



<p><b>HORIZON 2020</b></p> 		<p>Deliverable ID: <b>D5.3</b></p>	<p>Preparation date: <b>03 April 2019</b></p>
		<p>Milestone: Final proposed</p>	
<p>Terahertz based Ultra High Bandwidth Wireless Access Networks</p>		<p>Title:</p> <p><b>Initial DLL Layer Model and Simulator</b></p>	
		<p>Editor/Lead beneficiary (name/partner): <b>Noureddine Boujnah / WIT</b></p>	
		<p>Internally reviewed by (name/partner): <b>Zdravko Bozakov/ DELL EMC Luis Pessoa/INESC TEC</b></p>	
		<p>Approved by: <b>PSC</b></p>	
<b>Dissemination level</b>			
<b>PU</b>	Public		x
<b>CO</b>	Confidential, only for members of the consortium (including Commission Services)		

<b>Revisions</b>				
<b>Version</b>	<b>Date</b>	<b>Author</b>	<b>Organisation</b>	<b>Details</b>
<b>0.1</b>	31.08.18	Noureddine Boujnah	TSSG	In progress
<b>02</b>	15.09.18	Noureddine Boujnah	TSSG	In progress
<b>0.3</b>	28.09.18	Noureddine Boujnah	TSSG	Ready for internal review
<b>0.4</b>	11.10.18	Noureddine Boujnah	TSSG	Revised
<b>0.5</b>	21.03.19	Noureddine Boujnah	TSSG	Revised based on reviewers comments

## Table of Contents

Table of Contents .....	ii
1 Introduction .....	6
1.1 Summary .....	6
1.2 Structure of this document .....	6
1.3 Relationships with other deliverables .....	6
1.4 Contributors.....	6
1.5 Abbreviations .....	6
1.6 Acronyms .....	7
2 Overview of layer’s functionalities and simulators .....	8
2.1 Physical Layer Simulator [TUBS] .....	8
2.1.1 Short description.....	8
2.1.2 Parameter inputs .....	9
2.1.3 Output and Metrics .....	9
2.1.4 Requirements for other Layers .....	9
2.1.5 Data Formats .....	10
2.1.6 Simulation Cases .....	10
2.2 Initial data link layer model and simulator [TSSG].....	10
2.2.1 Short description:.....	10
2.2.2 Parameter inputs .....	10
2.2.3 Output and Metrics .....	11
2.2.4 Requirements for other Layers .....	11
2.2.5 Data Formats .....	11
2.2.6 Simulation Cases .....	12
2.3 Network Layer Simulator [DELL].....	12
2.3.1 Short description.....	12
2.3.2 Parameter inputs .....	13
2.3.3 Output and Metrics .....	13
2.3.4 Requirements for other Layers .....	13
2.3.5 Data Formats .....	13
2.3.6 Simulation Cases .....	13
3 Initial data link layer model.....	15
3.1 General overview .....	15
3.2 Concept .....	16
3.3 DLL modules .....	16
3.3.1 Frame generator:.....	17
3.3.2 Error model.....	18
3.3.3 Packets to frames conversion .....	20
3.3.4 Error control: .....	25
3.3.5 Medium access techniques: .....	28



---

4 Conclusion/Further work ..... 29



## List of figures

<a href="#">Figure 1: General overview of the initial DLL simulator</a> .....	15
<a href="#">Figure 2: Constant frame generation</a> .....	17
<a href="#">Figure 3 Random inter-arrival frame generation</a> .....	17
<a href="#">Figure 4 Variable frame length generation</a> .....	18
<a href="#">Figure 5 Error model tuning</a> .....	18
<a href="#">Figure 6 From packet to frame</a> .....	21
<a href="#">Figure 7 TX and RX buffers</a> .....	22
<a href="#">Figure 8 Data in the TX buffer with(red plot) and without(orange plot) retransmission</a> .....	24
<a href="#">Figure 9 The CRC operation at transmitter (left) and at receiver (Right)</a> .....	25
<a href="#">Figure 10 Mean number of re-transmissions as function of BER and frame length</a> .....	26
<a href="#">Figure 11 <math>v</math> as function of BER for two frame lengths, with and without FEC</a> .....	27

## List of tables

<a href="#">Table 1 Layer description</a> .....	8
<a href="#">Table 2 Input Parameters for the Physical Layer Simulator</a> .....	9
<a href="#">Table 3 Parameter inputs for DLL simulator</a> .....	10
<a href="#">Table 4 Input Parameters for the Network Simulator</a> .....	13
<a href="#">Table 5 Overhead sub-fields</a> .....	22



### Executive summary:

The deliverable D5.3, in its first part, gives an overview of the different simulators that are foreseen in work package 5 of the TERAPOD project. In the second part of D5.3, initial data link layer models and modules are presented, and simulations are carried to assess system performance in term of throughput, system load, frame error rate and re-transmissions.



# 1 Introduction

## 1.1 Summary

This deliverable deals with the initial modelling and design of the data link layer simulator for THz communications of TERAPOD project. Some of simulator's modules are proposed and modelled for point to point link such as: packet and frame generator, symmetric and asymmetric error model, error control module, framing module and interface with the physical layer.

## 1.2 Structure of this document

This document is organized as follow:

- Section 1 introduces the document.
- Section 2 gives an overview on communication layers and simulators.
- Section 3 describes Initial data link layer model and simulator.
- Section 4 concludes the document and proposes further work.

## 1.3 Relationships with other deliverables

The deliverable D5.3 is the second deliverable of work package 5. The work presented in this document relates to the following deliverables:

- D2.1 – Initial Requirements and Scenario Specifications: requirements for point to point scenario.
- D5.1 – Initial Physical Layer Model and Simulator: some parts of section 2 are repeated through all deliverables of WP5 such as: initial physical layer model and simulator as well as initial networking layer model and simulator.

## 1.4 Contributors

The following partners have contributed to this deliverable:

- TSSG- Nouredine Boujnah
- TUBS- Johannes Eckhardt
- Dell EMC- Zdravko Bozakov

## 1.5 Abbreviations

P2P – Point-to-Point link.

P2M –Point-to-Multi-point link

M2M – Multiple points to Multiple points link

BER – Bit Error Rate

FER – Frame Error Rate

CRC – Cyclic Redundancy Check

DLL – Data Link Layer

FEC – Forward Error Correction

L3 – Layer 3, network layer

MAC – Medium Access Technique

MTU – Maximum Transmissin Unit

NL – Network Layer

PHY – Physical Layer

SNR – Signal to Noise Ratio

SINR – Signal to Interference plus Noise Ratio

TDM – Time Division Multiplexing

WP – Work Package



CIR – Carrier over Interference Ratio  
 RF – Radio Frequency  
 RTS – Request To Send  
 CTS – Clear To Send  
 THz – Terahertz  
 ACK – Acknowledgement  
 NAK – No acknowledgement  
 FIFO – First In First Out

## 1.6 Acronyms

$T$ : Time slot duration.

$T_f$ : Frame duration.

$T_b$ : Bit duration.

$R$ : Data rate.

$r(t)$ : The instantaneous data rate.

$\bar{r}$ : Mean data rate

$p$ : Bit error probability

$P_f$ : Frame error probability.

$\lambda_{\text{frame}}$ : Frame arrival rate

$\bar{l}$ : Mean number of re-transmissions.

$\xi$ : Mean number of errors per time unit

$\lambda$ : Poisson parameter for traffic generator

$N_p(t)$ : Number of received packet from network layer during  $[0,t]$

$N(t)$ : Number of generated frames during time interval  $[0,t]$

$L_{\text{frame}}$ : Frame size.

$L_{\text{bloc}}$ : Block size.

$L_{\text{overhead}}$ : Frame overhead

$B_{\text{size}}$ : Buffer size (Mbits)

$M_k$ : Number of frames in the TX buffer at  $t=kT$ .

$A_k$ : Number of frames in the RX buffer at  $t=kT$ .

$$(x)_+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{else} \end{cases}$$



## 2 Overview of layer's functionalities and simulators

Table 1 depicts the main functionalities and modules, key performance indicators and tools expected to be used by WP5 to build THz simulators.

