



**UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH**



**TECHNISCHE
UNIVERSITÄT
WIEN**

**Implementation of traffic model for different types of applications
and development of QoS aware resource schedulers**

A Degree Thesis

Submitted to the Faculty of The

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

And

Technische Universität Wien

by

Marc Lechado Puig

In partial fulfilment

of the requirements for the degree in

TECHNOLOGY AND TELECOMMUNICATIONS SERVICES ENGINEERING

Advisors:

Ramon Ferrús

Stefan Schwarz

Barcelona, June 2020

Abstract

Implementation of traffic model to the new 5G technology is a requirement to see how it will work and how it will improve the communications in comparison with the old technologies.

Another important point is having a nice scheduler that assign in the best way the RBs to the multiple users to have the best communication possible. In this work have been implemented some traffic models that take into account communication properties like the latency, and also have been implemented different schedulers that assign RBs to the users depending on what this scheduler is based on.

It has been seen that the best scheduler is the once that is based on assign RBs to these users that have the best CQI.

Resum

La implementació de models de trànsit a la nova tecnologia 5G és un requisit per veure com funcionarà i com millorarà les comunicacions en comparació amb les tecnologies antigues.

Un altre punt important és tenir un bon planificador que assigni de la millor manera els RBs als múltiples usuaris per tenir la millor comunicació possible. En aquest treball s'han implementat alguns models de trànsit que tenen en compte propietats de les comunicacions com la latència, i també s'han implementat diferents planificadors que assignen RBs als usuaris depenent en què es basa aquest planificador.

S'ha vist que el millor planificador és el que es basa en assignar RBs a aquells usuaris que tenen el millor CQI.

Resumen

La implementación de modelos de tráfico en la nueva tecnología 5G es un requisito para ver cómo funcionará y cómo mejorará las comunicaciones en comparación con las tecnologías antiguas.

Otro punto importante es tener un buen planificador que asigne de la mejor manera los RBs a los múltiples usuarios para tener la mejor comunicación posible. En este trabajo se han implementado algunos modelos de tráfico que tienen en cuenta propiedades de las comunicaciones como la latencia, y también se han implementado diferentes planificadores que asignan RBs a los usuarios dependiendo de en qué se base este planificador.

Se ha visto que el mejor planificador es el que se basa en asignar RBs a esos usuarios que tienen el mejor CQI.

Dedication

I dedicate my work to my family and my friends.

I would like to gratitude to my parents and my brother, who have helped me in these moments that I felt I couldn't keep my mind concentrated. They also helped me to have a comfortable work environment.

Furthermore, I dedicate my thesis to my friends and my girlfriend. They helped me to have a good time and to be disconnected for a while.

Finally, I also dedicate my work to my college friends, specially to Alex Alujas, who shared with me a month of our Erasmus and try to help me as much as he could.

Acknowledgements

I'm very grateful to have had Stefan Schwarz as a supervisor in TU Wien. He has been helping me during all these months, including when I had to come back to Spain due to the pandemic. He always answered my emails as fast as he could and helped me to understand how the simulator works and how to improve the schedulers that I was implementing.

Also, I want to thank Ramon Ferrús, my supervisor in UPC. He accepted to be my supervisor when I proposed it to him. He helped me when I need it and he reviewed my documents to see if I could improve them before I attached them to the intranet.

Finally, as I commented before in dedication, I would like to point the help that I received from my Erasmus partner, Alex Alujas. We shared our lives for a month and we help each other when we need it. It was a pleasure to have this kind of company.

Revision history and approval record

| Revision | Date | Purpose |
|----------|------------|-------------------|
| 0 | 25/05/2020 | Document creation |
| 1 | 14/06/2020 | Document revision |
| | | |
| | | |
| | | |

DOCUMENT DISTRIBUTION LIST

| Name | e-mail |
|-------------------|-------------------------------|
| Marc Lechado Puig | marclechadopuig1997@gmail.com |
| Stefan Schwarz | sschwarz@nt.tuwien.ac.at |
| Ramon Ferrús | ferrus@tsc.upc.edu |
| | |
| | |
| | |

| Written by: | | Reviewed and approved by: | |
|-------------|-------------------|---------------------------|--------------------|
| Date | 25/05/2020 | Date | 15/06/2020 |
| Name | Marc Lechado Puig | Name | Stefan Schwarz |
| Position | Project Author | Position | Project Supervisor |

Table of contents

| | |
|--|----|
| Abstract | 1 |
| Resum | 2 |
| Resumen | 3 |
| Dedication | 4 |
| Acknowledgements | 5 |
| Revision history and approval record | 6 |
| Table of contents | 7 |
| List of Figures | 9 |
| List of Tables: | 10 |
| 1. Introduction..... | 11 |
| 1.1. Statement of purpose | 11 |
| 1.2. Requirements | 11 |
| 1.3. Methods and procedures | 12 |
| 1.4. Work Plan..... | 12 |
| 1.4.1. Work Packages | 12 |
| 1.4.2. Time Plan (Gantt Diagram)..... | 15 |
| 1.5. Incidences and Work Plan modifications | 16 |
| 2. State of the art of the technology used or applied in this thesis:..... | 17 |
| 2.1. 5G System Level Simulator | 17 |
| 2.2. 5G NR | 19 |
| 2.2.1. Introduction | 19 |
| 2.2.2. 5G vs LTE | 20 |
| 3. Methodology / project development: | 22 |
| 3.1. Simulator description | 22 |
| 3.2. Traffic Model implementation..... | 23 |
| 3.3. Scheduling strategies | 24 |
| 3.3.1. Scheduling strategy 1: buffer status..... | 25 |
| 3.3.2. Scheduling strategy 2: latency threshold | 26 |
| 3.3.3. Scheduling strategy 3: best CQI | 27 |
| 4. Results | 29 |
| 4.1. Scheduling strategy 1: buffer status..... | 29 |
| 4.2. Scheduling strategy 2: latency threshold | 30 |
| 4.3. Scheduling strategy 3: best CQI | 32 |

| | |
|---|----|
| 4.4. Comparison between schedulers..... | 33 |
| 4.5. Comparison between schedulers changing parameters..... | 34 |
| 5. Budget..... | 39 |
| 6. Conclusions and future development:..... | 40 |
| Bibliography:..... | 41 |
| Glossary | 42 |

List of Figures

| | Page |
|---|------|
| Figure 1. Gantt Diagram | 15 |
| Figure 2. Principal features of the simulator 1 | 18 |
| Figure 3. Principal features of the simulator 2 | 19 |
| Figure 4. 5G frequency ranges | 20 |
| Figure 5. The Landscape of 5G | 21 |
| Figure 6. Simulator's structure | 22 |
| Figure 7. User Allocation development for the first scheduling strategy 1 | 25 |
| Figure 8. User Allocation development for the first scheduling strategy 2 | 26 |
| Figure 9. Timer transformation | 26 |
| Figure 10. Pos transformation | 27 |
| Figure 11. RBs assigned | 27 |
| Figure 12. Best CQI development | 28 |
| Figure 13. Scheduler best CQI: RBs assigned | 28 |
| Figure 14. Transmission latency scheduling strategy 1 – PedA | 29 |
| Figure 15. Transmission latency scheduling strategy 1 – VehA | 30 |
| Figure 16. Transmission latency scheduling strategy 2 – PedA | 30 |
| Figure 17. Transmission latency scheduling strategy 2 – VehA | 31 |
| Figure 18. Transmission latency scheduling strategy 3 – PedA | 32 |
| Figure 19. Transmission latency scheduling strategy 3 – VehA | 32 |
| Figure 20. Transmission Latency Comparison – PedA | 33 |
| Figure 21. Transmission Latency Comparison – VehA | 33 |
| Figure 22. Transmission Latency Comparison - Deltat = 2 | 34 |
| Figure 23. Transmission Latency Comparison - Deltat = 4 | 35 |
| Figure 24. Transmission Latency Comparison - Deltat = 8 | 36 |
| Figure 25. Transmission Latency Comparison - Packet Size = 750 | 36 |
| Figure 26. Transmission Latency Comparison - Packet Size = 1500 | 37 |
| Figure 27. Transmission Latency Comparison - Packet Size = 3000 | 38 |
| Figure 28. Transmission Latency Comparison | 38 |

List of Tables:

| | Page |
|-------------------------|------|
| Table 1. Work Package 1 | 12 |
| Table 2. Work Package 2 | 13 |
| Table 3. Work Package 3 | 13 |
| Table 4. Work Package 4 | 14 |
| Table 5. Work Package 5 | 14 |
| Table 6. Gantt Diagram | 15 |
| Table 7. Budget | 39 |

1. Introduction

1.1. Statement of purpose

The purpose of this project is the implementation of traffic models for different types of applications and the development of Quality-of-Service aware resource schedulers.

This project consists on modifying the Matlab-based 5G System Level Simulator by adding different traffic models and adding some new scheduler based on schedulers that are already implemented in the simulator to get better results in terms of latency and QoS.

The project main goals are:

- Understand how the 5G System Level Simulator works
- Implementation of different types of traffic models
- Develop scheduling strategies
- Compare the developed scheduling strategies to standard state-of-the-art-schedulers in terms of latency

1.2. Requirements

Project requirements:

- Implement the traffic models to the 5G System Level Simulator
- Satisfy the Quality-of-Service requirements of as many users/applications as possible
- Develop scheduling strategies that account for the QoS requirements and trade off the needs of different applications

1.3. Methods and procedures

This work has been developed in a simulator previously developed by professors and PhD and master students of the TU Wien. In this university there are developing two 5G simulators, the System Level Simulator and the Link Level Simulator. For my thesis I've been using the System Level Simulator.

These simulators are based on Matlab software, therefore all the implementations that I have done for my project, as the implementation of traffic models or the implementation of new schedulers are also based on Matlab.

The project, that is carried out at Technische Universität Wien, consists in adding a new feature to the 5G System Level Simulator.

