally a builder is assigned to the whole duration of the generation scenario. However, the problem with the required memory per instance led to further divisions of the workload. Therefore, we divide the workload across time, in addition to space and frequency. This means each builder computes a couple of seconds of the simulation instead of the full duration. The number of time divisions $N_{time\ divs}$ controls how much resources each builder will require, given by their height in Figure F.3. The more time divisions, the smaller is the height of each builder, which means less coefficients to generate per builder, but more builders.

Figure F.3 shows builders represented as blue squares. It is assumed 2 frequencies, 3 BSs (TXs) and 4 UEs (RXs) [1].

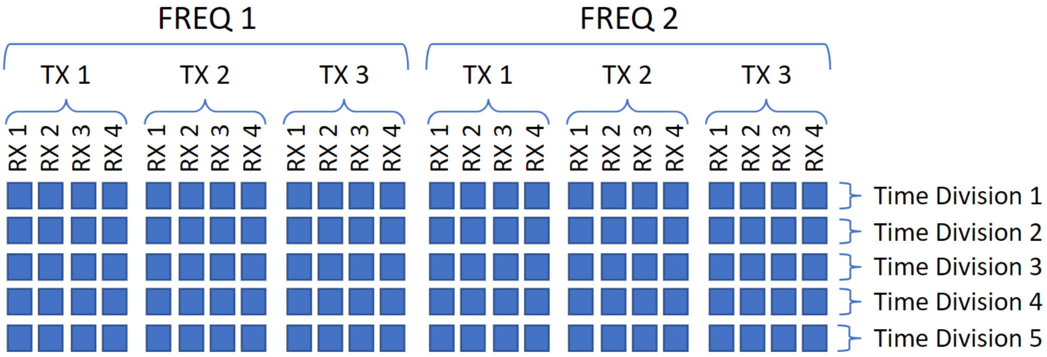


Figure F.3: Division of the workload in time, frequency and space.

See how the number of builders per time division $N_{builders}^{time\ div}$ is fixed to $N_{RX} \times N_{TX} \times N_{FREQ}$, where $N_{RX} = N_{UE}$ and $N_{TX} = N_{BS}$. Therefore, the total number of builders $N_{builders}$ depends on the $N_{time\ divs}$ as shown in Equation (F.6).

$$N_{builders} = N_{time\ divs} \times N_{builders\ per\ time\ div} \tag{F.6}$$

Number of Builders per Instance

We execute instances, not builders, so we need to know how many builders to use per instance $N_{builders}^{instance}$. Figure F.4 shows the four possible configurations to split builders in a time division across instances. In the figure, instances represented

[1] Quadriga calls TX and RX to separate the groups, however both transmit and receive.

as the boxes with a blue outline containing one or more builders. The options range from no further split, where $N^{instance}_{builders} = N^{time\ div}_{builders}$, to a single builder per instance.

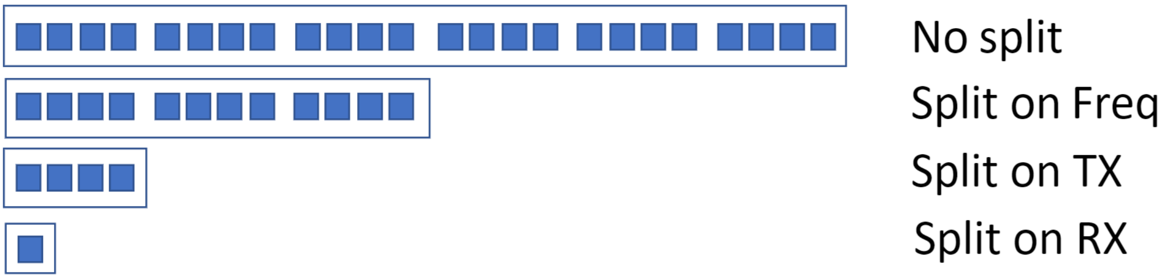


Figure F.4: Number of builders per instance.

$$N_{instances} = \frac{N_{builders}}{N^{instance}_{builders}} \tag{F.7}$$

Number of Instances in Parallel

The setting of this parameter should be the maximum possible, although it heavily depends on the requirements of each instance and the available resources. Normally, given the system available RAM (which tends to be the limiting factor), and the RAM each instance takes, this parameter gets set practically automatically. Expression (F.8) shows how it relates with $N_{batches}$.

$$N_{batches} = \frac{N_{instances}}{N_{inst\ parallel}} \tag{F.8}$$

Choice of Parallelisation Settings

The best way to proceed is: **trial and error.**

As a reference, for 800 GB simulations, with 64 time divisions, one builder per instance (parallelisation at the RX level), the heavier instances take 6 GB of RAM.

There simply are too many factors involved in choosing the perfect parameters and it depends on the machine.

