ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Ανάπτυξη Μηχανισμών Πρωτοκόλλων Ράδιο Πρόσβασης σε Προχωρημένα Δίκτυα 5ης Γενιάς Κινητών Επικοινωνιών

Απόστολος Χριστώνης

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Φουκαλάς Φώτιος
Επίκουρος Καθηγητής

Λαμία, Δευτέρα 22 Ιουλίου έτος 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Ανάπτυξη Μηχανισμών Πρωτοκόλλων Ράδιο Πρόσβασης σε Προχωρημένα Δίκτυα 5ης Γενιάς Κινητών Επικοινωνιών

Απόστολος Χριστώνης

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Φουκαλάς Φώτιος
Επίκουρος Καθηγητής

Λαμία, Δευτέρα 22 Ιουλίου έτος 2024

UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

Semi-Persistent Scheduling for Beyond 5G Radio Access Networks

by

Apostolos Christonis

FINAL THESIS
ADVISOR
Foukalas Fotios
Assistant Professor

Lamia, Monday 22nd July, 2024

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

*1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

*2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 22/07/2024

Ο Δηλών

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση
του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων
σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

Ευχαριστίες

Η παρούσα πτυχιακή εργασία πραγματοποιήθηκε στο Πανεπιστήμιο Θεσσαλίας, στο τμήμα Πληροφορικής και Τηλεπικοινωνιών της Λαμίας, κατα το ετος 2024.

Αρχικά θα ήθελα να ευχαριστήσω τον Επίκουρο Καθηγήτη κ.Φουκαλά Φώτιο για αυτή την ευκαιρία να εργαστώ πάνω σε ένα επίκαιρο θέμα, αυτό των 5G δικτύων. Επίσης, θερμά θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα Τσάκο Χρήστο για τον πολύτιμο χρόνο που διέθεσε για να μου δώσεις σημαντικές εξηγήσεις πάνω στο θέμα αλλα και για την βοήθειά του, την οποία δεν δίστασε ούτε στιγμή να μου τη δώσει.

Θα ήθελα ακόμη να ευχαριστήσω τον Καραβέλα Άρη (πτυχιούχος του τμήματός μας καθώς και νυν μεταπτυχιακός φοιτητής), ο οποίος μου επέτρεψε μέσα απο τις γνώσεις και την βοήθειά του να δω την σχολή με μία άλλη οπτική γωνία.

Τέλος θέλω να ευχαριστήσω την αδερφή μου (φοιτήτρια Οδοντιατρικής στο ΑΠΘ) Θεοφανώ και την μητέρα μου Ζωή, χωρίς τις οποίες δεν θα ήμουν εδώ που στέκομαι σήμερα.

Περίληψη

Η έλευση των κινητών επικοινωνιών 5ης γενιάς αναδεικνύει τη σημασία και την ανάγκη μιας αποτελεσματικής μεθόδου κατανομής πόρων. Ιδιαίτερα για υπηρεσίες όπως το Voice over Internet Protocol(VoIP), η ανάγκη για συνεπή μεταφορά δεδομένων αναδεικνύει την πρόκληση που δημιουργεί το υπέρμετρο signaling overhead, το οποίο παράλληλα καθιστά δύσκολη την κλιμάκωση για την εξυπηρέτηση μιας μεγαλύτερης βάσης χρηστών.

Στόχος της παρούσας εργασίας είναι μια λειτουργική υλοποίηση του Semi-Persistent Scheduling(SPS) ως εναλλακτική μέθοδο μείωσης αυτής της επιβάρυνσης στα προηγμένα δίκτυα ραδιοπρόσβασης 5ης γενιάς και βελτίωσης της συνολικής απόδοσης. Θα καλυφθεί επίσης λεπτομερώς η ενσωμάτωση του, στο έργο ανοικτού πηγέου κώδικαsrsRANProject καθώς και οι προκλήσεις που παρουσιάστηκαν.

Η διαδικασία ανάπτυξης του SPS, περιελάμβανε τη δημιουργία νέων δομών δεδομένων, την τροποποίηση των υφιστάμενων αλγορίθμων χρονοπρογραμματισμού και την ανάπτυξη μηχανισμών που αφορούν τη λειτουργία του SPS. Η εργασία μας επεκτάθηκε σε πολλά επίπεδα της στοίβας πρωτοκόλλων, συμπεριλαμβανομένων των επιπέδων RRC, MAC και PHY.