Table 1 Layer description

	Physical layer	Data Link Layer	Network Layer
Models and functionalities	<ul style="list-style-type: none"> <li>• Modulation and coding schemes</li> <li>• Waveforms</li> <li>• Channel models</li> <li>• RF impairments</li> </ul>	<ul style="list-style-type: none"> <li>• Framing/De-framing</li> <li>• Error control</li> <li>• Flow control</li> <li>• Transmission modes</li> <li>• Scheduling and MAC</li> <li>• DLL Swiching</li> <li>• DLL signaling</li> <li>• Scheduling</li> <li>• MAC adressing</li> </ul>	<ul style="list-style-type: none"> <li>• Packet format</li> <li>• DC Topology</li> <li>• DC components</li> <li>• Network interfaces</li> <li>• Routing algorithms</li> <li>• Control plane</li> <li>• Signalling</li> <li>• Logical adressing</li> <li>• Fault detection</li> </ul>
Key performance indicators	<ul style="list-style-type: none"> <li>• Error probability</li> <li>• PHY layer delay</li> <li>• BER</li> <li>• Quantized SNR/SINR</li> </ul>	<ul style="list-style-type: none"> <li>• Throughput/goodput</li> <li>• Latency</li> <li>• Frame Error Rate</li> <li>• Number of re-transmissions</li> <li>• DLL buffer status</li> </ul>	<ul style="list-style-type: none"> <li>• Packet arrival rate</li> <li>• NL throughput</li> <li>• Load /congestion</li> <li>• Packet loss</li> <li>• Delay</li> <li>• NL buffer status</li> <li>• Recovery time</li> </ul>
Tools	SiMoNe	Matlab Ns-3	Ns-3 Mininet CORE

### 2.1 Physical Layer Simulator [TUBS]

#### 2.1.1 Short description

The increasing demand for high data rates motivates the development of multi-gigabit communication systems in the THz frequency range. This frequency range shows propagation mechanisms which differ from the already developed and well-known systems. It is therefore mandatory to investigate the transmission of single bits over the physical layer to evaluate the transmission quality. A physical layer simulator models the transmission taking the channel coding, the modulation and the radio channel into account. Since the new devices created for THz communication may influence the transmission in a significant way, hardware impairments lie also in the scope of the TERAPOD project. The physical layer simulator will leverage the THz channel measurement and modelling from WP4 to correctly simulate the channel influence on the transmission. It is planned as a dedicated tool which fully represents the physical effects occurring in THz communication.





Two types of deployment can be imagined for the physical layer simulator. The first and principle type of deployment uses a random bit generator to create some significant error statistics for a certain input parameter set. Then an error pattern generator will be programmed which will emulate the physical layer characteristics. This will enable the integration of the physical layer characteristics to higher level simulators.

The second type of deployment consists in the utilization of real data provided by TSSG. The real data will be processed by the physical layer simulator and will be given back to TSSG after the transmission. In this way real data communication can be tested and evaluated.

### 2.1.2 Parameter inputs

The physical layer simulation depends on a finite set of configuration parameters and information input. Table 2 shows the different parameters and their suppliers.

Table 2 Input Parameters for the Physical Layer Simulator

Parameter	Origin	Comment
Frame length	DLL	Variable or fixed
Transmit Power	WP4	SNR at the receiver is an equivalent value which can be used
CIR of the radio channel	WP4	Modelled by raytracing
Characteristics of RF hardware	WP4	Including: antenna characteristics, device properties

### 2.1.3 Output and Metrics

To evaluate the data transmission over the physical layer in dependence of the different input parameters the bit error rate BER will be measured. The BER is a common figure of merit that will act as a base of comparison for the different combinations of input parameters. For illustration reasons the theoretically possible data rate will also be calculated but the actual data throughput needs to be evaluated at higher layers. The BER itself does not represent all information of the error behaviour since correlation between error phenomena is possible. Therefore, an error pattern generator will be developed which on the one hand incorporates the whole error statistics and on the other hand enables the implementation of the statistical representation of the physical layer to higher layer simulators. This will be done by a compiled .dll file that can be integrated in any other piece of software.

Furthermore, the delay introduced by the physical layer will be evaluated in dependence of the input parameters. In the case that real data is processed by the physical layer simulator the processed data created during the simulation will be provided in an appropriate data format.

### 2.1.4 Requirements for other Layers

As the physical layer interacts with the DLL some agreements must be met to guarantee the accurate functionality.

One constraint for the DLL protocol concerns the frame length. Every time information is sent over the physical layer, the number of bits sent without an interruption need to be a multiple of the code word length and the modulation order.

In the physical layer simulator, a time-continuous simulation will be realized. This enables the calculation of a representative statistic for a certain parameter input set and enables a simulation of a transmission of real data. Therefore, a common data format between the DLL and the PHY layer needs to be defined.



To estimate the simulation time and the feasibility of the real data simulation, the active period and the total simulated time of the DLL simulator need to be dimensioned in cooperation with the PHY layer simulation resources.

### 2.1.5 Data Formats

At different interfaces, information need to be exchanged between the simulators and their respective partners. Therefore, several data formats will be defined which enable the collaborative work flow.

For the sake of a reasonable data size, high volume simulation data will be exchanged in a machine-readable binary data file. For test and debugging reasons, a human-readable XML alternative for small data volumes will be used in parallel. The syntax of the binary data file will be defined by TUBS and TSSG.

The error pattern generator will be provided as a compiled .dll file, or if possible as a MEX file. The CTF created as an output of the channel modelling will be represented in a data format which will be developed on the base of the data format presented in the IEEE802.15.3d channel modelling document [1].

### 2.1.6 Simulation Cases

The simulation cases for the data centre scenario in terms of the geometry will conform to the use cases presented in the deliverable D2.1. Regarding the used modulation and coding schemes, the simulation will consider all schemes which will be supported by the hardware developed in work package three. Generally, all combinations of the input parameters can be simulated. Simulation cases with a special interest will be worked on over the course of the project.

## 2.2 Initial data link layer model and simulator [TSSG]

### 2.2.1 Short description:

Data link layer lies between the network and physical layer; it is responsible for: traffic flow control, error control and medium access. The initial DLL model and simulator will focus on simple scenario P2P; we start by modelling some of THz DLL functionalities aligned with data center requirements. We model traffic flow, the number of frames in the system and buffer status, error generator and error control.

Through this deliverable, we propose basic algorithms and run simulations related to throughput for different channel conditions. For parameters setting, we use inputs from physical and network layer. The wireless link is more sensitive to channel impairment then additional functionalities at DLL and physical layer should be implemented, DLL and physical layer should coordinate together to track changes in the propagation channel and reduce data loss.

### 2.2.2 Parameter inputs

Table 3 Parameter inputs for DLL simulator

Parameter	Designation	Comments
Transmission time in micro second	$T$	During this time, the TX transmits one frame and the receiver processes it and sends a feedback to the transmitter, to be aligned with NL requirements. In this deliverable assumed to be fixed.
Bit time in nano second	$T_b$	Bit duration
Response frame length in bits	$L_{ack}$	Each frame transmitted will receive an acknowledgement (possibly each group of frames with one ACK).



Parameter	Designation	Comments
Switching time	$T_{sw}$	Time to switch from one node to another, it depends on antenna technology.
Error model	.dll/MEX file	Input raw frames, output processed frames with errors.
Topology parameters	From DELL/EMC from D2.1	Distance between nodes and DC topologies.
Simulation Time	$T_{sim}$	Between 0.1 s to 60 s
Packet length	$L_p$	At least 500 bits
Bandwidth	BW	To be set according to physical layer simulations.

### 2.2.3 Output and Metrics

- Frame Error Rate: simulation of FER for different frame sizes and bit error probability.
- Mean number of errors.
- Mean number of successful transmissions.
- Mean number of re-transmissions: for limited and infinite number of retransmissions.
- Throughput and goodput (useful throughput).
- Load at TX buffer: model proposed and simulation for high frame arrival and harsh propagation conditions (high frame error probability  $P_f$ ).
- Number of frames at RX buffer: model proposed.

### 2.2.4 Requirements for other Layers

- Networking layer:
  - Packet length
  - Packet arrival rate
  - Traffic priority and requirements
  - Possible DC topologies
  - Network interfaces
  - Network elements where DLL can be implemented: switches, repeaters, routers, servers...
- Physical layer:
  - Frequencies and bandwidths
  - Modulations techniques
  - Coding techniques
  - Quantized SNR / BER
  - Physical layer delay profile.