1.4. Work Plan

1.4.1. Work Packages

| | | |
|---|--|--------------------------|
| Project: Literature Review | WP ref: 1 | |
| Major constituent: Reading | Sheet 1 of 1 | |
| Short description: Reading documents about the standard state-of-art-schedulers and manuals about the Vienna 5G System Level Simulator. | Planned start date: 19/02/2020 Planned end date: 26/02/2020 | |
| | Start event: End event: | |
| Internal task T1: Understand how the simulator works | Deliverables: | Dates: |
| | M2M_traffic | 19/02/2020 26/02/2020 |
| <i>Table 1. Work Package 1</i> | | |

| | | |
|---|---|------------------------------------|
| Project: Traffic Models | WP ref: 2 | |
| Major constituent: Software (programming with Matlab) | Sheet 1 of 1 | |
| Short description: Implement different types of traffic models using Matlab to have an interesting traffic mix for the scheduler. | Planned start date: 24/02/2020 Planned end date: 30/04/2020 | |
| | Start event: End of WP1 End event: Traffic model implemented working as it is expected | |
| Internal task T1: Implement traffic models to the simulator Internal task T2: Assign RBs to each user Internal task T3: Calculate the latency of each packet Internal task T4: Plot the buffer evolution in time and all the latencies | Deliverables: Traffic models implemented in the simulator | Dates: 24/02/2020 20/04/2020 |
| <i>Table 2. Work Package 2</i> | | |

| | | |
|---|---|------------------------------------|
| Project: Scheduling strategies | WP ref: 3 | |
| Major constituent: SW and simulation | Sheet 1 of 1 | |
| Short description: Develop scheduling strategies that account for the Quality-of-Service requirements and trade off the needs of different applications. | Planned start date: 01/05/2020 Planned end date: 20/05/2020 | |
| | Start event: End of WP2 End event: Scheduler works as it is expected | |
| Internal task T1: Develop scheduling strategies Internal task T2: Minimize the amount of wasted resources. Internal task T3: Implement a scheduler that attempts to assign as many RBs as are required to transmit all the data in the buffer of a user | Deliverables: Simulation results | Dates: 21/04/2020 18/05/2020 |
| <i>Table 3. Work Package 3</i> | | |

| | | |
|--|---|------------------------------------|
| Project: Compare values | WP ref: 4 | |
| Major constituent: Simulation | Sheet 1 of 1 | |
| Short description: Comparison of the developed scheduling strategies to standard state-of-art schedulers (round robin, proportional fair) in terms of throughput, latency, QoS satisfaction. | Planned start date: 20/05/2020 Planned end date: 31/05/2020 | |
| | Start event: End of first scheduler implemented End event: Review of latency results | |
| Internal task T1: Compare the developed scheduling strategies to standard state-of-art-schedulers | Deliverables: Figures comparing the values | Dates: 10/05/2020 25/05/2020 |
| <i>Table 4. Work Package 4</i> | | |

| | | |
|--|--|------------------------------------|
| Project: Memory writing | WP ref: 5 | |
| Major constituent: Writing | Sheet 1 of 1 | |
| Short description: Collect all the information obtained and write the final memory of the project. | Planned start date: 01/06/2020 Planned end date: 20/06/2020 | |
| | Start event: End of WP4 End event: Memory finished | |
| Internal task T1: Collect information Internal task T2: Write the memory | Deliverables: Bachelor thesis | Dates: 26/05/2020 03/06/2020 |
| <i>Table 5. Work Package 5</i> | | |

1.4.2. Time Plan (Gantt Diagram)

| TASK NAME | START DATE | END DATE | DURATION (days) |
|-------------------------------|------------|------------|-----------------|
| Literature Review | 19/02/2020 | 26/02/2020 | 7 |
| Implement traffic models | 24/02/2020 | 30/04/2020 | 66 |
| Develop scheduling strategies | 01/05/2020 | 20/05/2020 | 19 |
| Compare values | 20/05/2020 | 31/05/2020 | 11 |
| Memory writing | 01/06/2020 | 20/06/2020 | 19 |

Table 6. Gantt Diagram

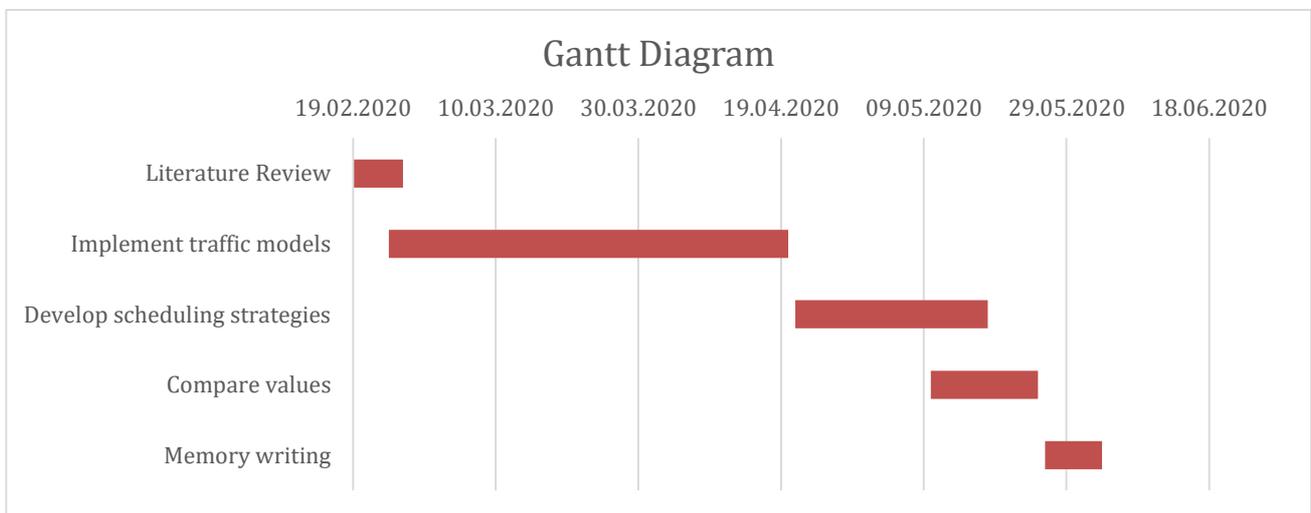


Figure 1. Gantt Diagram

1.5. Incidences and Work Plan modifications

During all the project there has been many incidences. All these incidences were about programming, because a new functionality has been implemented to the 5G System Level Simulator.

These incidences are based on trying to run some codes that didn't work and it need to be changed or the results obtained were not right or were not what we expected.

About the Work Plan, there are many modifications from the proposed plan. The main problem was implementing the traffic models. It required more time than expected, specifically, one month more. It changed all the proposed plan.

For the next tasks, as develop scheduling strategies and compare values, it required less time, in days, than expected, because I dedicated more hours each day.

2. State of the art of the technology used or applied in this thesis:

2.1. 5G System Level Simulator

In cellular communications, simulations are an inevitable tool for understanding the mutual interactions of all involved players in the network. Especially for gaining insight in the performance of a large-scale scenario, a real-world measurement approach becomes too costly and laborious.

The 5G System Level Simulator tackle the need for simulating large scale networks, capturing the change in network layouts and physical transmission, coming up with the expected 5G standard. While the 5G standard has not yet been fully specified as of this writing, it is commonly agreed upon that future networks will become more heterogeneous. Therefore, this simulator allows to create networks of arbitrary layout with several tiers of Base Stations (BSs) and various user types in the same simulation.

The simulator average over many spatial constellations and channel realizations and thus obtain results for average throughput per user/BS, average Signal to Interference and Noise Ratio (SINR) performance and ratio of successful transmissions. To determine the quality of each individual link, the instantaneous SINR is evaluated. For each transmission, the received power of all transmitters (desired and interfering) is calculated by combining distance dependent path loss, channel realization, antenna pattern and shadowing. It is possible to choose from several models and options for each of these individual propagation effects. Additionally, this is not a static choice that is set for the whole simulation, but is chosen dependent on the link conditions (e.g., Line-Of-Sight (LOS)/Non Line of Sight (NLOS)). These link-types can again be distinguished by different means.

The current version of the 5G System Level Simulator supports heterogeneous networks with an arbitrary number of BS tiers and user types, including mobile users. Thanks to the construction of the BSs objects with attached antenna objects, BSs with Remote Radio Heads (RRHs) and Distributed Antennas Systems (DASs) are available for simulations.

Regarding the network geometry, not only BSs and users can be placed, but also 3-D blockages, resembling walls and buildings. Consequently, randomly generated cities can be created, such as a Manhattan grid layout or randomly placed buildings with arbitrary orientation. The new transmission features of 5G, such as mmWave and massive Multiple-Input Multiple-Output (MIMO) are represented in the simulator by the corresponding channel model for the right frequency range.

The following figures presents a summary of the principal features of the simulator.

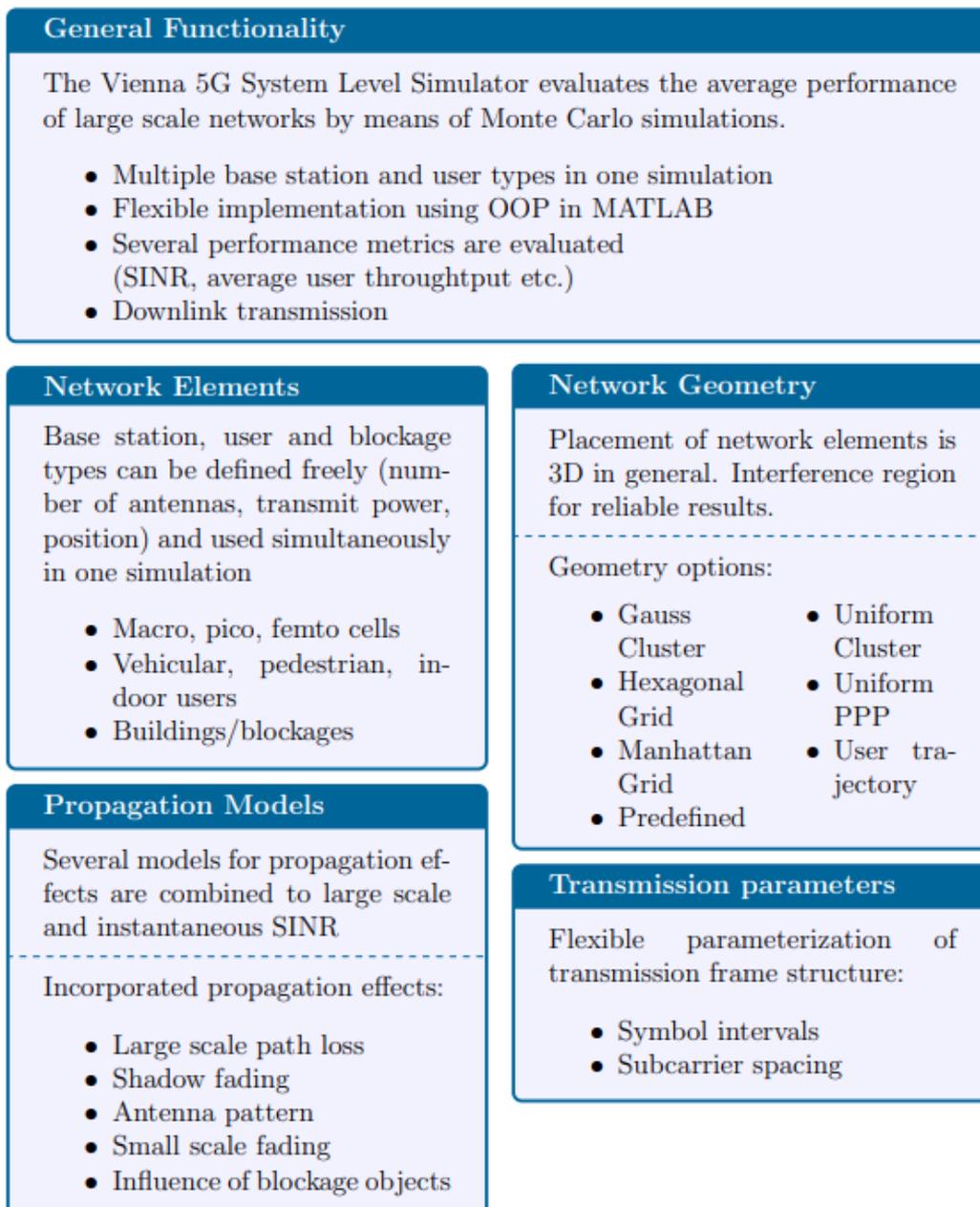


Figure 2. Principal features of the simulator 1. Ref.: Vienna 5G System Level Simulator v1.0 – List of Features