Τα βασικά σημεία αυτής της εργασίας έχουν ως εξής. Η λειτουργική υλοποίηση του SPS στο πλαίσιο του ανοικτού κώδικα srsRANProject. Η λεπτομερής παρουσίαση των προκλήσεων και των λύσεων της ενσωμάτωσης του SPS στον υπάρχοντα κώδικα και αρχιτεκτονική, και τελος η παρουσίαση της μείωσης του signaling overheadπου επιτυγχάνεται με την υλοποίηση του SPS .

Οι προσομοιώσεις μας δείχνουν ότι το SPS μπορεί να μειώσει σημαντικά την επιβάρυνση του χρονοπρογραμματισμού ωστόσο διατηρώντας παρόμοιο, και σε ορισμένες περιπτώσεις ελαφρώς μειωμένο, packet latency. Υπήρξαν ωστόσο περιορισμοί και παραχωρήσεις που έπρεπε να γίνουν, όπως η υποστήριξη μόνο ενός UE και η μη δυναμική περιοδικότητα. Η παρούσα εργασία έχει τέλος ως στόχο να αποτελέσει τη βάση για μελλοντική έρευνα των μηχανισμών χρονοπρογραμματισμού σε Beyond 5G networks, συμβάλλοντας έτσι, άμεσα, στο διαρκώς μεταβαλλόμενο τοπίο των κινητών επικοινωνιών.

Abstract

The advent of 5th Generation Mobile Communications highlights the importance and need of an efficient resource allocation method. Particularly more so for services such as Voice over Internet Protocol (VoIP), the need for consistent data transfer highlights the challenge of excessive control signaling overhead, which makes it difficult to scale up to accommodate a larger user base.

The goal of this paper is to present a working implementation of Semi-Persistent Scheduling(SPS) as an alternative method of reducing that overhead in advanced 5th generation Radio Access Networks(RAN) and improving overall performance. The integration of SPS into the open source srsRANProject and the challenges presented will also be covered in detail.

The process of developing SPS, included creating new data structures, modifying existing scheduling algorithms and developing mechanisms pertaining the operation of SPS. Our work extended in multiple layers of the protocol stack, including the RRC, MAC and PHY layers. The key points of this paper are:

- A working implementation of SPS within the framework of the open source srsRAN Project.
- Detailing the challenges and solutions of integrating SPS into the existing code stack and architecture.
- A presentation of the reduction in the scheduling overhead achieved with the implementation of SPS

Our simulations signal that SPS can substantially reduce the scheduling overhead while maintaining similar, and in some cases slightly reduced, traffic latency. There were however limitation and concessions that had to be made, such as the support for only a single UE and non-dynamic periodicity.

This paper is meant to provide a foundation for future research of scheduling mechanisms in Beyond 5G networks, contributing to the ever changing landscape of mobile communications.

Abbreviations

5G	5th Generation
VoIP	Voice over Internet Protocol
SPS	Semi-Persistent Scheduling
RAN	Radio Access Network
RRC	Radio Resource Control
MAC	Medium Access Control
PHY	Physical (Layer)
DCI	Downlink Control Information
QoS	Quality of Service
UE	User Equipment
URLLC	Ultra-Reliable Low-Latency Communication
3GPP	3rd Generation Partnership Project
gNB	Next Generation NodeB
PDCCH	Physical Downlink Control Channel
PDSCH	Physical Downlink Shared Channel
PUSCH	Physical Uplink Shared Channel
HARQ	Hybrid Automatic Repeat Request
MCS	Modulation and Coding Scheme
RNTI	Radio Network Temporary Identifier
CS-RNTI	Configured Scheduling - Radio Network Temporary Identifier
C-RNTI	Cell - Radio Network Temporary Identifier
CRC	Cyclic Redundancy Check
SDR	Software Defined Radio
eMBB	Enhanced Mobile Broadband
mMTC	Massive Machine Type Communications
CU	Central Unit
DU	Distributed Unit
FEC	Forward Error Correction
ARQ	Automatic Repeat Request
PDCP	Packet Data Convergence Protocol