Execution on Multiple Machines

Unfortunately, there are ways of making the parameter choice more complicated. Namely, using the resources of several machines instead of just one. We only make some considerations here, there are automated scripts for helping setting up parameter in these cases.

This parallelisation engine supports easy deployment for simultaneous computations on multiple machines. The only requirement for execution in a multi-machine setting is builder consistency. As long as the builders are transferred or generated again in the new machine with the same settings, it works.

If RAM is the limiting factor, then several time divisions help making less demanding builders. Moreover, if the generation happens in multiple machines simultaneously, it is also a strategy to have as sufficient parallel instances to have as many as possible running simultaneously. Also, the three parameters can be used to make the number of instances divisible by the number of machines such that each machine has assigned the same load making it likely they finish simultaneously. In case the machines have asymmetrical computing resources, the instances running in each machine can be adjusted to run more or less instances per machine.

Parallelisation Drawback

The only purpose of this section is to illustrate the only drawback of this approach, the loss of repeatability with when doing time segmentation. More concretely, repeating a simulation with the same parameters gives always the same result, but a simulation where only the number of time divisions changes should have identical results as previously but a slight change happens. Figure F.5 illustrates a zoomed-in vision of the differences. The left half is slightly lower, the top half is slightly higher, and there's an abrupt transition between the two.

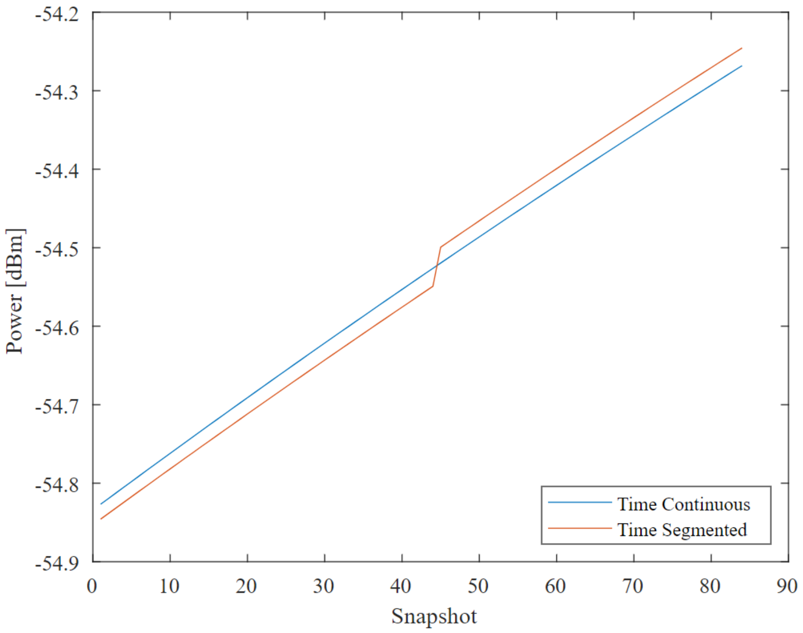


Figure F.5: Discontinuity of time segmented versus time continuous simulations.

In F.5 each snapshot is a quarter of a millisecond long, so indeed the changes are minuscule and can be ignored. Their severity increases with the time division length, but in realistic simulations there is no noticeable difference between an uninterrupted channel and a channel segmented in time and then stitched. Figure F.6 shows 2 seconds of a more realistic simulation. The stitching happens at 1 second mark, showing how imperceptible the transition is in the big picture. As said, all values drift a very small percentage to the ones generated without the parallelisation technique of time division, but we show the drawback only affects repeatability, completely unnoticeably.