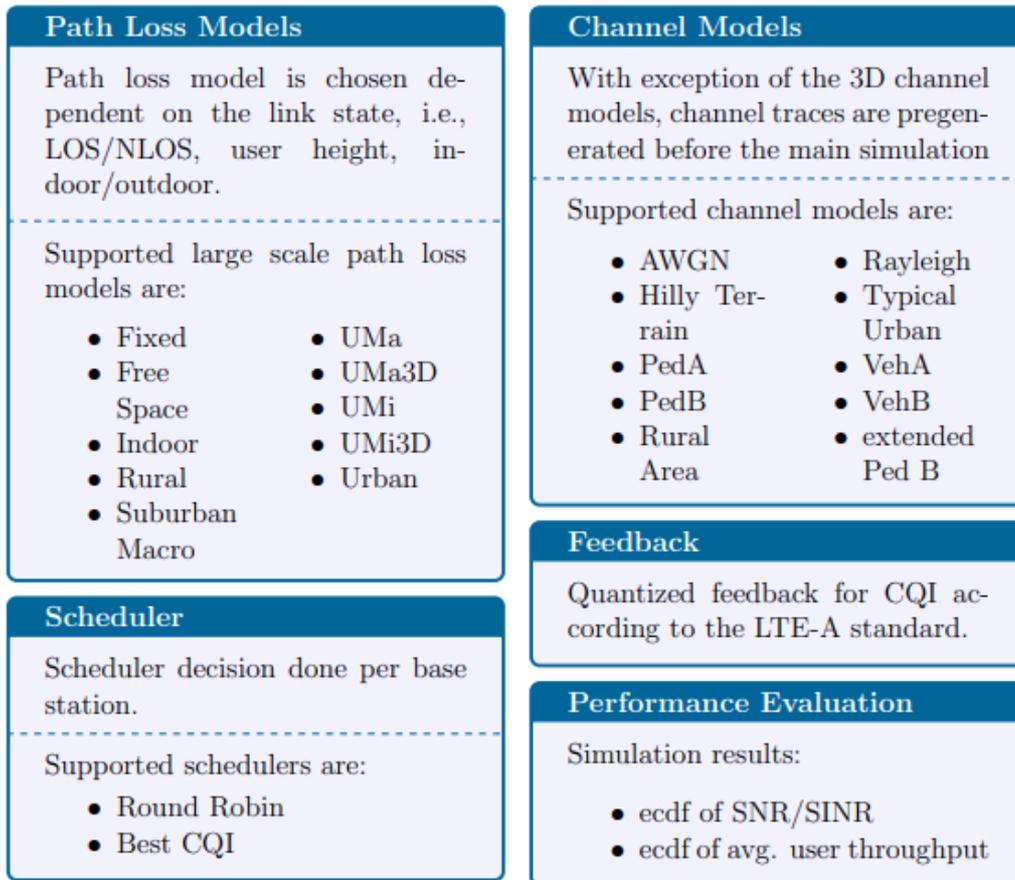


Figure 3. Principal features of the simulator 2. Ref.: Vienna 5G System Level Simulator v1.0 – List of Features

2.2. 5G NR

2.2.1. Introduction

5G NR (New Radio) is a new radio access technology (RAT) developed by 3GPP. It is a new air interface being developed for 5G. This technology is being developed improving other technologies like LTE, LTE-Advanced and LTE Pro technologies.

It includes two frequency ranges: FR1, which operates below 6 GHz, and FR2, that includes bands above 24 – 100 GHz.

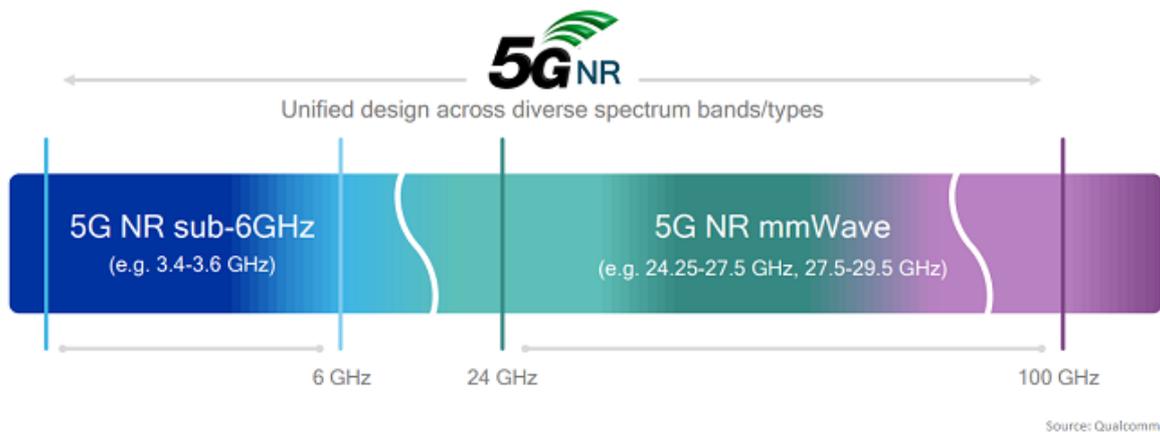


Figure 4. 5G frequency ranges. Ref.: <https://www.rfpage.com/what-are-5g-frequency-bands/>

5G uses a waveform modulation technique also used by LTE and IEEE 802.11 (Wi-Fi), and it is OFDM. It makes 5G the first mobile generation that is not based on a new waveform and multiple access design. Another important element in this technology is a flexible framework to enable efficient multiplexing of diverse 5G services. Thanks to that, latencies will be far lower than current LTE networks.

3GPP has defined three broad areas of use for 5G NR:

- Enhanced Mobile Broadband (eMBB) for intensive applications like video streaming or immersive gaming.
- Ultra-reliable and Low-Latency Communications (uRLLC) for critical applications and services like command and control functions in autonomous driving and remote control in healthcare and manufacturing services.
- Massive Machine Type Communications (mMTC) supporting massive IoT, connecting low cost, low energy devices with small data volumes on a mass scale, such as smart cities.

2.2.2. 5G vs LTE

5G technology is based on LTE technology, so the new network will have better features than actually. The new interface will be able to have connected more devices in the same geographic zone, specifically, around a million of devices per square kilometre. Nevertheless, with LTE technology, just 4000 devices could be connected. With this improvement, the problem with technical limits of the amount of data that can be transferred across the frequency spectrum will be solved.

Using the new technology called Massive MIMO, that is based in multiple directed nodes to focus and follow users around cells, coverage, speed and capacity will be also improved.

In terms of latency, that is the time that the data takes to go between your device and the server you want to access, there will be a great progress. Nowadays, 4G networks have an average latency of 50 ms approximately, but with 5G networks will be possible have latencies below than 10 ms that can reach 1 ms of delay in the best cases.



Figure 5. The Landscape of 5G. Ref.: <https://es.finance.yahoo.com/noticias/comparamos-red-5g-vs-lte-165245750.html>

3. Methodology / project development:

3.1. Simulator description

The Vienna 5G SL Simulator is written in MATLAB and is utilizing OOP. Individual parts of the simulator are separated into different packages and, e.g., network elements, such as BSs, are defined in classes. In general, the simulator is written in a modular fashion, such that new functions can be added easily, without the need to alter other parts of the code.

The simulator's structure is defined by four major parts, which are displayed in Fig. 6.

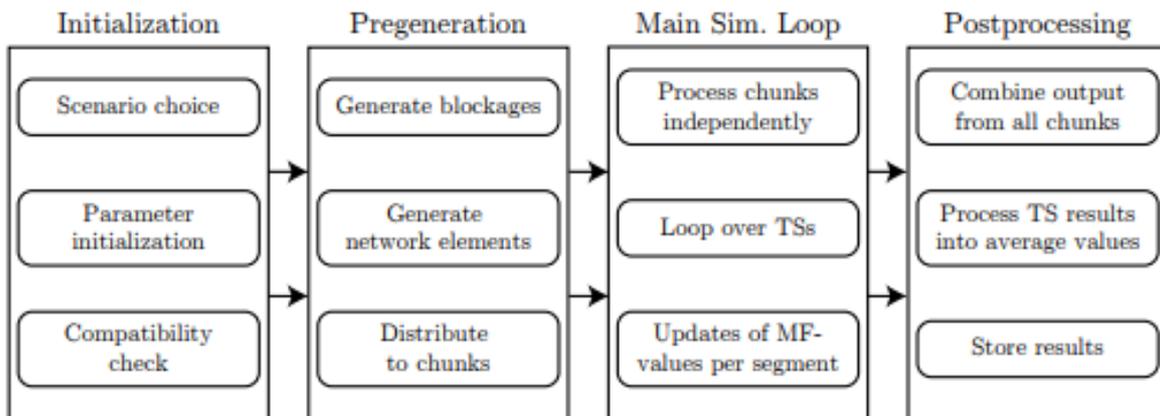


Figure 6. Simulator's structure. Ref.: User Manual. The Vienna 5G System Level Simulator. Institute of Telecommunications, TU Wien

To start with, a scenario has to be chosen. In the scenario file there are the main parameters declared like the transmission parameters and the configuration of the network elements (base station and user parameters). There are several scenarios as hexagonal grid with interference ring, it is an example of how to simulate a hexagonal grid network of three-sector BSs with a defined number of rings of BSs and rings of interfering BSs or the Manhattan city layout, that it is an example of how to simulate buildings and streets arranged according to a Manhattan grid with BSs placed on the rooftop of buildings and users distributed along the streets. For the project the basic scenario is used, that configures the simulation of a simple simulation. Once the scenario is chosen, the launcher file needs to be run.

The simulation starts and the simulation object is created and the predefined parameters are attached to this object. After that, the simulation time line is generated with the network element objects. The following line contains the main simulation loop. It contains a loop over chunks, whereas each chunk contains a loop over time slots. When the main simulation loop is finished, the simulator combines the output from all chunks, process TS results into average values and save the results into a variable called result.

Finally, there are plotted different figures like the empirical cumulative distribution function (ecdf) of the average user throughput, the user SINR, the user BLER and a map that shows how the BSs and users are distributed in the scenario.

3.2. Traffic Model implementation

The first task of the project is the implementation of traffic models for different types of applications. The first traffic model implemented is the simplest one. It has different properties, the number of packets that want to be generated, the packet size, the maximum time of duration of the simulation, the buffer, the time between packets that is called Δt and the time when a packet should be generated. This traffic model is based on generating packets for each user and update the buffer when the bits of every packet are sent. It has two functions: `checkPacket` and `updateBuffer`. The function `checkPacket` checks if the packet has been generated or not. If it has been generated, the buffer, that it is an empty vector, starts to being filled with the bits of the packet size parameter. The function `updateBuffer` updates the buffer removing the bits that have been sent. To run this code a launcher is needed. In this script the main traffic model properties and the traffic model object are initialized and the functions are called. Finally, what it does is plot a figure representing the buffer evolution in time.

The second traffic model implemented is an improvement of the first one but it is focused in the latency property. It has the same properties than before but it also includes the time when a packet is fully sent, the latency, a latency threshold and the vector generation time that will save the time when a packet is generated.