### 2.2.5 Data Formats

- Matlab file containing generated frames during a period  $T_{sim}$ .
- Frames with errors.
- Statistics related to: evolution of the number of frames in the system.



## 2.2.6 Simulation Cases

We run simulations for different scenarios; our work for this deliverable will be limited to P2P scenario:

- Frame generation: constant and variable throughput.
- Transmission with variable frame length: to study the impact of frame size on link performance such as FER and throughput.
- Re-transmission option without FEC and with FEC: impact on goodput
- Buffer status with and without retransmissions.

## 2.3 Network Layer Simulator [DELL]

### 2.3.1 Short description

As today's network topologies grow increasingly more complex, and are prone to repeated change, the beneficial use of network simulation tools have never been more relevant. It can be extremely useful to monitor the behaviour and performance of a network under various conditions and modify the environmental attributes associated with it. Though the terms simulator and emulator are quite often used interchangeably in the context of network design and testing, it is important to distinguish the difference between the two and understand what each type of tool can bring to the TERAPOD project.

Network simulation is a technique where a software program precisely models a network's behaviour and the interactions between its underlying network entities – switches, links, node and associated protocols. As a result, virtually arbitrary levels of fidelity can be realized given enough computation time and resources. On the other hand, network emulation aims to increase the efficiency for simulating complex network system by trading off some degree of reproducibility and accuracy to reduce the time and resource required for analysis. Emulation approaches aim to realistically replicate large-scale networks in a controlled environment, frequently utilizing existing stacks (e.g., TCP/IP) and technologies (e.g., virtualization) or close approximations thereof to test a specific aspect of a networked system.

Both methods and application types are used primarily in the fields of research & development and academia to shed light on fundamental networking concepts such as standard network node modelling, queueing, traffic analysis, and techniques like software-defined mechanisms or parallel system integration. Both approaches come with support for the most popular technologies and networks – including, but not limited to, wireless sensor networks, wireless LANs, LTE-5G, ADHOC networks etc. For any projects or work concerning the design, development and testing of a network – being able to properly simulate or emulate said network is a crucial tool.

In the case of TERAPOD, network simulators/emulators are of vital importance to demonstrate and study the communication properties of ultra-high bandwidth wireless THz communication. At the network layer, the approaches mentioned above will be used to assess the benefits of augmenting datacentre networks with dynamically configurable wireless links. Precisely, we will analyze how such links can be used to eliminate network hot-spots and improve utilization levels of the available network resources under typical datacentre traffic workloads. To this end, forwarding performance, queuing levels, and the responsiveness of SDN supported routing/forwarding approaches must be evaluated. Three different network simulation/emulation tools are considered for this purpose:

- *ns-3* – the third iteration of *ns*, a discrete event network simulator where events occur at a particular instance and mark state changes in the system. [2]
- *Mininet* – a network emulator that creates a virtual network of hosts, switches, controllers and links on a single Linux kernel. [3]
- *CORE* – Common Open Research Emulator, a tool for emulating networks on one or more machines that can also be connected to live networks. [4]



### 2.3.2 Parameter inputs

Table 4 Input Parameters for the Network Simulator

Parameter	Designation	Comments
DC topology	G	Logical DC topology graph subject to geometrical constraints
Wireless link capacity	C	Subject to geometry constraints
Wireless link latency	D	Total link latency
Wireless link setup time	$T_s$	
Maximum Transmit Unit	MTU	Length in bytes
Simulation Time	$T_{sim}$	Duration of the simulation
Traffic matrix/traces	$F$	Including statistical traffic characteristics

### 2.3.3 Output and Metrics

Generally, we expect our simulations to yield the following outputs:

- Total utilization: given equal network conditions, the overall utilization of the network is a key target metric. The goal of the work is to demonstrate a higher utilization.
- Number of network hotspots: network events, where the link utilization and/or queuing delays exceed a predefined threshold for a specific amount of time.
- Convergence time: time necessary for SDN infrastructure to respond to a network change by bringing up a new wireless link and the time necessary for network protocols to adapt to the resulting topology changes.

The above list will be refined and extended upon once the experimental environment has been finalized.

### 2.3.4 Requirements for other Layers

The physical and DL layers must provide the selected simulation tool with a high-level description of the terahertz wireless links, subject to the physical constraints of the developed datacentre. From the perspective of the DC operator, the links are characterized by a constant capacity (rate) over a specific time duration, as well as the delay (latency) experienced by data traversing the link. Thus, a simple latency rate curve representation, which has long been used for performance analysis from network calculus, will provide a suitable starting point for further evaluations. If necessary these can be extended to probabilistic equivalents at later stages of the work, to account for variability of the wireless links.

In addition, the setup latency must be provided (e.g., total time needed to provision a new wireless link).

### 2.3.5 Data Formats

Text based data formats will be used to encode the used network topology and associated component parameters, encoded e.g., using XML, JSON, depending on the utilized simulation tool.

### 2.3.6 Simulation Cases

We plan to simulate a fat-tree and spine-leaf datacentre topologies, as these are the most common wired Data Centre network topologies. In addition, the scenarios will use pre-generated traffic traces (e.g. using the traffic generator DCT<sup>2</sup>gen [6]) which will be injected into the topologies. As a result, the setup



will allow us to reproducibly compare the defined performance metrics with or without terahertz wireless links.



### 3 Initial data link layer model

#### 3.1 General overview

The initial simulator includes modules related to frame generation, theoretical error model and buffers to store frames for transmission or possibly re-transmissions:

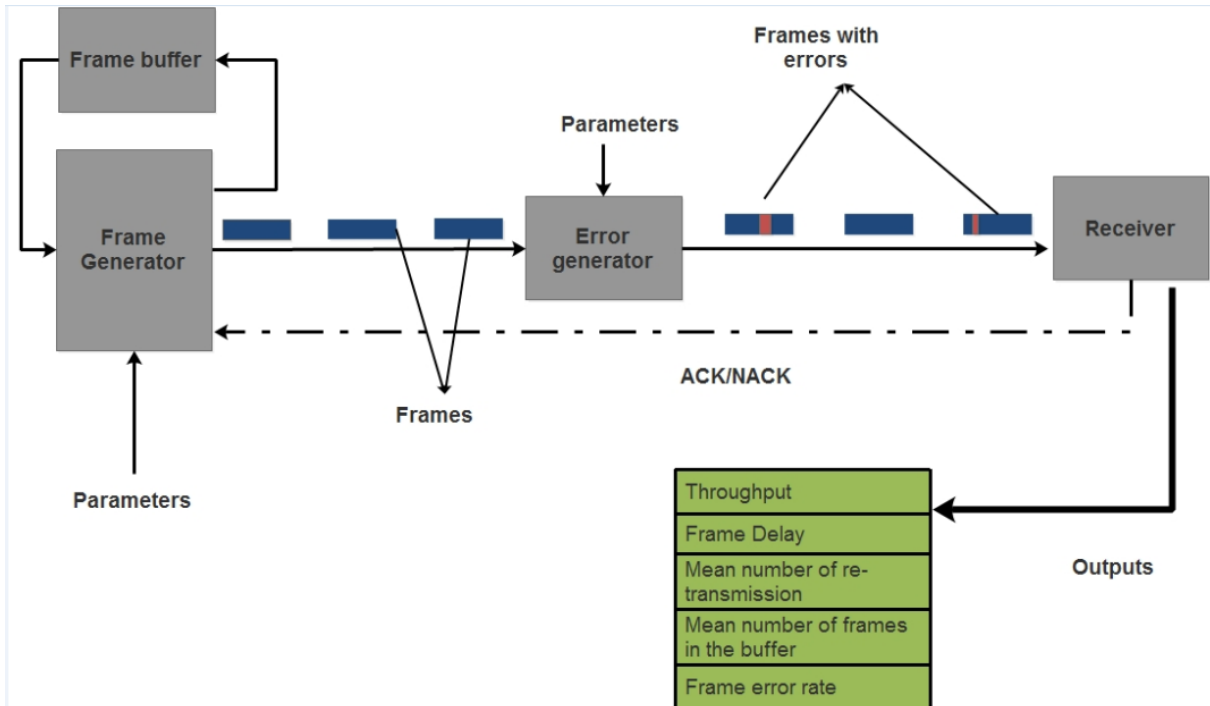


Figure 1: General overview of the initial DLL simulator

Figure 1 shows an initial architecture of a P2P THz link. Generated frames are characterized by a length  $L$  and a time slot duration  $T$ , the total traffic duration is  $T_{sim}$ . In fact,  $T$  needs to be determined based on: the total time required for framing, 2 ways propagation delay and processing time at receiver side plus a guard interval to avoid overload.