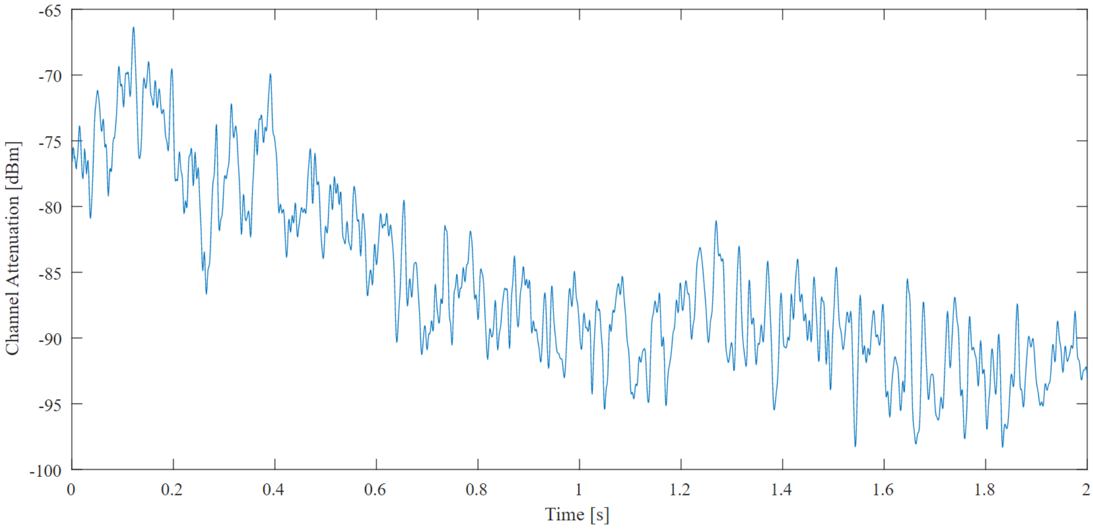


Figure F.6: Time segmented simulation with stiching at 1 second mark.

Qualitatively speaking, choosing of different random generator seeds represents a channel considerably different, so much so that the difference it's easily identified visually. Hence, one can safely make the case that the stitched channel is well within the range of possible values for a channel response. And, to the best of our knowledge, this is a bug internal to Quadriga, and it has been reported.

F.3 Other Optimisation Strategies

Since the bottleneck of our workflow currently resides in the channel generation process, we can optimise our workflow by reducing the number of channel generations we need when we want to change certain parameters. Another way is evidently by reducing the time and/or memory each generation takes.

We start with two techniques to reduce the number of required generations, subsetting and derivation. Then we see two techniques to reduce the memory requirements, changing the floating-point precision and extrapolate PRBs (tentative).

Subsetting

We can take subsets of information from a bigger channel generation and simulate less UEs, BSs, frequency bands, smaller bandwidths, antennas with less elements, and shorter meetings in time.

This is the exact equivalent to selecting certain rows of a matrix, the difference is that we are doing it in 7D tensors.

Derivation

We can compute/derive information from a trace to obtain a trace under other specifications.

Although this works, it has not been used in this work. To scale up information, we can interpolate it, and to scale down we can average it or taking a subset. Let us suppose we have a trace for numerology 2. Can we also obtain the same trace for numerology 1 and 3?

Numerology 1 has less timestamps than numerology 2. To solve that we simply take every second time instant from numerology 2, which corresponds to the instants of numerology 1. We could also average every 2 timestamps. Also, in numerology 1 we have smaller subcarrier spacings, therefore, more resolution in frequency. For this, we can interpolate between PRBs.

To obtain numerology 3 we interpolate in time numerology 2 and subset in frequency. Note, however, that numerology 3 would have double the bandwidth of numerology 2 for the same number of PRBs, therefore, we need to use less PRBs to have the same carrier bandwidth at each frequency band.

Floating-point Precision

Although all the computations are made using double-precision for convenience since that is the default data type in Matlab, the channel coefficients are stored as single-precision floating points, to reducing memory usage.

The choice of single- instead of double-precision results from the range of values supported by each. Matlab constructs the single-precision data type according to IEEE Standard 754 [100] using 32 bits to store numbers roughly from 1.8×10^{-38} to 3.4×10^{38}. As such, single-precision allows us to represent numbers in the logarithmic scale in the range $[-380, 380]$ dB for field intensities or symbol energies, and half that for powers, i.e. $[-190, 190]$, which is sufficient for our application. This covers the range, regarding how precise the representation is, since only 23 of those 32 bits are used for the mantissa (fractional part), we have $1/2^23$ of precision before the representable value changes. In other words, about 6 decimal places that we're sure of being correct. We do not require that much precision, therefore singles can be used without loss of useful information.

Singles use 4 bytes per data point requiring half of the necessary memory of doubles. The gain is substantial when Terabytes of data are concerned. Additionally, besides saving and loading, subsequent operations on single-precision data are faster.

Power Variations across Frequency Band (tentative)

This strategy has not been implemented yet, but it is foreseen to yield major memory gains, and consequently result in major computation time savings. It should reduce the required memory between 50 and 275 times, depending on the bandwidth of that simulation.

The strategy consists in generating far less PRBs. If the channel does not change considerably from PRB to PRB, then we do not need to compute responses for every PRB. The extreme case is only requiring one PRB for the whole carrier bandwidth. But we may verify that only every 10 PRBs, or every 100 MHz it is worth having a frequency response.

Our choice to what how many PRBs is worth having is a trade-off between the error we are willing to accept and the computational burden of simulating all PRBs. In this trade-off, it matters the scheduling and beamforming strategy, i.e. do we give all resources and compute beams for the whole band (wideband), or are we more granular in our resource distribution and beam computation? Moreover, having 1-PRB channel traces would additionally yield advantages in terms of numerology compatibility, massively facilitate storage and even reduce simulation times since many processes like MIESM would simply be skipped and any operations that require access to memory would be considerably faster as well.