Contents

1 Introduction

In the modern era of wireless communication, the emergence of 5G technology has signalled a period of significant transformation. This evolution brings with it a revolution in the capability of networks to support a range of varied devices, services and applications. In contrast to the previous generations of mobile networks, which primarily focused on increasing the bandwidth for consumer devices, 5G introduced through even its first Release[1], a broader vision for connectivity. This vision not only aims to provide faster network access but also more reliable and efficient data transfer. Among the three primary use case categories identified by 5G—Enhanced Mobile Broadband (eMBB), Ultra-Reliable and Low-Latency Communications (URLLC), and Massive Machine Type Communications (mMTC)—it is URLLC that presents the most significant challenges. URLLC promises not only ultra low latency but also high levels of reliability, which traditionally have been seen as almost mutually exclusive in network design[2]. This was due to enhancing reliability required mechanisms like redundancy and re-transmission, inherently introducing additional latency. The innovation of 5G lied in its ambition to bridge this gap, offering a network capable of achieving this and also supporting applications such as autonomous vehicle navigation, tactile internet, and remote medical procedures. But 5G is about much more than this. It represents a fundamental change in how communications are structured and provides the "waterway" via which new innovations will be enabled.

The main motivation for developing SPS, was to provide a working paradigm of the protocol. Whilst there are multiple research papers delving into the theoretical advantages of SPS and the potential reduction of scheduling overhead[3], there are almost none to be found that actually implement the protocol and provide real world samples. With the increase of mobile devices connected to the network there is an ever-pressing concern of bandwidth limitations. The goal of SPS is to address those concerns by addressing latency requirements, reducing bandwidth usage, enhancing QoS while facilitating as infrastructure for features such as URLLC. Overall the development of SPS was motivated by a need to improve network efficiency, meet strict latency requirements whilst maintaining a high Quality of Service and keeping, or in some cases even reducing, operational costs from ballooning.

Beyond the introduction, this paper will first present a general overview of how scheduling works, with chapter 2 going more in depth of what the implementation actually entailed. Finally simulation results will be presented, along with an afterword. References and appendices will be presented at the end.

1.1 RAN architecture

The RAN or Radio Access Network is a component of the 5G architecture that is mainly responsible for the connection of different UEs to the core 5G network. The RAN has one central goal, that is to transmit packets between the UEs and the mobile core network. This implies that the RAN is deeply interwoven with the management and scheduling implemented by the core network.

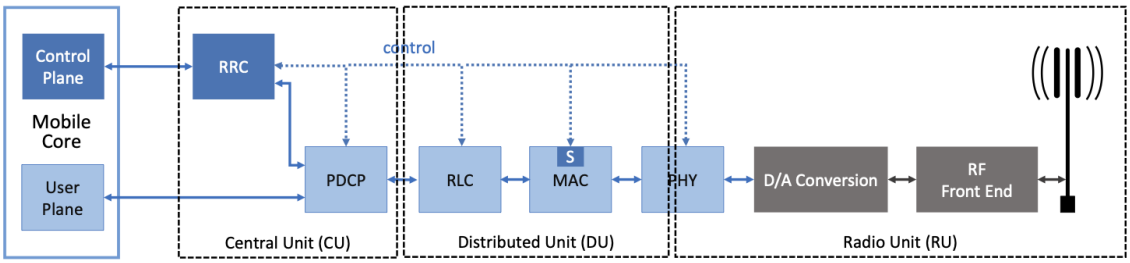

Figure 1.1: CU/DU Split of RAN[4]

The RAN packet processing pipeline, which can also be considered a protocol stack as well, has multiple stages, as specified by the 3GPP standard. Specifically:

- Physical Layer (PHY), the lowest layer of the protocol stack, which is mainly used for signal modulation/demodulation, forward error correction(FEC) and channel encoding/decoding.
- Media Access Control (MAC), which is the next layer after PHY, is mainly responsible for handling access to the radio resources. Main tasks include scheduling, multiplexing/demultiplexing as well as packet buffering.
- Radio Link Control (RLC), is a stack protocol that exists between the MAC and PHY layers and assigned tasks such as packet segmentation and reassembly, while also responsible for reliability through error correcting with Automatic Repeat Requests (ARQ).

These three layers constitute what is commonly referred to as the Distributed Unit (DU). The DU is part of RAN subdivision which splits the protocol stack in three "units". The next layers are part of the Central Unit (CU)

- Radio Resource Control (RRC), the highest layer in the Central Unit of the CU/DU, it is mainly responsible for establishing connection with the UEs, maintance through quality assurance(i.e signal degradation due to distance from gNB) and management of radio resources. It works as a coordinator between UEs and the RAN, increasing efficiency in radio resource allocation and mobility management.
- Packet Data Convergence Protocol (PDCP), is the lowest layer of the CU and mainly responsible for handling data convergence between the higher layers and the radio interface. It includes functionalities such as IP header compression/decompression, ciphering, integrity protection etc.