The determination of the time slot duration will be studied in M2M and P2M scenarios, one time slot can be dedicated to one link or shared between many links. The error generator, lying between transmitter and receiver, adds bit or burst errors to transmitted frames.

Functionalities described in this deliverable are:

- **Framing:** Packets coming from upper layer are mapped into frames, this module receives commands from error control module to add sophisticated CRC and select frame length and overhead.
- **Media Access:** Review of some algorithms used by previous works to access the shared link, and also techniques that could be suitable for data center use case, For P2P scenario we can implement a continuous access technique to the link or conditional access based on the buffer status as well as the channel status.
- **Error control:** Error module is required to mitigate frame loss (no feedback received) and corruption (NAK received), in this deliverable we consider Cyclic Redundancy Check (CRC) technique to decide if the frame is well received or corrupted, a command message is sent to re-transmit the frame.
- **Buffer load:** Buffers at both transmission sides store frames to be transmitted and frames to be processed respectively. The number of frames in the system should be studied based on DLL



assumption; the aim is to determine the buffer size for high data rate communication and harsh propagation environment.

- Additional functionalities will be considered such as interfacing between DLL and physical layer to match between frame length and physical layer parameters. The interface between DLL and networking layer will be also investigated to reports issues related to DLL overflow, frame drops and lack of radio resources.

## 3.2 Concept

Data link layer receives packets from upper layers and maps them into frames. Packets are segmented into small blocks (Data Units) with size  $L$  block, then assembled into frames with variable size. Preambles containing a sequence number, markers and protection bits using CRC are added to each frame. Using data units, the frame's length can be monitored. Moreover, DLL is responsible for flow control, error control and medium access.

We start by implementing a frame generator as described in section 3.1, at the transmitter side. Parameters of the traffic generator are:

- Frame length  $L$
- Frame rate
- Interframe time (equal to Time slot)

Moreover, frames with variable size can be generated using small blocks of data, each block is assigned a reference number. A theoretical error model is also described and implemented on MATLAB, it generates errors randomly, it can be symmetric or asymmetric. Parameter related to the theoretical error model module includes:

- Bit error probability  $p$
- Frame length  $L$
- Bit/frame duration (the deliverable D5.1 defines the bit and symbol duration already)

Each generated frame will be stored in a buffer with a given size. A FIFO method is implemented for frame transmission if the frame is correctly received it is discarded from the queue else it remains for a future retransmission (in this deliverable, it will be retransmitted in the next time slot). Only simplex mode is considered in this deliverable, further studies will be devoted for half and full duplex architecture, more investigation will be also addressed to handshaking strategies that fit TERAPOD requirements.

## 3.3 DLL modules

DLL simulator consists of independent modules linked to each other performing a set of functionalities. Links between modules should be a data flow, command messages or signalling.

In this deliverable we model and simulate the following modules:

- Frame generator: modelled and implemented, the model can be enhanced by realistic traffic collected from the data center.
- Error module: modelled in D5.3 and partially implemented, it can be validated by the physical layer error model that could be prepared by T5.1 after THz channel measurement and modelling.
- Error control: model and simulation of goodput for different parameter setting with and without retransmission.
- Framing: partially modelled and will be enhanced in the future, this module maps arriving packets to DLL frames.





- Buffer status at TX and RX: partially modelled, will be improved in the future, wireless THz DLL requires more memory to store frames and information related to MAC and flow control than wired DL.

### 3.3.1 Frame generator:

This module generates many traffic profiles, the idea is to create a plethora of traffic profile that could be suitable for a data center THz network. Parameters of frame generator are: frame size and inter frame period.

Scenario1: continuous transmission of bits:

Continuous transmission of bits during the simulation period is equivalent of the transmission of a long frame. At the physical level, it is possible to determine bits that are corrupted; then it is possible to decide on few parameters of error statistics such as BER, number of errors per time unit, mean time between two successive failures. Realistic communication traffic cannot be associated with this scenario.

Scenario 2: Constant frame generation

Time slot and frame size are fixed, receiver is well synchronized with the transmitter. Data rate  $r$  of the link is constant, and it is given by:

$$r(t) = \frac{L}{T} \tag{eq.1}$$

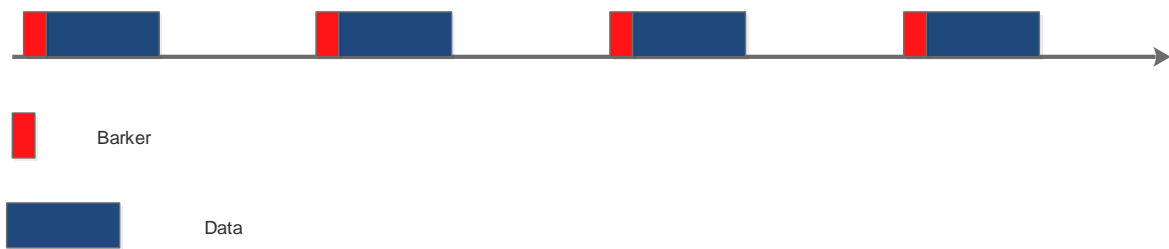


Figure 2: Constant frame generation

Once the synchronization is achieved between the transmitter and the receiver, then the receiver should know about the frame arrival. Figure 2 shows distribution of frames as function of time for this scenario. This scenario can suit real time traffic such as voice traffic, TV broadcasting.

Scenario 3: Random frame generation with fixed frame size: Frames are generated at random time with fixed frame size as it is shown in figure 3. There are issues related to frame synchronizations at receiving side and more processing complexity.



Figure 3 Random inter-arrival frame generation

Frame inter arrival time follows the shifted exponential distribution with parameters  $\lambda$  and  $\tau$ . then the probability density function is;

$$f(t) = \begin{cases} \lambda e^{-\lambda(t-\tau)} & t > \tau \\ 0 & else \end{cases} \tag{eq.2}$$



Transmitter waits for  $\tau$  seconds before starting generation of a new frame with rate  $\lambda$ , this time is required for processing at receiver side and for feedback from receiver of the previous frame, the mean inter-generation time is  $\frac{1}{\lambda}$ . The instantaneous data rate is:

$$r(t) = \frac{L}{T_f + t} \quad (\text{eq.3})$$

$r(t)$  is a random variable, its mean value is given by:

$$\bar{r} = \lambda L \int_{\tau}^{+\infty} \frac{e^{-\lambda(\tau-t)}}{T_f + t} dt \quad (\text{eq.4})$$

Relationship between  $\lambda$  and  $\lambda_{\text{frame}}$  will be investigated in presence of error model.

**Scenario 4:** Variable frame size generator: The frame generator outputs frames with variable length  $L$  within a fixed time slot  $T$ . The THz channel affects transmissions by adding errors to transmitted frames. For harsh channel conditions, Transmission of frames with low frame size may reduce FER, for good channel conditions sending frames with high frame enhance throughput. Choice of frame size is monitored based on feedback received from the receiver (ACK/NAK) as well as physical layer.



Figure 4 Variable frame length generation

The instantaneous data rate is:

$$r(t) = \frac{L(t)}{T} \quad (\text{eq.5})$$

### 3.3.2 Error model

The error module inserts errors randomly using some stochastic distributions that can be validated by real measurements and tests. There are two kinds of errors: bit error and burst errors:

- Binomial distribution of errors
- Poisson distribution

Based on physical link simulation, we can tune the error model as depicted in figure 5.