It has been implemented with four functions, `setTMparams`, `checkPacket`, `updateBuffer` and `calculateLatency`. In `setTMparams` the traffic model properties are initialized. The base of `checkPacket` function is the same than in the first traffic model but now when a packet is generated it saves this time in the vector `generationTime`. Later, the code checks if the buffer is empty or not, and if it is not it compares the time when the packet was generated with the current time. If it is larger than the latency threshold the packet is thrown away due to the information is not useful any longer. To make it possible what is done is equalise the buffer to zero to know that the packet has been thrown away and the time when this packet is fully sent is equalised to infinite. In `updateBuffer` function the program checks if the buffer is empty, if it is not it updates the buffer removing the bits that have been sent and it checks the buffer status again. If the buffer is empty once it was updated, it saves the instant when the packet has been fully sent. The last function, `calculateLatency`, calculates the latency of each packet taking into account the time when the packet has been generated and the time when the packet has been fully sent. To test this code an example script was programmed where the parameters and the traffic model object were initialized and the functions were called. At the end, two figures are plotted, the first one shows the buffer status evolution in time and the second represents the latency of each packet. Once it worked, it was implemented to the 5G SL Simulator. Finally, there were implemented two functions in class `ResultsSuperclass.m` that consists in plotting the data sequence of the number of bits that could not be sent to know the buffer status and in plotting an ecdf of the latencies.

3.3. Scheduling strategies

In the thesis have been developed three different scheduling strategies based on round robin scheduler. This scheduler consists on add newly arriving users at the end of the queue and it assigns the resource blocks (RBs) to these users that are at the front of the queue. Those users will then be placed at the end of the queue. It is formed by five functions:

- `scheduledL`: it is called for every simulation slot and allocates users and transmit power to RBs.
- `addUsersDL`: it is called whenever a user connects to the attached base station. It adds users to the scheduler queue.

- `removeUsersDL`: it is called whenever a user disconnects from the attached BS. It removes users from the scheduler queue.
- `scheduleDLCommon`: performs all calculations of the scheduler like the CQI for all RBs that are scheduled for one user.
- `updateAttachedUsersDL`: takes a list of users that should be attached to the BS/Scheduler.

To develop the different scheduling strategies, function `scheduleDL` has been modified to assign RBs in the way it wants to be performed.

3.3.1. Scheduling strategy 1: buffer status

For the first scheduling strategy, what have been done is taking into account the buffer status. First of all, function `checkPacket` is called to generate packets for users that are waiting in the queue. After that, the program checks for each user if there is data in the buffer or not. If it is not empty, this user is considered in the RB allocation. RB allocation is a 100x2 matrix that at the beginning all their values are -1. Once a user is considered in the RB allocation, the values of rows are changed from -1 to the id value of the user.

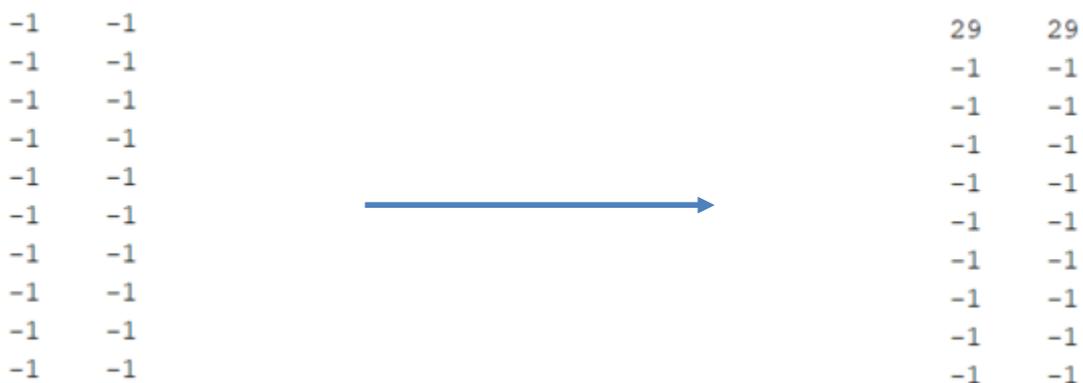


Figure 7. User Allocation development for the first scheduling strategy 1

This process is repeated until all the RB allocation is filled with all users that have some data in their buffers.

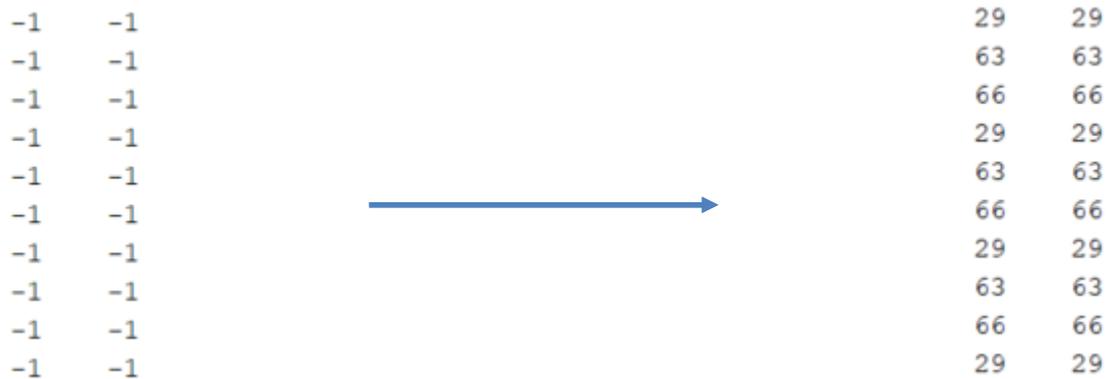


Figure 8. User Allocation development for the first scheduling strategy 2

This scheduler is just focused in the buffer status, when a user has some data in his buffer, RB are assigned to this user. The next two schedulers are based on this one, both schedulers take into account the buffer status but also have some improvement.

3.3.2. Scheduling strategy 2: latency threshold

This scheduler is based on looking at how long the packets of each user are already in the buffer, to assign RBs with priority to those users whose packets are already approaching to the parameter latency threshold. In this way, fewer packets should be thrown away.

To start with, function checkPacket is called to generate packets for users that are waiting in the queue. Moreover, two empty vectors are declared whose are called timer and pos. After that, the program checks for each user if there is data in the buffer or not. If the user has the buffer not empty, the vectors start being filled. Timer is filled with the time remaining for each packet of each user to reach the latency threshold, and vector pos is filled with the position that each user takes up in the queue. Later, timer is sorted in a descending way to give priority to these packets that are approaching to the deadline and pos is sorted depending in how timer was sorted to know in which order the RBs have to be allocated.



Figure 9. Timer transformation

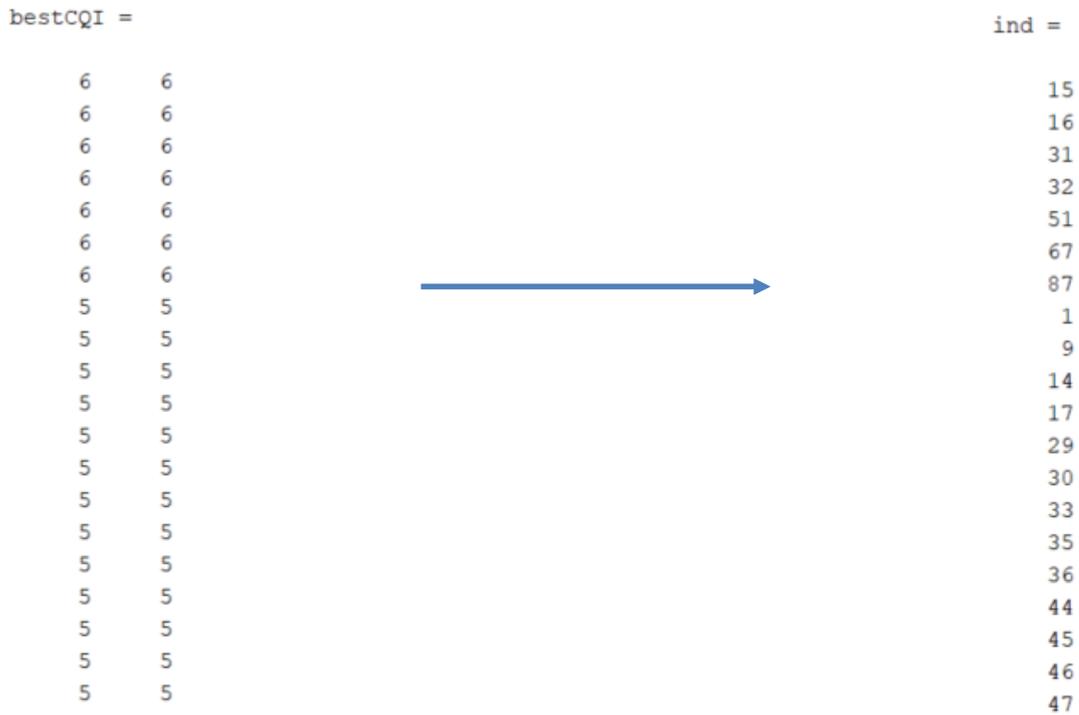


Figure 12. Best CQI development

In this example there are the best twenty channels of a user, with an id equal to 9, with their allocation. And the RBs would be allocated in the user allocation matrix in the next way.

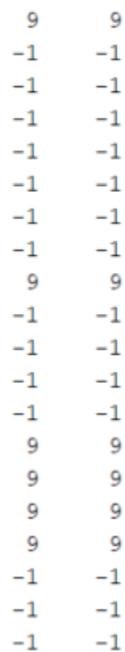


Figure 13. Scheduler best CQI: RBs assigned

This process is repeated for all the users until the user allocation matrix is filled.

4. Results

To test the performance of all the schedulers, there have been made twenty simulations per scheduler and for each of them there have been made the simulations in two different channel mode type, in Pedestrian mode and Vehicular mode. The results are based on study how good are the latencies for each scheduler depending in the channel mode the simulator is working on and it is also based on study which percentage packets are thrown away. To make it possible, it is compared plotting ecdf figures of the latency values.

4.1. Scheduling strategy 1: buffer status

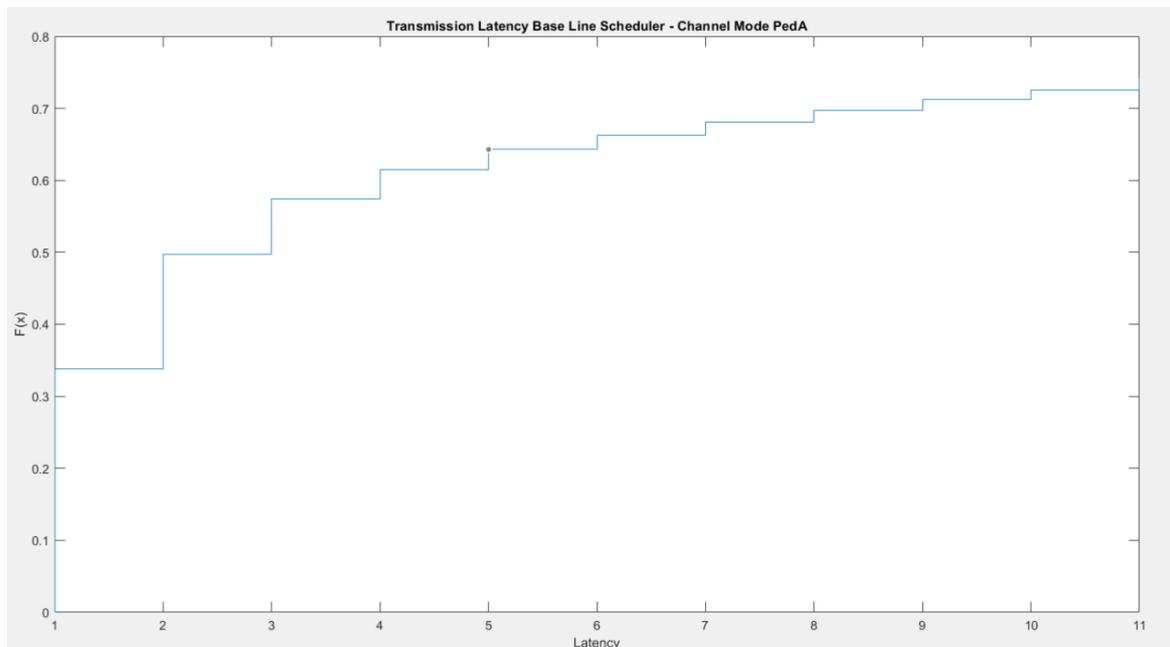


Figure 14. Transmission latency scheduling strategy 1 - PedA

For the first scheduler with channel mode Pedestrian can be observed that most of the 50% of latency values are concentrated in low values. As it can be seen, there were a 74% of packets that have been fully sent, and just a 26% were thrown away.