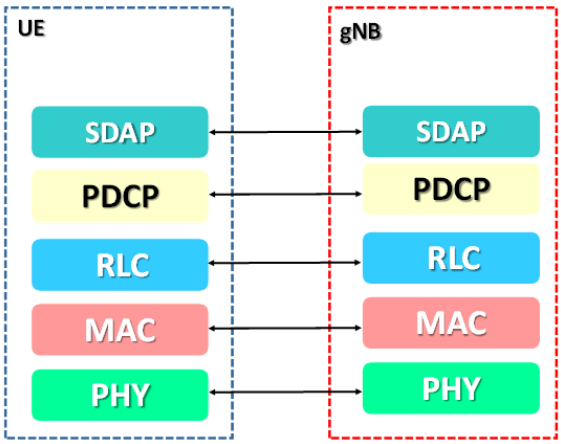

Figure 1.2: RAN Protocol Stack

This summarizes a high level description of the RANs internal protocol structure. This is in line with the 3GPP specifications and the paper doesn't stray far from those specifications, with certain exceptions that needed to be made due to the way the framework upon which we developed the protocol was structured.

1.2 Open Source Solutions and srsRAN

The introduction of 5G networks represents a major change, moving away from the hardware-focused infrastructures of the past to more flexible and adaptable software-based systems. This transition has been accelerated by the rise of open-source solutions in RAN, allowing for collaborative innovation which is crucial for adapting to the evolving requirements of 5G networks. Among such open-source platforms, srsRAN[5] is a noteworthy example, showing the power of community-driven development to create network solutions.

As a software suite, srsRAN provides tools and frameworks for implementing a wide range of 5G functionalities from the upper layers of the stack to the lower ones. Its compatibility with both commercial and research-oriented applications demonstrates its role in bridging the gap between academic research and practical, real-world deployment. srsRAN offers both a gNB and a UE application and it can be used with third-part CN which makes it ideal to build a complete end-to-end mobile wireless network. It is written in C++, runs in Linux and offers extensive documentation and active community support. For all the above reasons, srsRAN was chosen for this thesis to implement and evaluate our solution.