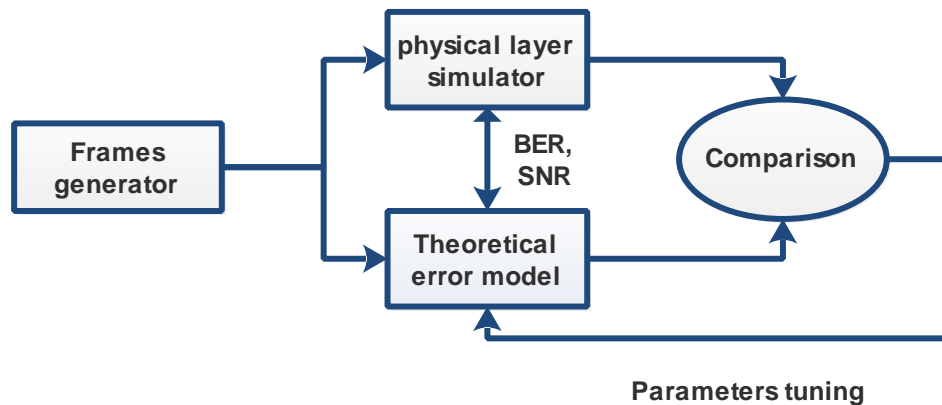


Figure 5 Error model tuning

Parameters to be tuned are:



- Number of errors per time unit
- Mean number of errors
- Mean number of errors by block of  $L_{\text{block}}$  bits
- Number of corrupted frames by time unit
- Frame Error Rate

Theoretical model can be updated; each time new enhancement for the physical layer simulator takes place.

### 3.3.2.1 Binomial distribution of errors with parameters $p$ and $L$

The probability of  $k$  errors in the frame is:

$$\text{Prob}(k \text{ errors}) = C_L^k p^k (1-p)^{L-k} \quad (\text{eq.6})$$

Where,  $C_L^k = \frac{L!}{k!(L-k)!}$ ,

And the probability of at least one error occurs in one frame is:

$$P_f = 1 - (1-p)^L \quad (\text{eq.7})$$

The mean number of errors with given  $p$  and frame length  $L$  is:

$$\bar{k} = Lp \quad (\text{eq.8})$$

As equation (Eq.8) shows, to reduce errors, we need to shorten frames when bad channel conditions are detected ( $p$  is high). If FEC implemented, the receiver could detect and correct several errors in the frame; the sender will receive a positive response from the receiver when all errors are detected and corrected. FEC techniques will be implemented at the physical layer and will enhance THz transmission.

The probability of  $k$  successful transmissions after  $n$  transmissions is:

$$P(k, n) = C_n^k (1-p)^{kL} (1 - (1-p)^L)^{n-k} \quad (\text{eq.9})$$

The mean number of decoded frames at receiver side between 0 and  $nT$  is:

$$k_{succ}(n, p, L) = n(1-p)^L \quad (\text{eq.10})$$

The error probability can be retrieved from  $k_{succ}$  as:

$$p = 1 - \frac{1}{L} \log\left(\frac{k_{succ}}{n}\right) \quad (\text{eq.11})$$

Then, it is possible to determine the error probability at the physical layer just by counting the number of previous successful transmission and applying (eq.11).

### 3.3.2.2 Poisson distribution with parameters $\xi$ and $T_f$

Errors can be described as stochastic process depending on time, as the performance of the wireless link can change with time. Fluctuations come from updates of physical layer parameters and the channel itself.

$$\text{Prob}(k \text{ errors during } T_f) = \frac{(\xi T_f)^k}{k!} e^{-\xi T_f} \quad (\text{eq.12})$$

$T_f$  and  $L$  are related by:

$$T_f = LT_b \quad (\text{eq.13})$$



The probability that at least one error occurs in the frame, if  $L$  and  $T$  are assumed to be deterministic, is:

$$P_f = 1 - e^{-\xi T_f} \quad (\text{eq.14})$$

For variable frame length, and uniform distribution of frame size, frame error probability is given by:

$$P_f = 1 - \frac{1}{L_{max} - L_{min}} \sum_{k=L_{min}}^{L_{max}} e^{-\xi k T_b} \quad (\text{eq.15})$$

$\xi$  denotes the number of errors by time unit, it is related to  $L$ ,  $p$  and  $T$  as:

$$\xi(L, T) = \frac{Lp}{T} \quad (\text{eq.16})$$

For Poisson traffic described in subsection 3.3.1, the time slot is:  $T(t) = LT_b + t$

Then, the mean number of errors by time unit is given by:

$$\bar{\xi}(L) = Lp \int_{\tau}^{+\infty} \frac{e^{-\lambda(\tau-t)}}{LT_b+t} dt \quad (\text{eq.17})$$

Channel and physical layer parameters choice affect  $p$ , for instance, for good channel conditions  $p$  is low, then it is possible to add more information bits to the frame and increase the throughput. The constant frame generation is more suitable for static transmission, but in a realistic scenario, traffic and errors are not statics.

### 3.3.3 Packets to frames conversion

Packets arrive in DLL at a given rate, packet size is randomly distributed, a mapping process is required to build output frames.

#### 3.3.3.1 The dynamic conversion equation

DLL maps packets into frames with size  $L$  and overhead  $L_{overhead}$ . The following equation holds:

$$N(t)(L - L_{overhead}) = \sum_{i=1}^{N_p(t)} L_p(i) \quad (\text{eq.18})$$

Equation (eq.18) describes at time  $t$ , the mapping between  $N_p(t)$  packets into  $N(t)$  frames with size  $L$  and overhead size  $L_{overhead}$ , Frame rate can be computed as:

$$\lim_{t \rightarrow +\infty} \frac{N(t)}{t} = \frac{1}{L - L_{overhead}} \lim_{t \rightarrow +\infty} \left[ \frac{N_p(t)}{t} \frac{1}{N_p(t)} \sum_{i=1}^{N_p(t)} L_p(i) \right] \quad (\text{eq.19})$$

Then

$$\lambda_{frame} = \frac{\bar{L}_p}{L - L_{overhead}} \lambda_p \quad (\text{eq.20})$$

$\bar{L}_p$  is the average packet size.

If the frame error increases, the node stops its transmission till the channel becomes available or sends short frames to keep the frame error rate below a threshold. We will show in the Error control paragraph that frame error rate FER decrease if  $L$  decrease, the drawback is that throughput decrease too. To describe traffic flow coming from a higher layer (network layer or optical DLL layer), the following inputs are required:

- Packet lengths for data center scenario.

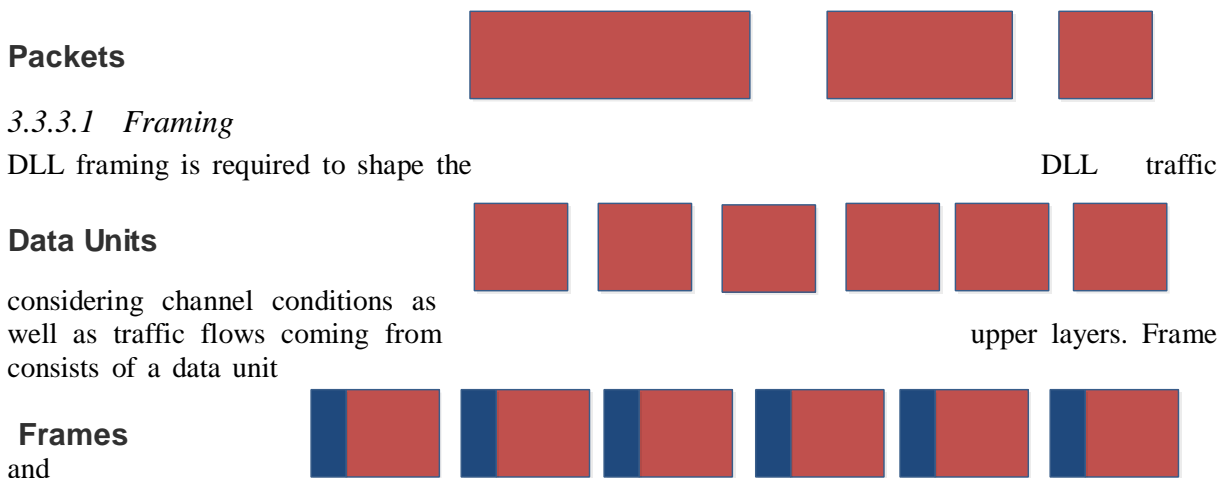


- Mean packet inter-arrival time: can be collected during an observation period.
- Service requirements of each flow: delay, data rate and drop rate.

A joint work with T5.3 will aim at analysing traces coming from network layer to determine the above parameters.

Packets are segmented and converted into small data units or cells with length  $(L-L_{overhead})$ . Framing operation introduces a delay at the transmitter as well as de-framing at receiver, sophisticated hardware with good processing time will be required to perform this functionality.

The determination of the stochastic distribution of frames using the dynamic equation (eq.18) seems to be a challenging task. Therefore, we can assume a given frame distribution. Further studies on this subject will be addressed in the future, as the data center traffic profile is still unavailable.