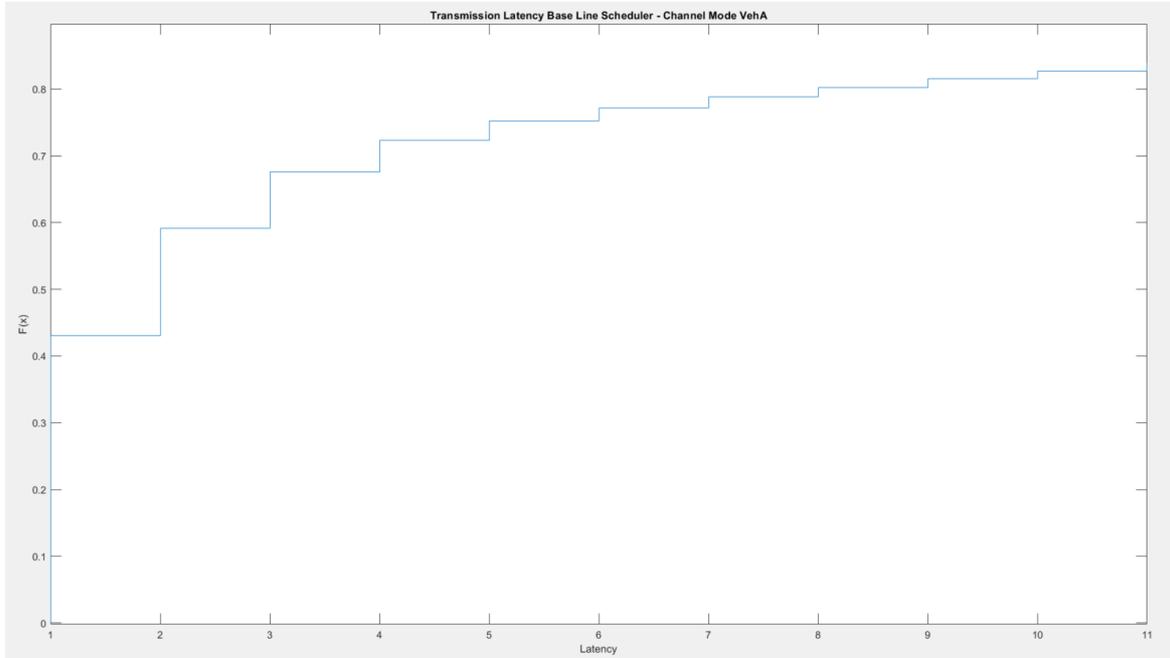


Figure 15. Transmission latency scheduling strategy 1 - VehA

For the channel mode Vehicular the results are better than the Pedestrian mode. Using this channel mode there is reached more than a 70% of packets sent for low latency values. In addition, almost an 84% of the packets are fully sent, so just a 16% of them are thrown away.

4.2. Scheduling strategy 2: latency threshold

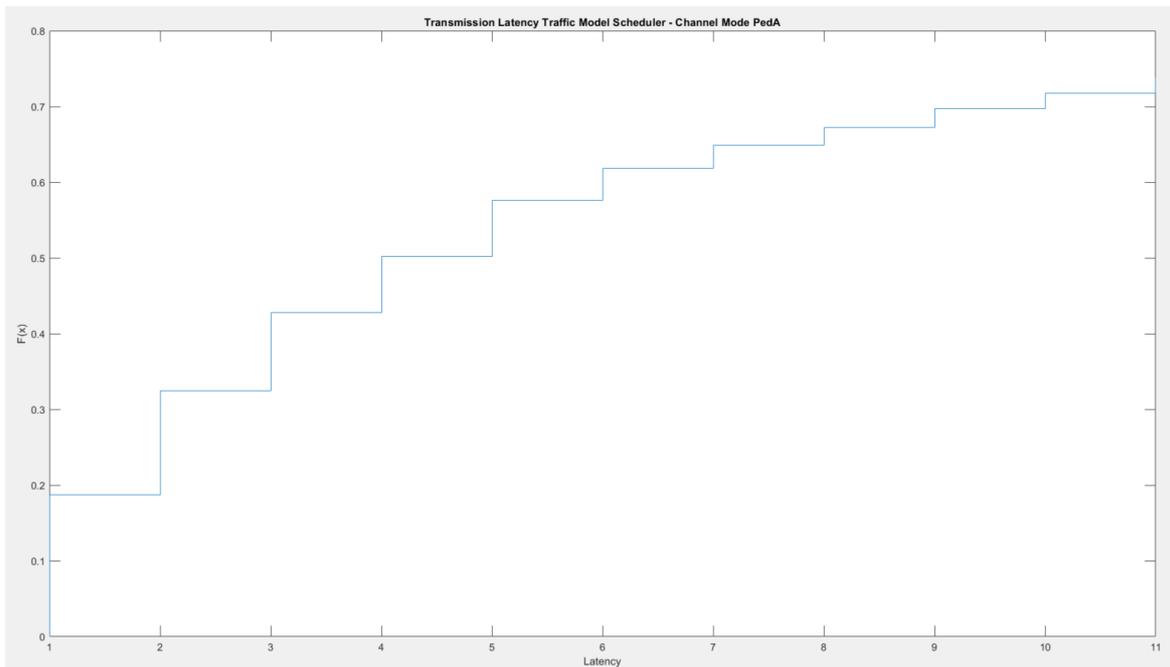


Figure 16. Transmission latency scheduling strategy 2 - PedA

For the scheduler that is based on assign RBs to these users that their packets are approaching to the latency threshold the results are not as good as expected. For the Pedestrian mode, most of the packets are sent with high latency values, which means that the packets take too much time to be sent. In terms of packets that are thrown away, almost a 26% of the packets are wasted.

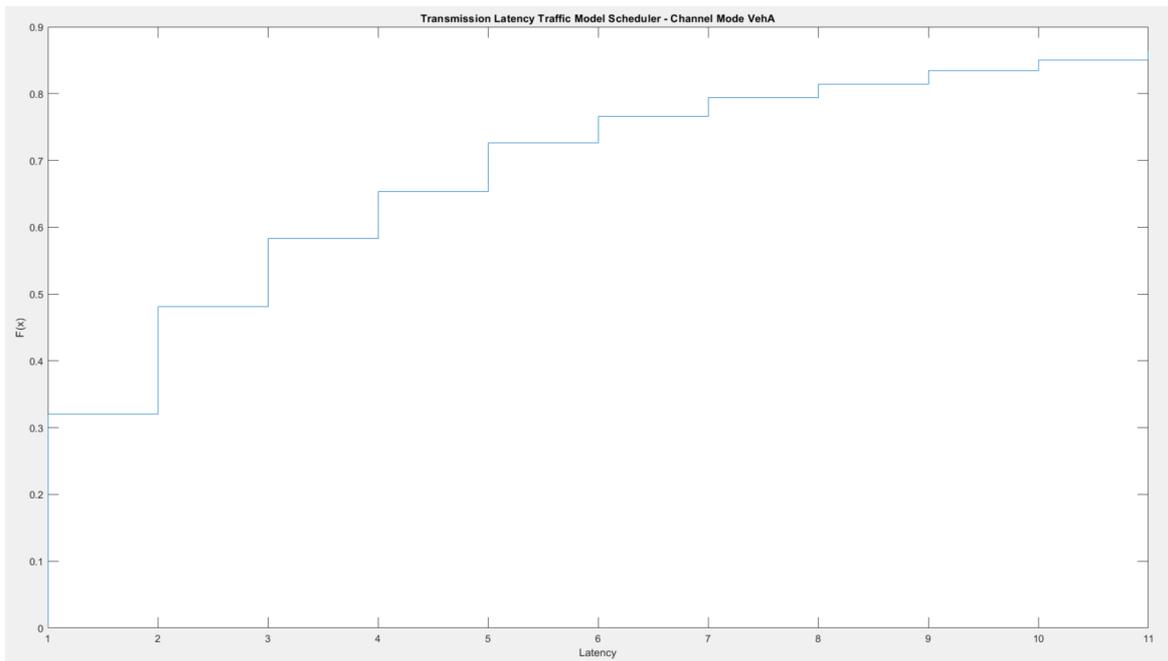


Figure 17. Transmission latency scheduling strategy 2 - VehA

Using the Vehicular channel mode, in terms of latency, the values are better than in the Pedestrian mode, but they are not what it would have to be, they keep being too bad, and just a 14% of the packets were thrown away.

In general, the results for this scheduler were not what we expected and one of the reasons of having these results would be that we are wasting too much resource blocks for users that have too poor channel quality and can therefore not transmit reliably.

4.3. Scheduling strategy 3: best CQI

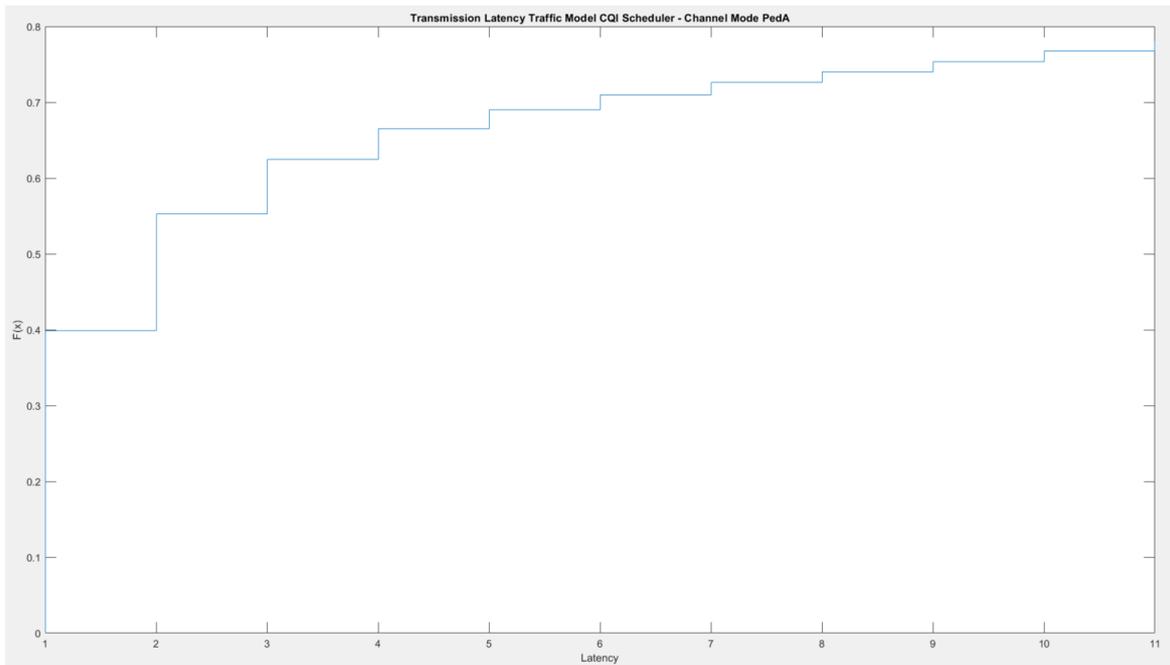


Figure 18. Transmission latency scheduling strategy 3 - PedA

As it can be seen, running the simulator with the scheduler that is based on the best CQI and with channel mode Pedestrian type, the results are really good, getting a high percentage of packets fully sent with low latency values and getting a 22% of packets lost.