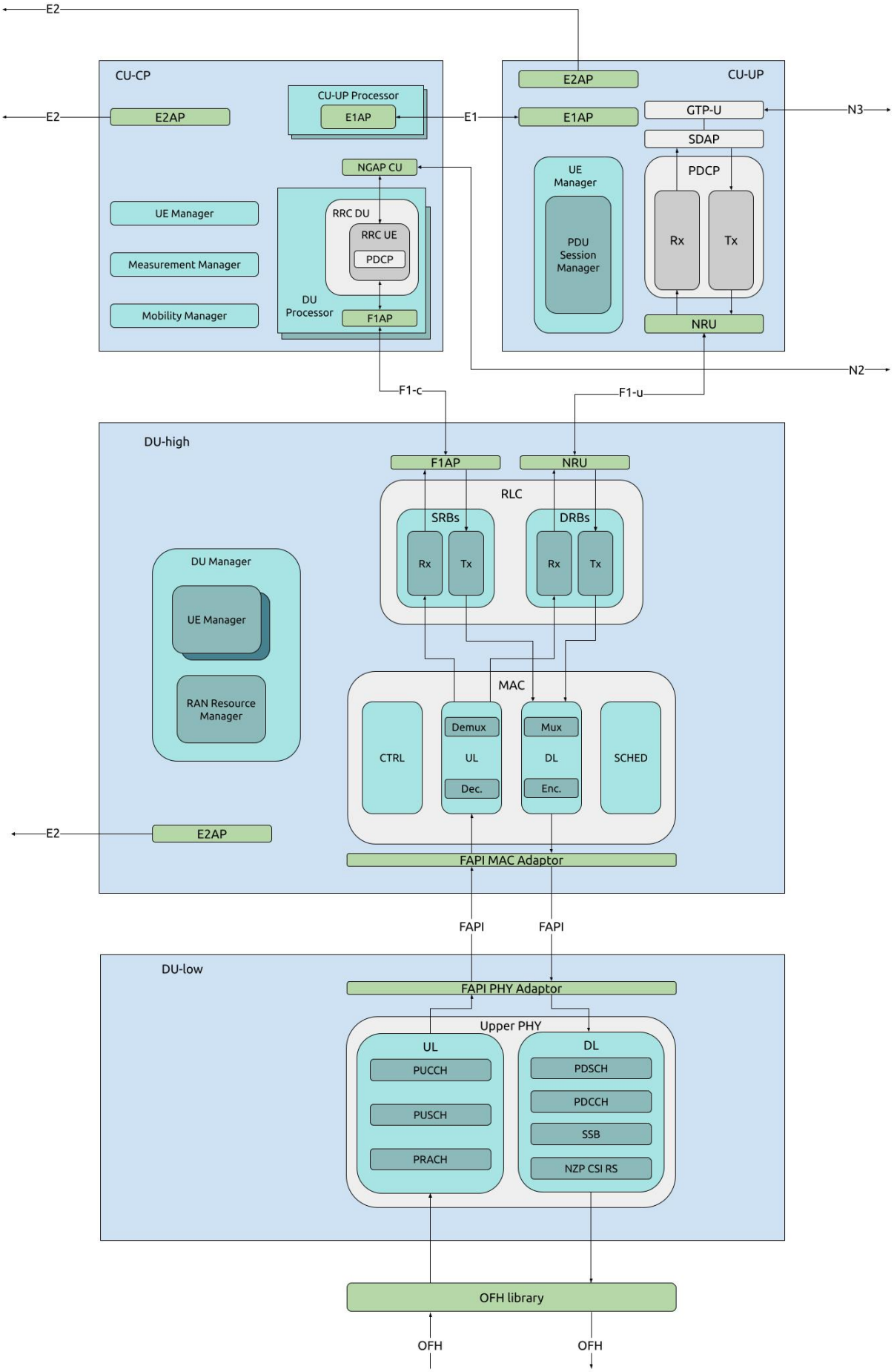

Figure 1.3: High level overview of the srsRAN Project codebase architecture showing all main components and interfaces[5]

2 Literature Review

This literature review aims to examine key works that were used for this paper, specifically ones focusing on Semi-Persistent Scheduling (SPS) and its implementation in modern communication networks.

2.1 Evolution of Semi-Persistent Scheduling

Feng et al. (2019)[1] suggest a new way to improve Semi-Persistent Scheduling (SPS) for applications that need very low delay in LTE and 5G networks. Their method uses prediction to make resource allocation more efficient, overcoming the drawbacks of traditional SPS. This shows that SPS can be modified and enhanced to meet the strict demands of new 5G applications.

The foundational principles of SPS are explored in depth by Jiang et al. (2007).[3] Although their work focuses on LTE systems, it provides crucial insights into the performance benefits of SPS for Voice over IP (VoIP) services. Their analysis of the trade-offs between dynamic and semi-persistent scheduling laid the groundwork for further research in this area. Furthermore, additional information was taken from multiple other research papers such as Semi-Persistent Scheduling for 5G Downlink based on Short-Term Traffic Prediction[6], Estimation of Effective Radio Resource Usage for VoIP Scheduling in OFDMA Cellular Networks[7], Control traffic overhead for VoIP over LTE[8] and 5G RAN: physical layer implementation and network slicing[9] which provided the necessary theoretical background necessary to understand 5G networks and undertake the implementation of SPS.

2.2 SPS in the context of 5G Technologies

Morais (2020)[10] offers an in depth overview of 5G physical layer technologies in his book. While not exclusively focused on SPS, this book covers key technologies associated with 5G communication networks. The author's discussion of enabling technologies for mobile and fixed wireless access helps situate SPS within the larger framework of 5G innovations.

Peterson et al. (2023)[4] take a systems approach to private 5G networks in their book. This perspective is valuable for understanding how scheduling mechanisms like SPS fit into the overall architecture of 5G systems. Their work highlights the importance of considering scheduling not in isolation, but as part of an interconnected system of

technologies and protocols.

The technical specifications for Medium Access Control (MAC) protocol in 5G networks are detailed in the 3GPP TS 38.321 document (2021). This specification is crucial for understanding the standardized framework within which SPS must operate. It provides the technical foundation for implementing SPS in compliance with 5G standards.

Rodrigues (2023)[11], in his master's thesis, explores open-source solutions for non-public 5G networking. This work is particularly relevant for understanding the practical challenges and opportunities in implementing advanced scheduling mechanisms like SPS in real-world, open-source 5G systems.

Finally, the online resource provided by Jaeku Ryu, under the tech blog ShareTechnote[12] include technical explanations of configured scheduling in 5G networks. While not a peer-reviewed academic source, this resource offers valuable practical insights into the implementation details of SPS and related scheduling mechanisms.