3.3.3.1 Framing

DLL framing is required to shape the

considering channel conditions as well as traffic flows coming from

and

overhead; we propose the framing procedure described in figure 6. To mitigate to channel impairment in terahertz link, we should define a new frame structure suitable for transmission over the wireless channel. For example, for larger frame size, two or more data units can be combined in one frame. Further studies will be carried out to determine the data unit size. For multiple traffic flows, frames are built from blocks belonging to the same flow; this solution can also be investigated in more details in the future.

Figure 6 From packet to frame

As described in subsection 3.2.2, frame length depends on BER, measured by the physical layer. BER measurement reflects the channel status and can be exploited to set  $L$ . Each frame consists of payload and frame overhead: header and tail. Table 4 gives a description of overhead subfields containing frame



identifier, a block identifier, a synchronization sequence(optional) and CRC subfield. In the simulator, a function responsible for frame building should be created, the output of this function is linked to the TX buffer. In the TX buffer, a table containing SN of frames waiting for transmission and their corresponding waiting time should be computed periodically. SN with highest waiting time should be prioritized for transmission to reduce mean delay per frame.

Overhead length can be optimized, for example, for a synchronous network, each node knows when it should start transmission and when it should receive. For an asynchronous network, a specified sequence with given statistical properties should be inserted at frame header to indicate the start of a frame or a transmission.

Table 5 Overhead sub-fields

Designation	Description
SN	Sequence number of the frame, the MAC module selects frame with SN to be transmitted
BRN	block Reference Number: each block requires a reference number, to be mapped to its packet at receiver side
Unique number	Node Unique Number: the number assigned by the manufacturer to the node
Sync(optional)	Synchronization sequence is required for asynchronous network: added to frame header to recognize the frame at the receiving side, we consider simple synchronization sequence such as 11011101(1 byte)
CRC sequence	CRC sequence: adding redundancy bits using polynomials division to each block/frame in order to detect and correct errors.
Flag	To determine start and end of frame
Type	Type of frame: Data frame or control frame

### 3.3.3.2 Initial buffer design:

Buffer design at the transmitter and receiver side depends on transmitter and receiver capabilities in term of:

- Frame processing at the transmitter side.
- Arrival rate of packets coming from higher layers.
- Physical layer algorithms performance such as coding, modulation.
- Channel impact on the signal.



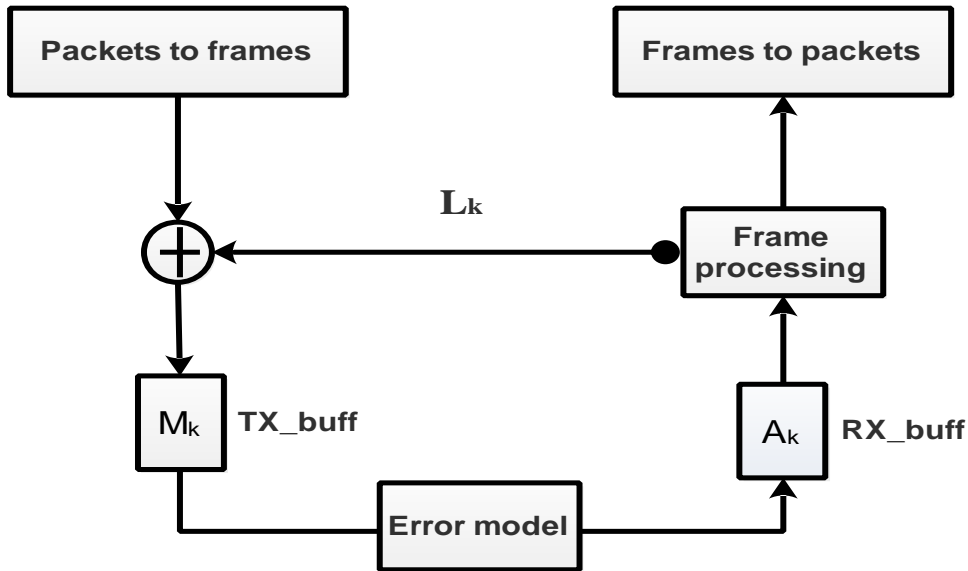


Figure 7 TX and RX buffers

Figure 7 shows the TX and RX buffers for P2P architecture with error control functionality. In the following, we analyze factors that impact buffer load in term of the number of frames in the system. More features could be implemented based on this analysis such as scheduling algorithms based on load. At the receiver side frames are processed, only correctly decoded frames are accepted, a NAK should be sent to the sender to re-transmit corrupted frames. Frames remaining in the system should be reduced, via DLL/PHY parameters, to increase data throughput as well as minimize mean frame delay.

- **Buffer design at the transmitter side**

The number of frames  $M_k$  at time  $t = (k + 1)T$  in the system is modelled by the dynamic equation as:

$$M_{k+1} = \min(M_k, L_k) + (M_k - 1)_+ + N_{k+1} \quad (\text{eq.21})$$

Where:

$$L_k = \begin{cases} 1 & \text{if re-transmission} \\ 0 & \text{else} \end{cases}$$

And

$$N_k = \begin{cases} 1 & \text{if frame received} \\ 0 & \text{else} \end{cases}$$

$$M_0 = 0, L_0 = 0$$

$N_{k+1}$  is the number of frames arriving from framing module during the interval:  $[kT, (k + 1)T]$ , where  $T$  is the service time for one frame. In This deliverable, only deterministic time slot is used, but we can also consider a random time slot duration,  $M_k$  increases for high load as well as for high re-transmission rate.

In some theoretical works, buffer size is assumed to be infinite. In practice buffer have a limited capacity  $B_{\text{size}}$  that depends on actual technology. The buffer state can be modelled as Markov chain, and then from a state  $n$  it is possible to go to state  $(n+1)$  or state  $(n-1)$ . Transition probabilities of the chain can be expressed as function of BER, frame length, mean packet length and frame overhead.

- **Buffer design at the receiver side**

Each received frame should be stored at the RX memory then processed, if no errors detected, it will be sent to upper modules for packet reconstruction and then transmitted to upper layer, otherwise it is



considered as corrupted and a feedback message is sent back to the transmitter requiring a re-transmission of it.

The dynamic equation of the number of frames in RX memory during an interval:  $[kT, (k + 1)T]$  is:

$$A_{k+1} = (A_k + 1 - L_{k+1} - P_{k+1})_+ \quad (\text{eq.22})$$

Where;

- $P_{k+1}$ : is the number of frames without errors to be mapped into packets.
- $L_{k+1}$ : the number of corrupted frames to be retransmitted by the transmitter,  $L_{k+1}$  depends on the error model
- Eq.21 and eq.23 share the same parameter  $L_k$ : we assume one feedback message from the receiver to each frame received.

RX sends a message to TX to delay its transmission when its buffer is overloaded, unless TX buffer is not loaded, otherwise a load report should be sent to higher layer to take appropriate decisions to avoid discarding frames before the load reach critical threshold. This matter can occur for high data rate scenarios and low device memory. Emphasis on buffer design for different possible scenarios will help to select the suitable buffer size in both the transmitter and the receiver.