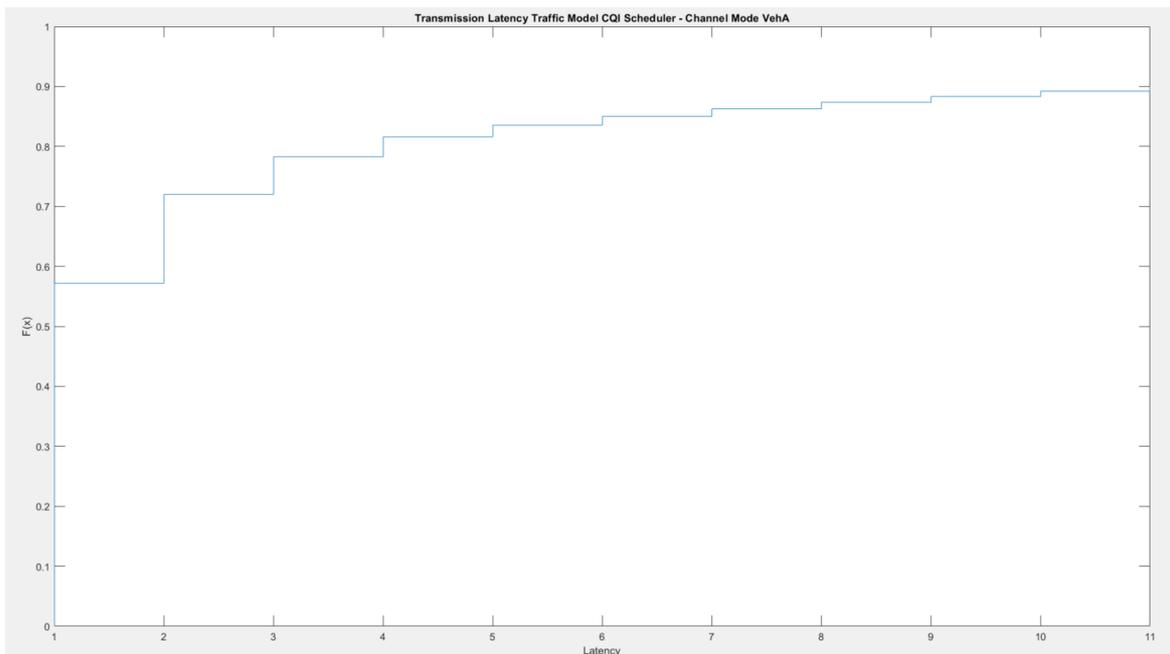


Figure 19. Transmission latency scheduling strategy 3 - VehA

When the channel mode is the Vehicular type the results are much better than using Pedestrian type. Using it, almost an 80% of the packets are sent with latency values equal to 3 or less and there is just a 10% of wasted packets.

4.4. Comparison between schedulers

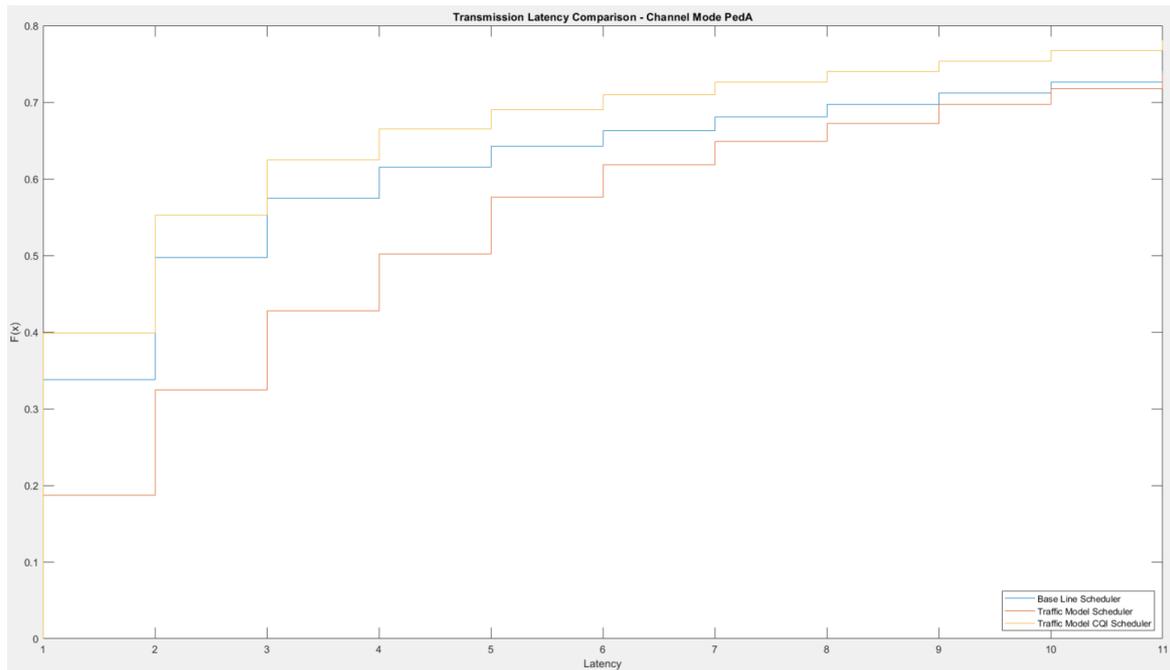


Figure 20. Transmission Latency Comparison - PedA

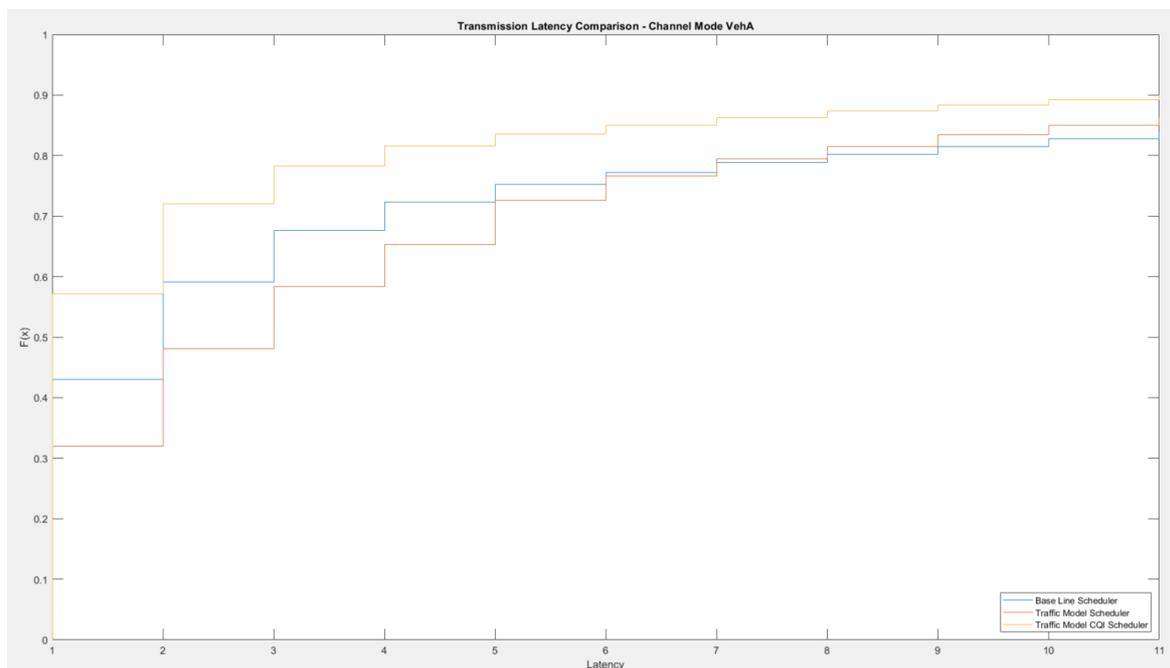


Figure 21. Transmission Latency Comparison - VehA

As it can be observed, for both channel mode type, Pedestrian and Vehicular, the best results are obtained using the third scheduler strategy, whose is based on the best CQI. Therefore, the channel quality is a property that has to be taken into account and give to it the importance that it deserves. Nevertheless, and as it is commented before, the results obtained using the second scheduler were not what we expected because the channel quality has not been taken into account.

In general, the number of packets wasted are so similar for each channel mode type. For Pedestrian channel mode type it varies between 22% and 26% and for Vehicular mode it varies between 10% and 16%.

4.5. Comparison between schedulers changing parameters

To investigate how the parameters of the traffic model impact the performance of the system, simulations changing Δt and packet size have been done. It consists in do simulations changing one of the parameters and keep a fixed value for the other parameter to see how the first property affects to the system. And after that, change the role to the parameters.

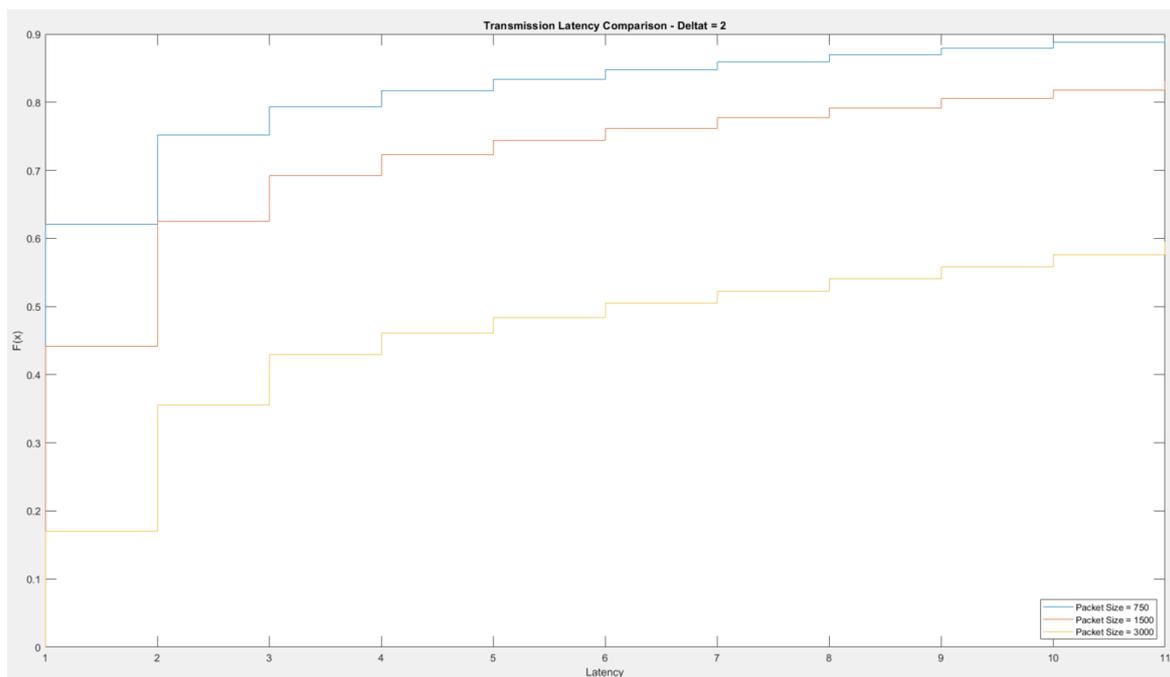


Figure 22. Transmission Latency Comparison - Deltat = 2

With $\text{deltat} = 2$, which means that every two instants a packet is generated, there is observed that we obtain better results as fewer as the packet is. When the packet size is 750 bits, most of the packets are fully sent in low latency values and there is also obtained a low percentage of wasted packet, specifically, around a 10%. Nevertheless, with a packet size of 1500 and 3000 bits this percentage is quite higher, 27% and 40% respectively.