In conclusion, the works collectively highlight the potential of SPS to address key challenges in 5G resource allocation, particularly for low-latency applications. Additionally, the works used within the context of this paper provide the much necessary knowledge and framework from which to understand the potential of SPS in modern URLLC as well as the broader 5G communications spectrum. They also emphasize importance of standardization efforts undertaken by organizations such as 3GPP. Finally they provide a solid foundation for our implementation and analysis of SPS in 5G networks, informing both our approach and our understanding of its potential impact on next-generation mobile communication networks.

3 Scheduling in 5G Networks

3.1 Scheduling

Scheduling is the process of allocating resources to each UE with the explicit purpose of data transmission. As presented in Moray's Key 5G Physical Layer Technologies[p.248][10] "Scheduling is key function of NR (and any multiuser mobile system) and to a significant extent determines the overall behavior of the system." and in general the NR scheduling mechanisms are similar to those of LTE communications, with the exception of NR having a greater level of detail when compared with LTE, more so in terms of time domain scheduling at the lower levels(i.e PHY). The choice of a scheduling algorithm is one of the most, if not the most, crucial part of real time resource allocation in RAN, which will serve a plethora of different users and services. As such there have been a diverse range of solutions provided, with SPS being one.

3.2 Uplink Scheduling

Uplink scheduling works with what is colloquially known as grants. A UE sends a scheduling request (SR) to the gNB, which also includes information such as the amount of data that is to be transmitted. Dynamic scheduling is the basic allocation algorithm in which the scheduler for each time slot, determines which devices will receive information and which will transmit. The main difference with semi persistent scheduling is that the algorithm provides the transmission parameters in advance to each UE or device.

3.2.1 Dynamic Uplink Scheduling

The fundamental role of a dynamic uplink scheduling algorithm is to control the number of UEs that will transmit on certain resources and with which specific transmission parameters. In other words, the scheduler decides which UEs will transmit at the current time instance and on which radio resources by allocating them blocks in the resource grid. At the same time, the scheduler has to inform every UE for the resources allocated to them and this is done by using the control signalling and the PDCCH channel. Every UE then has to decode the information on the PDCCH channel, extract the information on the resources it could transmit and then carry out the transmission on those resources.

Overall, Dynamic scheduling is the process that allocates PUSCH resources by DCI messaging. When a UE has data waiting in the Logical Channels but no pre-configured

UL grant , the UE will message the gNB for a Scheduling Request.[11]