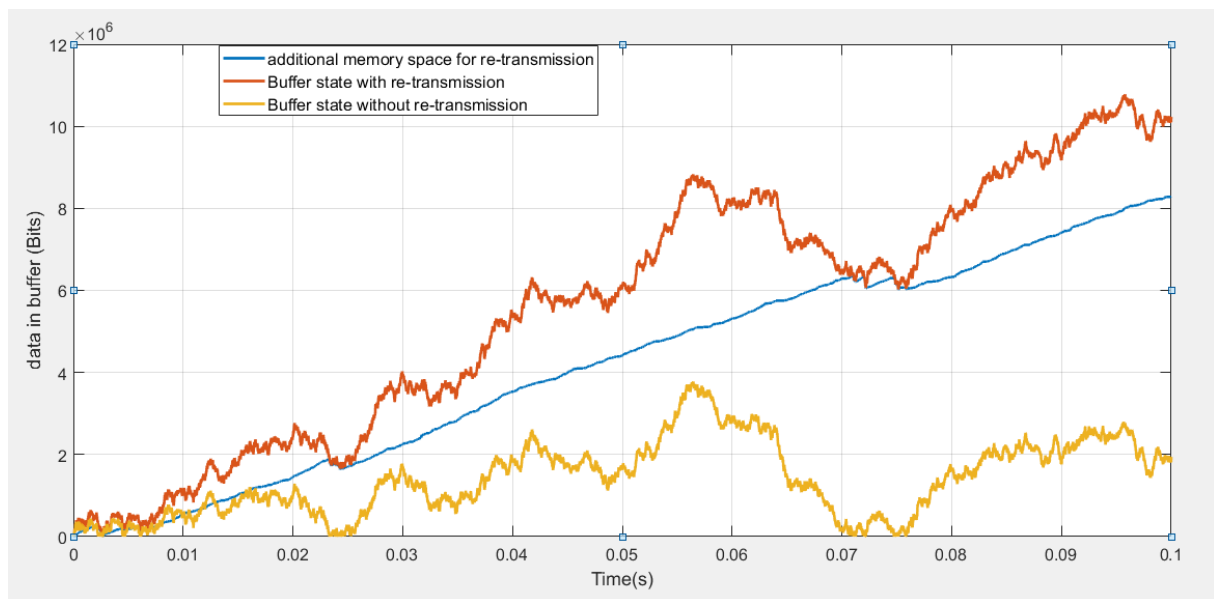


Figure 8 Data in the TX buffer with retransmissions (red plot), and without retransmissions (orange plot)

Figure 8 shows the simulation of load evolution of the TX buffer as function of time with and without re-transmission procedure, the blue plot describes the amount of memory required for only re-transmissions. Parameters set up are the following:

- $T \lambda_{frame} = 95\%$
- $L = 10000 \text{ bits}$
- $P_f = 0.1$
- $T = 1 \mu s$





- $T_{sim}=0.1s$
- $R=100Gbps$

To assess the link performance, we consider a highly loaded system and a frame length of  $L=10000$  bits. By simulating the dynamic evolution of the memory of TX node for P2P scenario and considering errors in the transmission channel, we remark that for high packet arrival rate or high service time and harsh propagation conditions, the system can overload if re-transmission is activated. Further investigations will be carried to model the buffer status of the system using Markov Chain and how to avoid system overload. Enhancement of Error control module can reduce the memory load.

### 3.3.4 Error control:

Techniques for error control range from insertion of redundancy bits at frames tail to re-transmission operations.

#### 3.3.4.1 Data integrity:

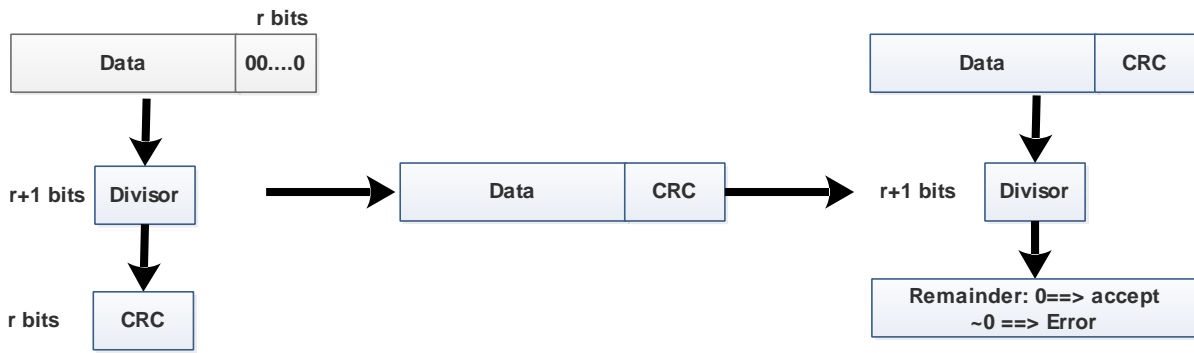


Figure 9 The CRC operation at transmitter (left) and at receiver (Right)

The mathematical operation of Cyclic Redundancy Check CRC is described by the equation:

$$D(X)X^r = d(X)p(X) + R(X) \tag{eq.23}$$

Where

- $D(X)$ : The data polynomial
- $d(X)$ : The polynomial divisor with degree  $r+1$ .
- $R(X)$ : The CRC polynomials, coefficients of  $R(X)$  are inserted in frame tail.

CRC-12, CRC-16 and CRC-32 polynomials are commonly used by many communication systems.

Protection of frame's header can be also performed to increase, for example, the sequence number and block number can be affected by errors, then CRC for this subfield can be added. To guarantee more protection to CRC field it is suggested that frame tail uses sophisticated modulation and coding schemes (MSC) such as using low order modulation. At the receiver side, we assume always that CRC field is protected against errors. Figure 9 describes how to build the CRC subfield and how to check the frame integrity at the receiver side.

In addition to the channel coding implementation at the physical layer, we add more data integrity techniques, in DLL, to mitigate channel impairments, for example, each data block is associated with its CRC field and any corrupted block can be re-transmitted in the next time slot, rather than re-transmission of the whole frame, then it is possible to increase data throughput and reduce frame error rate.



### 3.3.4.2 Retransmission without FEC:

Corrupted frames at receiver side are discarded, as a solution to this issue, the re-transmission mechanism can be implemented, FEC can be applied at the physical layer to detect and correct errors. It is also possible to use both FEC with low rate and re-transmission to increase success rate probability. The drawbacks are increasing frame delays, more computation complexity. One successful re-transmission occurs after  $(l-1)$  failures with  $l > 1$ , the probability of re-transmission of  $(l-1)$  frames is:

$$P_r(l) = P_e(p, L)^{l-1} (1 - P_e(p, L)) \quad (\text{eq.24})$$

The mean number of re-transmissions is given by:

$$\bar{l} = \sum_{l=1}^{+\infty} l P_r(l) = \frac{1}{1 - P_e(p, L)} = \left( \frac{1}{1-p} \right)^L \quad (\text{eq.25})$$

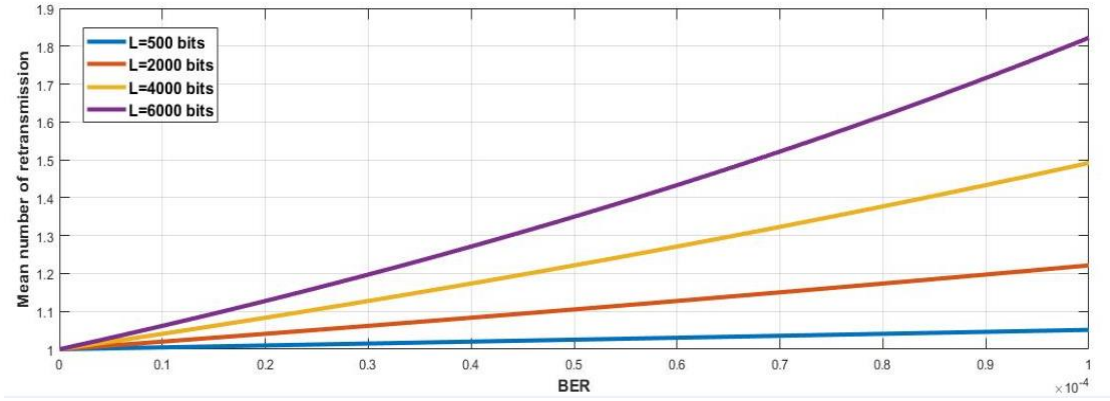


Figure 10 Mean number of re-transmissions as function of BER and frame length

Figure 10 shows that the number of re-transmission of short frames is less than long frames for the same BER. The instantaneous goodput is given by:

$$R(l) = \frac{L}{lT} \quad (\text{eq.26})$$

The mean goodput is given by:

$$\bar{R} = \sum_{l=1}^{+\infty} R(l) P_r(l) = \frac{-L^2(1-p)^L \log(1-p)}{T(1-(1-p)^L)} \quad (\text{eq.27})$$

$\bar{R}$  can be expressed also as function of mean number of re-transmissions  $\bar{l}$ :

$$\bar{R} = \frac{L \log(\bar{l})}{T \bar{l} - 1} \quad (\text{eq.28})$$

The number of re-transmissions increases when frame length increases for given BER. System overload can occur when frames remain a long time in the buffer. To cope with this issue, we can design MAC layer with variable frame length. Channel conditions affect bit error probability, more precisely BER is a function of signal to noise ratio SNR. Coding and modulation techniques implemented in the physical layer may reduce errors within a frame by adding additional bits for error detection and correction.