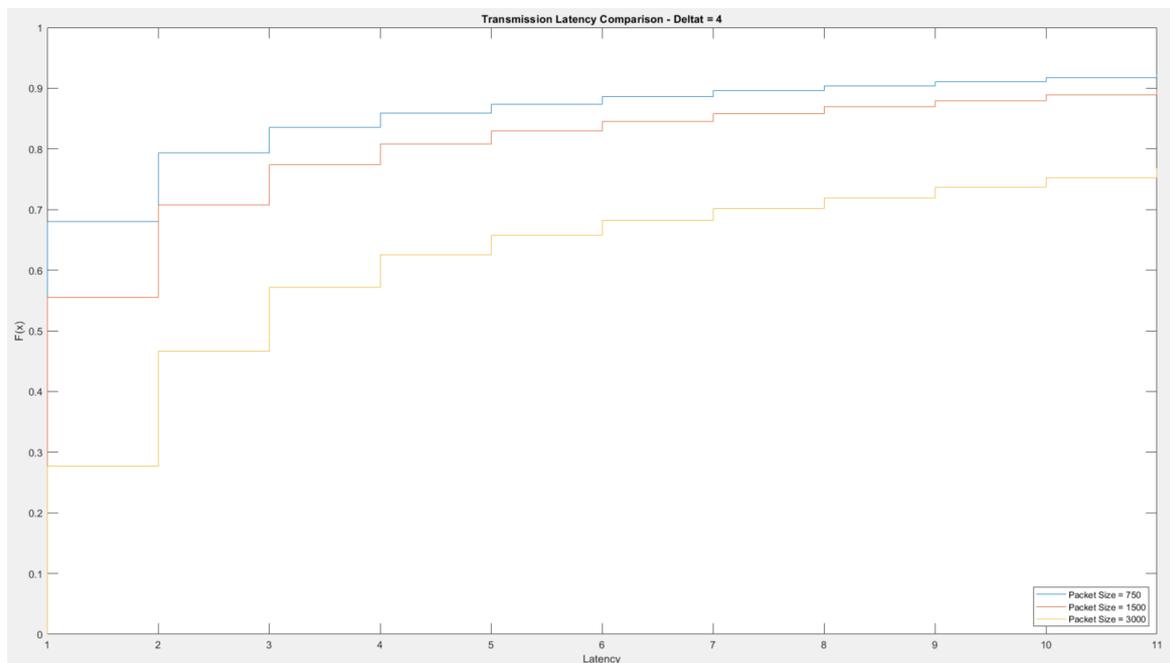


Figure 23. Transmission Latency Comparison - Deltat = 4

With $\text{deltat} = 4$ the comparison between the different packet size follows the same line than with $\text{deltat} = 2$, but now the latency values are more centred in low values and the percentage of wasted packets are lower. That is due to there are more time between the generation of each packet, hence there is more time to fully send them. In this case, the percentage of packets that have been thrown for packet size 750, 1500 and 3000 bits are 8%, 10% and 23% respectively.

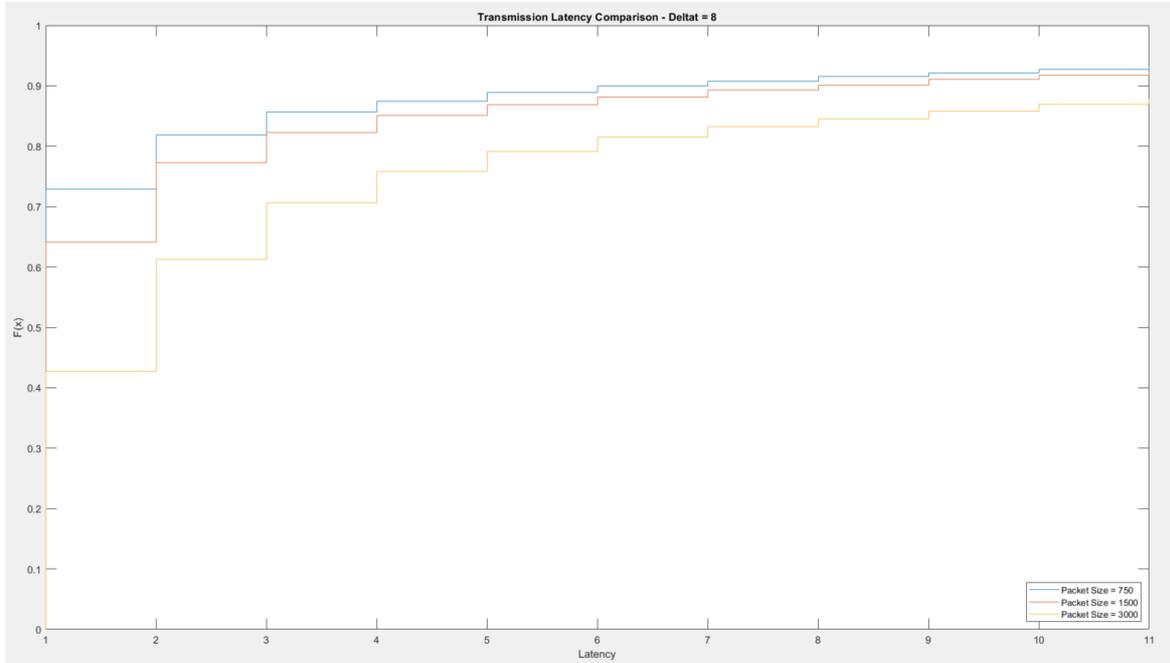


Figure 24. Transmission Latency Comparison - Deltat = 8

With $\text{deltat} = 8$, as it can be seen the latency values keep being concentrated in low values for all the cases and about the percentage of lost packets the value is very similar for the different packet sizes. These percentages are 7%, 8% and 12% for packet size equal to 750, 1500 and 3000 bits.

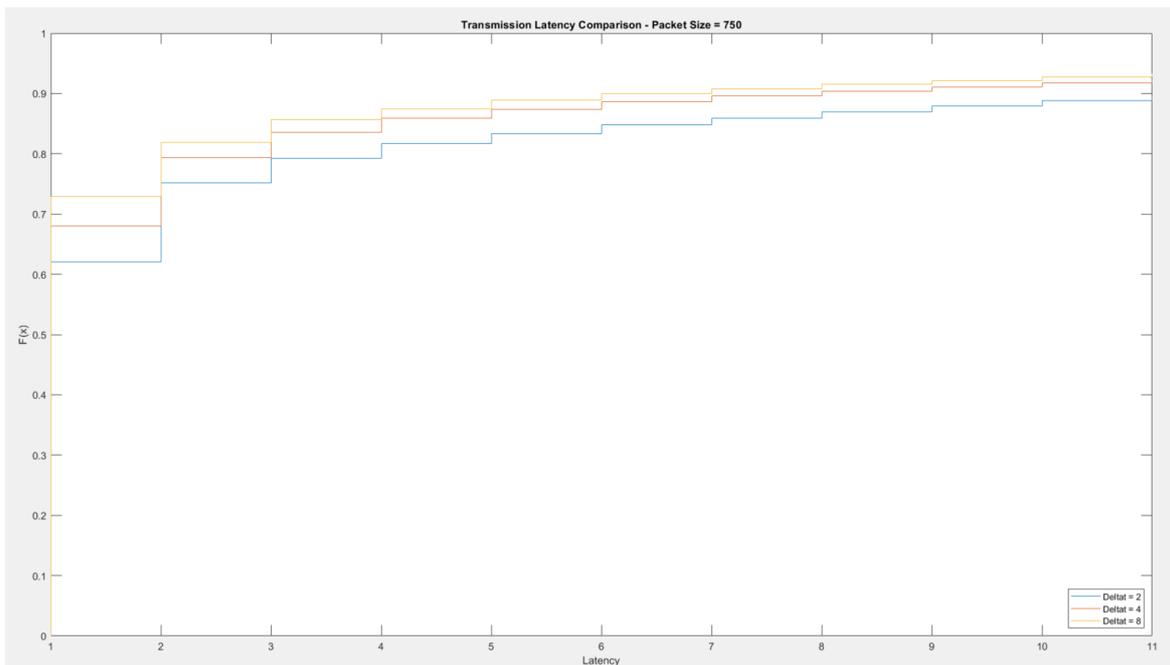


Figure 25. Transmission Latency Comparison - Packet Size = 750

When the packet size is equal to 750 bits and Δt is varying it can be observed that for all the Δt s we obtain a high percentage of packets sent with low latency values. Furthermore, there is a high percentage of packets fully sent. For $\Delta t = 2$ this percentage is almost 90%, for $\Delta t = 4$ it is around 92% and for $\Delta t = 8$ it is almost 94%.