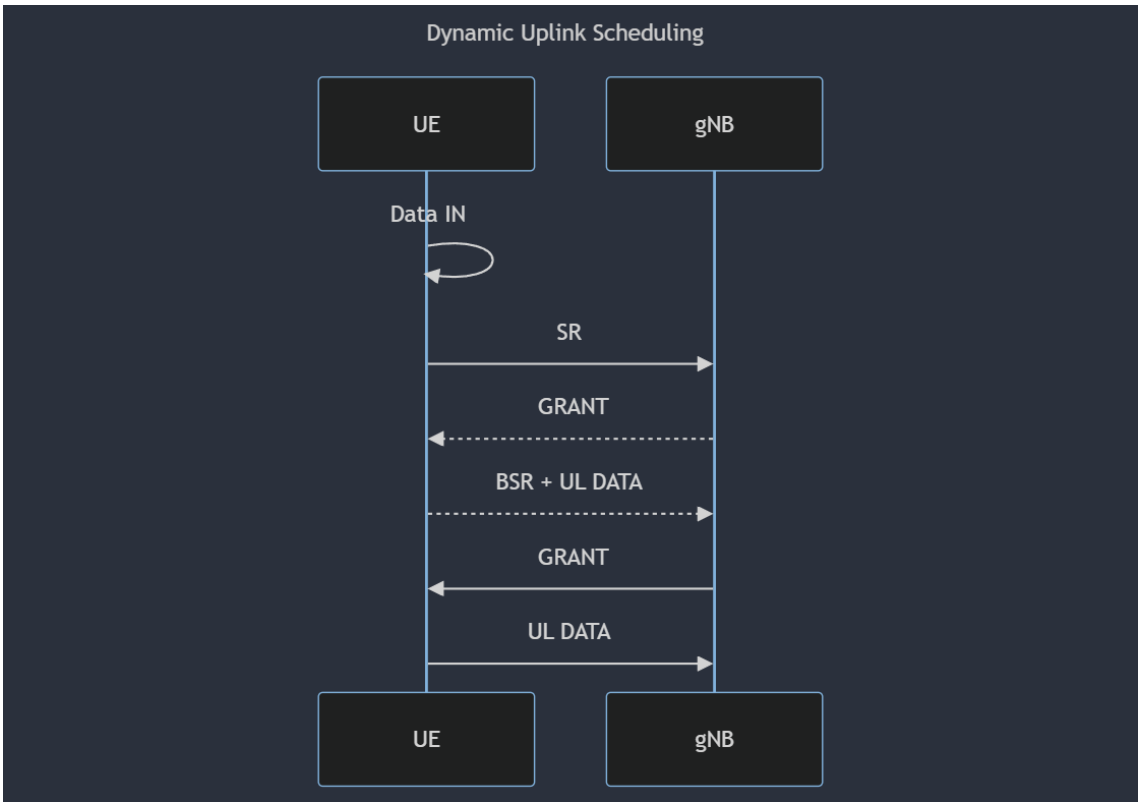

Figure 3.1: Dynamic Uplink Scheduling

3.2.2 Semi-Persistent Uplink Scheduling (Configured Grant)

SPS or configured scheduling, conversely, can assign the PUSCH without having to deploy a DCI message for every transmission. It instead configures all the required transmission data once, which also helps reduced the aforementioned signaling overhead, thus enabling a larger number of users to be scheduled for transmission. Additionally, there has been a push for SPS to be used as a potential solution to reducing buffering times for low latency packets.

As defined by 3GPP, SPS uplink scheduling has 2 different types of grants.

- The first type of configured grant, aptly named Type 1 configured grant, is represented as DCI_0_0, is provided by the RRC layer and provides the pre-configured parameters required by the UE in order to perform PUSCH scheduling. The RRC layers provides all the required parameters, specifically ConfiguredGrantConfig and rrc-ConfiguredUplinkGrant. As a result the UE can transmit after it successfully finished the RRC connection as long as it has data. It should however continue monitoring the PDCCH channel for any changes on its transmissions dictated by the gNB.

- Type 2 configure grant, which is represented in the code as DCI_0_1, gets provided to the UEs using the PDCCH. It also makes use of an auxiliary L1 signal, the DCI.