If the number of re-transmissions is limited to  $N_{ret}$ , the mean number of re-transmissions is:

$$\bar{l} = \sum_{l=0}^{N_{ret}} l P_r(l) = \frac{(N_{ret}-1)P_e(p, L)^{N_{ret}} - N_{ret}P_e(p, L)^{N_{ret}-1} + 1}{1 - P_e(p, L)} \quad (\text{eq.29})$$

And the mean goodput becomes:



$$\bar{R} = \sum_{l=1}^{N_{ret}} R(l)P_r(l) = \frac{L}{T} (1 - P_e(p, L)) \sum_{l=1}^{N_{ret}} \frac{1}{l} P_e(p, L)^{l-1} \quad (\text{eq.30})$$

The mean data rate depends on channel conditions as well as on  $L$  and service rate. The service rate is mainly related to the receiver capabilities of frame and bits processing as well as converting frames into packets. Enhancements of terahertz devices can contribute to the throughput enhancement, for example reducing processing time and device's noise.

### 3.3.5.2 Retransmission with FEC at physical layer implemented

By insertion of control bits for frame correction at the physical layer. The total number of bits affected by errors are reduced to  $M < L$ . The frame error probability is then:

$$P_f = 1 - \sum_{k=M}^L C_L^k p^{L-k} (1-p)^k \quad (\text{eq.31})$$

With FEC technique  $P_f$  will be reduced comparing to the previous case, where no FEC considered.

The mean number of re-transmissions with error correction capability is then given by:

$$\bar{l}(p, L) = \frac{1}{\sum_{k=M}^L C_L^k p^{L-k} (1-p)^k} \quad (\text{eq.32})$$

The number of re-transmissions will decrease when  $M < L$ ,  $(L-M)$  is the number of detected errors; we assume also that the number of detected errors at the receiving side is equal to the number of corrected errors at the physical layer. Although processing frame with FEC improves frame error rate and throughput, it introduces more delays to each frame transmitted.

The mean useful throughput is given by:

$$\bar{R} = \frac{-L \sum_{k=M}^L C_L^k p^{L-k} (1-p)^k \log(\sum_{k=M}^L C_L^k p^{L-k} (1-p)^k)}{T(1 - \sum_{k=M}^L C_L^k p^{L-k} (1-p)^k)} \quad (\text{eq.33})$$

Let define the parameter  $\nu = \frac{\bar{R}T}{L}$ , figure11 describes, for a fixed frame time slot  $T$ , the variation of  $\nu$  as function of BER and frame size  $L$ .

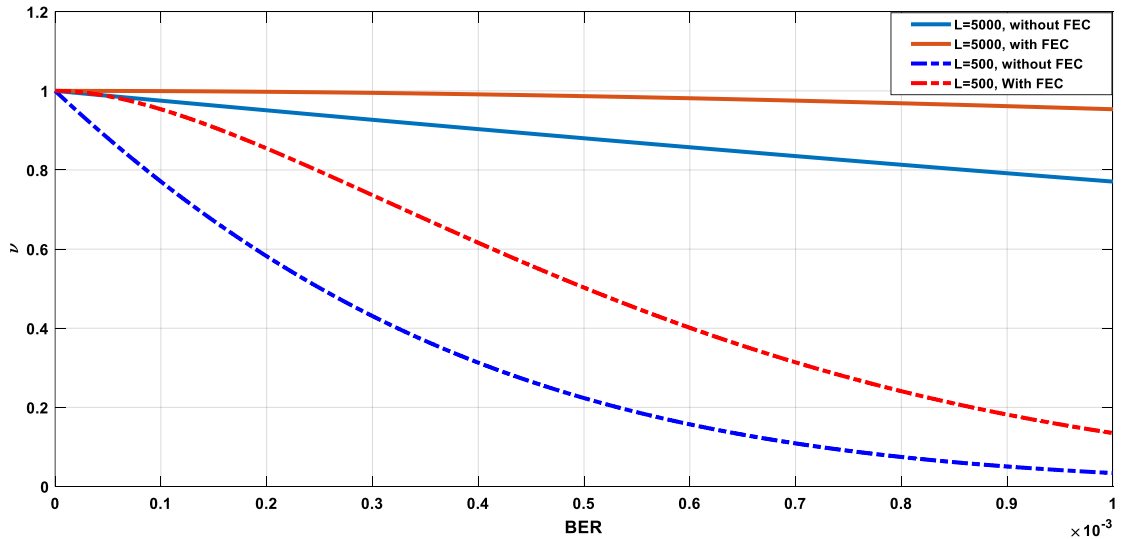


Figure 11  $\nu$  as function of BER for two frame lengths, with and without FEC

BER depends on channel conditions, the distance between source and destination and receiver performance.

By increasing  $L$ , data throughput increases, and frame error rate will increase too, to keep the frame error rate under a given threshold, we can choose  $L$  properly or we change dynamically frame length by tracking the channel changes.



### 3.3.5 Medium access techniques:

The medium access module is responsible for the organization of transmission in the shared link, it avoids interference and reduces collisions between shared users. In literature, many algorithms were proposed. For terahertz communications, antenna beams are tight, and three choices are possible:

- Terahertz communications is assisted by lower band technology: Terahertz system coordinates with an intermediate system (for example, WIFI at 2.4 GHz) to facilitate link establishment. Messages such as request to send (RTS) and Clear to send (CTS) can be sent over this lower band. Transmission initiation and termination are managed at low band system, and terahertz band is used when the link is set. It will be advantageous to use this technique to route signalling traffic to lower band then to increase bandwidth efficiency for the THz band. Drawbacks remain in beam steering, for example, errors in the angle of arrival estimation lead to beams misalignment, additional timing and power are required [1].
- MAC techniques based on beam/multi-beam switching techniques: transmissions are scheduled based on some criteria such as delay minimization, throughput maximization, and interference reduction. This technique works without any assistance from lower band technologies, then link establishment, synchronization, node discovery is performed by terahertz nodes. [5]
- Terahertz communications assisted by the optical network: the terahertz link will be deployed to alleviate the load on the optical network, it is possible to use the existing network to set up terahertz communications and to synchronize between nodes.

In this deliverable, we assume continuous frame transmission and no frame scheduling is required, if the buffer is not empty, transmission is triggered for each new frame period, and the error control module will manage the retransmission procedure. More investigations on MAC techniques for multiple communication scenarios will be scheduled in the future.



## 4 Conclusion/Further work

The deliverable D5.3 gives a general description and different inputs/outputs of each layer and related simulators. It also highlights dependencies with deliverables from other work packages. The main part of the deliverable is devoted to the description of the initial architecture of the DLL based on point to point link scenario and the modelling of some DLL functionalities.

The functionalities modelled and simulated in D5.3 are the following: frame generation, theoretical error model, buffer model, and error control. Models will be tuned and enhanced in the future using realistic data center traffic and simulation outputs from physical layer simulator using DC THz channel.

Simulations of throughput, frame error rate, number of retransmissions and buffer load are also presented for different channel conditions and physical layer parameter set up assumption (for example: with or without FEC).

DLL depends mainly on the physical layer, RF devices as well as network layer, the interface between consecutive layers helps to select the appropriate algorithms for DLL such as error control and MAC algorithms.

As future work, we will select the appropriate MAC algorithms for THz DLL as well as flow control. We will also enhance functionalities and models described in this deliverable such as framing and error control and feedbacks from the receiver. Network synchronization will also be investigated in the future.



## References

- [1] IEEE 802.15.3d IEEE Standard for High Data Rate, Wireless Multi-Media Networks, Amendment 2: 100 Gb/s Wireless, Switched Point-to-Point Physical Layer
- [2] ns-3 discrete-event network simulator for research and educational use - <https://www.nsnam.org/>
- [3] Mininet network emulator, creating virtual networks on a single machine - <http://mininet.org/>
- [4] CORE, Common Open Research Emulator for network emulation on multiple machines - <https://www.nrl.navy.mil/itd/ncs/products/core>
- [5] X.W. Yao, J. Jornet "TAB-MAC: Assisted beamforming MAC protocol for Terahertz communication networks" Nano communication Network (Elsevier) Journal, vol. 9, pp.36-42, 2016
- [6] Philip Wette, Holger Karl, DCT2Gen: A traffic generator for data centers, Computer Communications, Volume 80, 2016, Pages 45-58, ISSN 0140-3664,