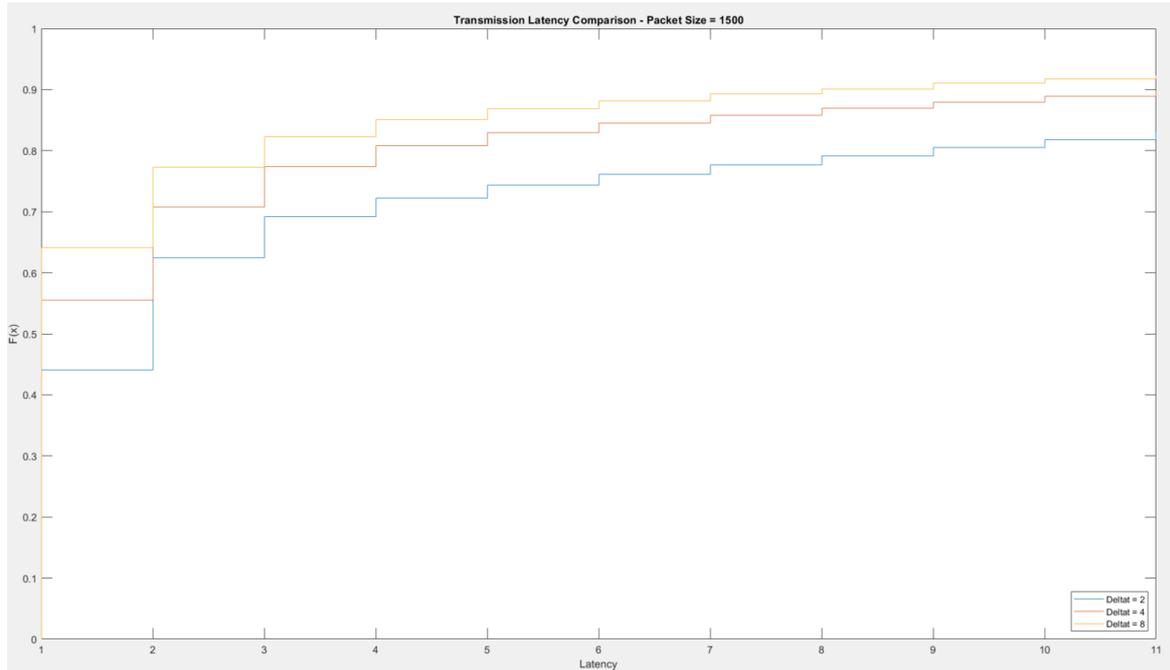


Figure 26. Transmission Latency Comparison - Packet Size = 1500

When the packet size increases to 1500 bits, there are less percentage of packets sent in low latency values and there is also a low percentage of packets fully sent without any problem, but the difference between 750 and 1500 is not too much. About the percentage of lost packets for Δt equal to 2, 4 and 8 are 17%, 10% and 8% respectively.

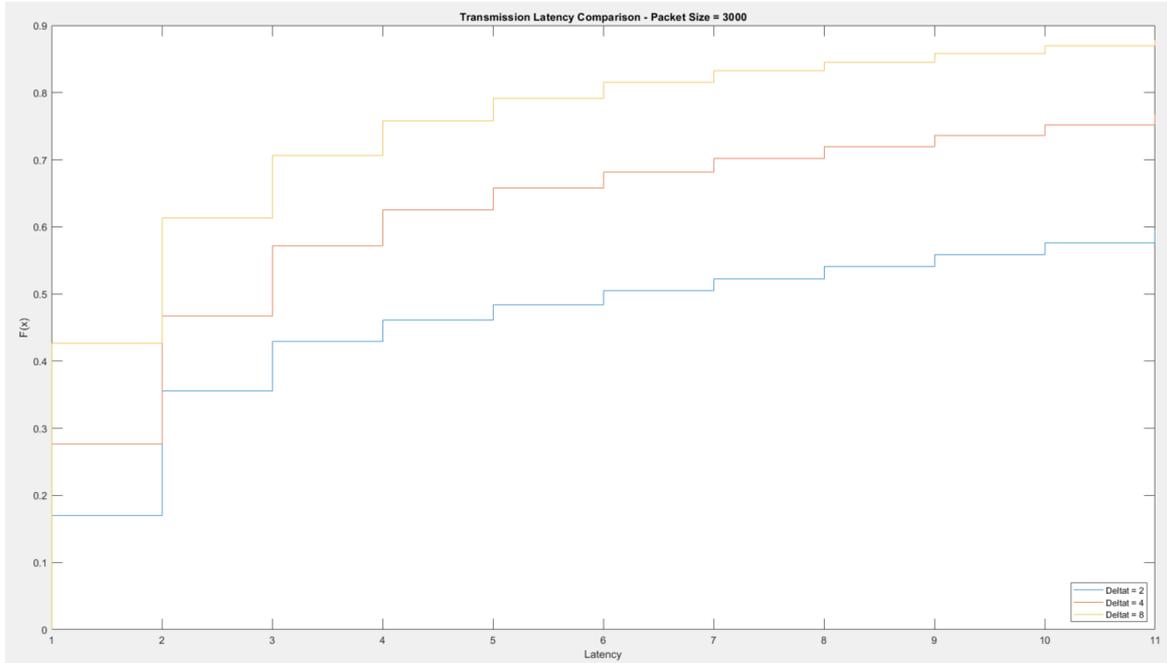


Figure 27. Transmission Latency Comparison - Packet Size = 3000

When packet size is equal to 3000 bits there are less packets that have been sent with low latency values and there are also more packets lost, specifically, with $\text{deltat} = 2$ there are a 40% of packets lost, with $\text{deltat} = 4$ this percentage is 23% and for $\text{deltat} = 8$ it is 12%.

In the next figure we can see all these comparisons in the same figure:

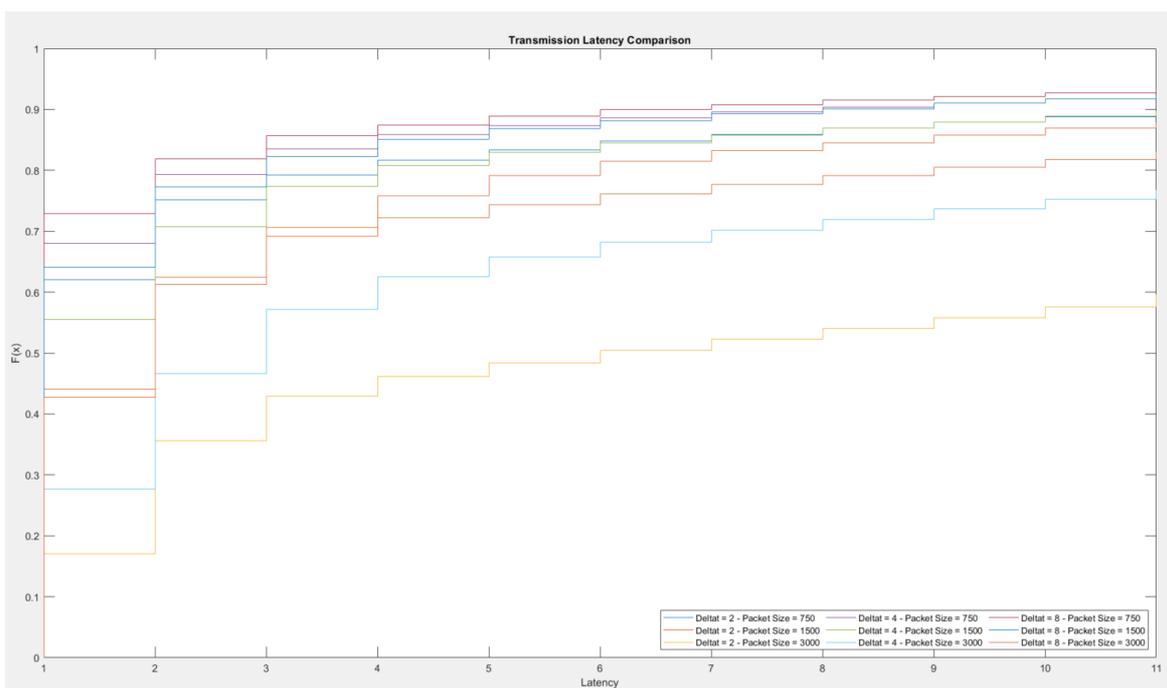


Figure 28. Transmission Latency Comparison

5. Budget

The costs that the project has to take into account are just the number of hours dedicated to this thesis at a cost of an intern engineer and the software used, but this software is Matlab and his license for students is free. Nevertheless, the license for educational institutions is about 500€.

| Task | Cost/hour (€) | Time (hours) | Total Cost (€) |
|--------------------------------------|----------------------|---------------------|-----------------------|
| <i>Literature Review</i> | 10 | 10 | 100 |
| <i>Implement traffic models</i> | 10 | 150 | 1500 |
| <i>Develop scheduling strategies</i> | 10 | 90 | 900 |
| <i>Compare values</i> | 10 | 40 | 400 |
| <i>Memory writing</i> | 10 | 40 | 400 |
| Total | 10 | 330 | 3300 |

Table 7. Budget

6. Conclusions and future development:

As a conclusion, it has been seen that there are different strategies to schedule a network, all of them with their advantages and disadvantages. It can be said that the best option is the scheduler that is based on assign RBs to users with channels with best CQI. Using this scheduler there is obtained the best results in terms of latency and percentage of wasted packets, having over a 75% of the packets sent in low latency values and around a 10% of lost packets in the best case.

Focusing into the results, it has been seen that when the Δt increases there is more time between packets, hence there is more time to send a packet before another is generated. And when a packet size decreases there are less information to send and it is easier to send all the packet without any problem.

For a future development, more traffic models could be implemented in order to have an interesting traffic mix for the scheduler. Furthermore, there are more ways to schedule a network, so another task that could be done for a future is to implement more schedulers to see their performance and to compare them with the schedulers implemented in this work.

Bibliography:

- [1] Markus Laner, Philipp Svoboda, Navid Nikaein, Markus Rupp. “Traffic Models for Machine Type Communications”.
- [2] Martin Müller, Fjolla Ademaj, Agnes Fastenbauer, Thomas Dittrich, Blanca Ramos Elbal, Armand Nabavi, Lukas Nagel, Stefan Schwarz, Markus Rupp. “User Manual. The Vienna 5G System Level Simulator”. Institute of Telecommunications, TU Wien, Vienna, October 2018.
- [3] Sacha Kavanagh. “What is 5G New Radio (5G NR)”. [Online] Available: <https://5g.co.uk/guides/what-is-5g-new-radio/>. [Accessed: 29 May 2020].
- [4] Marco Contento. “What is 5G NR?”. [Online] Available: <https://www.telit.com/blog/what-is-5g-nr/>. [Accessed: 30 May 2020].
- [5] Rubén Castro. “5G vs 4G: ¿Cuáles son las diferencias?”. [Online] Available: <https://www.wikiversus.com/moviles/5g-vs-4g/>. [Accessed: 30 May 2020].

Glossary

| | |
|--------------|--|
| 3GPP | 3 rd Generation Partnership Project. 19, 20 |
| AWGN | Additive White Gaussian Noise. 19 |
| BS | Base Station. 17, 18, 22, 23, 25 |
| BLER | Block Error Rate. 23 |
| CQI | Channel Quality Indicator. 1, 2, 3, 19, 25, 27, 28, 32, 34 |
| DAS | Distributed Antennas System. 17 |
| DL | Downlink. 24, 25 |
| ECDF | Empirical Cumulative Distribution Function. 23, 24, 28 |
| FR | Frequency Range. 19 |
| GHz | Gigahertz. 19, 20 |
| IEEE | Institute of Electrical and Electronics Engineers. 20 |
| IoT | Internet of Things. 20 |
| LoS | Line-of-Sight. 17, 19 |
| LTE | Long Term Evolution. 19, 20 |
| LTE-A | Long Term Evolution Advance. 19 |
| MIMO | Multiple-Input Multiple-Output. 18, 20, 21 |
| Ms | Milliseconds. 21 |
| NLoS | Non-Line-of-Sight. 17, 19 |
| NR | New Radio. 19, 20 |
| OFDM | Orthogonal Frequency Division Multiplexing. 20 |
| OOP | Object-Oriented Programming. 18, 22 |
| QoS | Quality of Service. 11, 14 |
| RAT | Radio Access Technology. 19 |
| RB | Resource Block. 1, 2, 3, 13, 24, 25, 26, 27, 28, 31 |
| RHH | Remote Radio Head. 17 |
| SINR | Signal to Interference and Noise Ratio. 17, 18, 19, 23 |
| SL | System Level. 22, 24 |
| SNR | Signal to Noise Ratio. 19 |
| TS | Time Slot. 22, 23 |