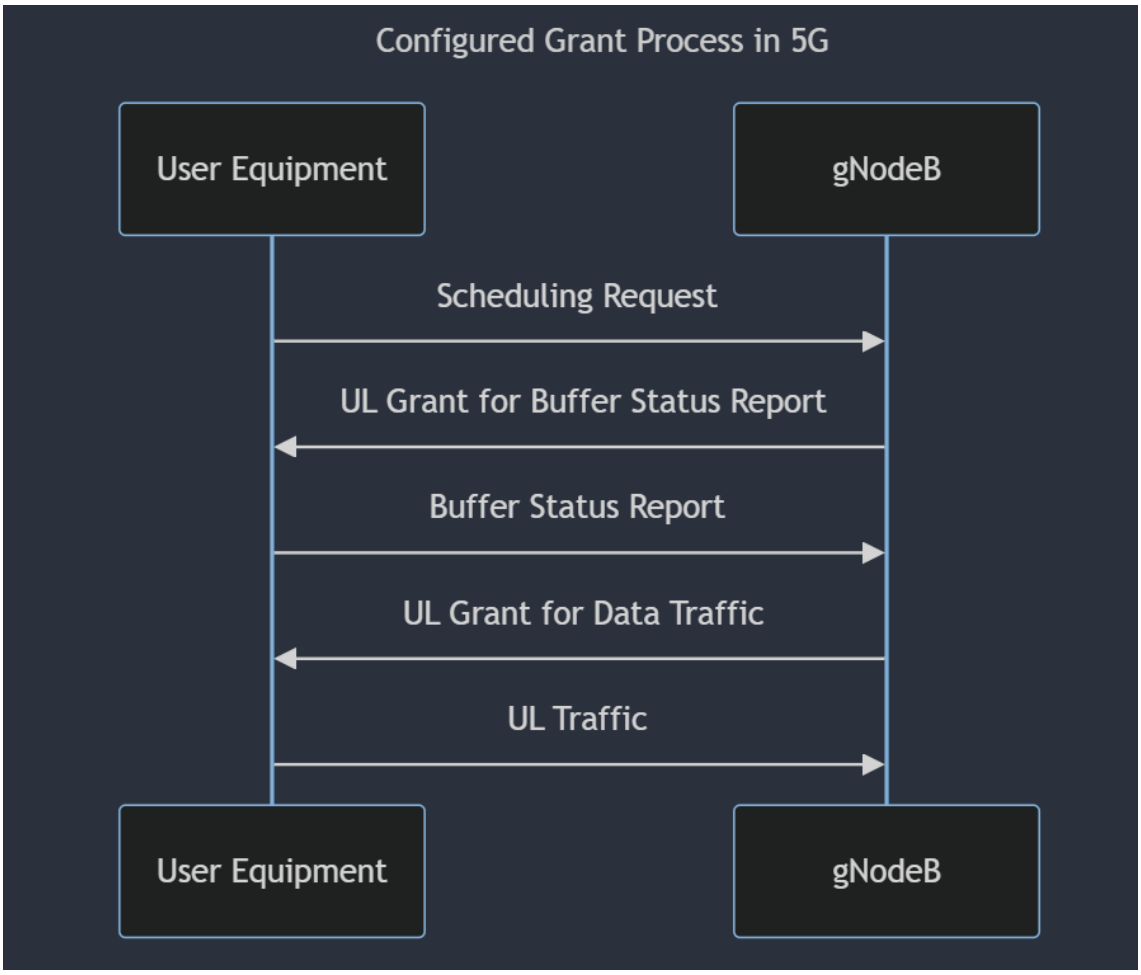

Figure 3.2: Configured Grant Process

In this case of configured grant only the ConfigureGrantConfig is give by the RRC layer. The DCI additionally can enable or disable the grant, which allows for further control of the resources allocated to the UEs. So we could say that the grant in this case is partially provided by the RRc layer and partially given still by the DCI as in the case of the dynamic schedulling. The big difference with dynamic schedulling is that the DCI is only provided once at the activating of the grant and not for every data tranmsission. Upon reception of the DCI the UE keeps the grant and repeats it at a periodic basis for data transmission.

3.3 Downlink Scheduling

One characteristic of both dynamic and SPS is the Radio Network Temporary Identifier (RNTI), which is a type of cyclic redundancy check(CRC) mask that is provided to UEs and is used to decode the DCI messages that they receive. It can be used to identify one UE from another. Dynamic scheduling and SPS use different types of RNTI, with dynamic using Cell-RNTI(C-RNTI) and SPS using ConfiguredScheduling-RNTI(CS-RNTI). There are other types of RNTI such as TemporaryCell-RNTI(TC-RNTI) and RandomAccess-RNTI(RA-RNTI) but those are outside the scope of this thesis.

During downlink scheduling the UEs monitor the Physical Downlink Control Channel(PDCCH) for a DCI message, that would indicate downlink data intended for that UE. The DCI message is sent by the PHY layer of the gNB, and is masked with C-RNTI for dynamic scheduling or CS-RNTI for SPS. The allocation of resources is decided by the scheduler using different metrics such as the Channel Quality Indicator(CQI), SNR etc.

3.3.1 Dynamic Downlink Scheduling

In every subframe, UEs will monitor the PDCCH for a scheduling assignment. One of the main reasons dynamic scheduling has a large latency delay, is that the UEs attempt to blindly decode the PDCCH using the RNTI CRC mask provided by the DCI message in order to verify if the message is intended for it. After the DCI message is decoded, the UE will then extract the location of the PDSCH which will contain the actual DL transmission data.

3.3.2 Semi-Persistent Downlink Scheduling

In SPS, the UEs have specific periodic, pre-configured resources, and based upon the periodicity of those resources, expects downlink data traffic to occur. These "pre-allocated" resources are partially provided by RRC signalling as well as a DCI message that will also activate the SPS. The parameters of transmission configured by RRC layer are the nrofHARQProcesses, which tallies the total number of configured HARQ processes used for SPS, and finally periodicity which is the time interval during which the UE can expect a transmission opportunity for SPS.

The actual radio resources of the PDSCH are provided by the DCI message that activates the SPS. In particular, the DCI message will include the time and frequency resources that will be made available to be transmitted for downlink user data traffic, all according to the periodicity provided. In contrast to dynamic scheduling the DCI does not include the HARQ process ID for the transmissions following this one, which in context makes sense, since this particular DCI message pertains to not just one transmission. Instead the HARQ process ID is determined by the both the gNB and UE based on a predefined formula given by 3GPP[13] and presented here:

$$HARQProcessID = [floor(CURRENT\_slot * 10/(numberOfSlotsPerFrame * periodicity))] modulo\ nrofHARQProcesses$$

Finally similarly to Dynamic scheduling, the UE still has to constantly monitor the PDCCH in case it is dictated by the gNB for SPS deactivation or a dynamic grant which will override the SPS grant.

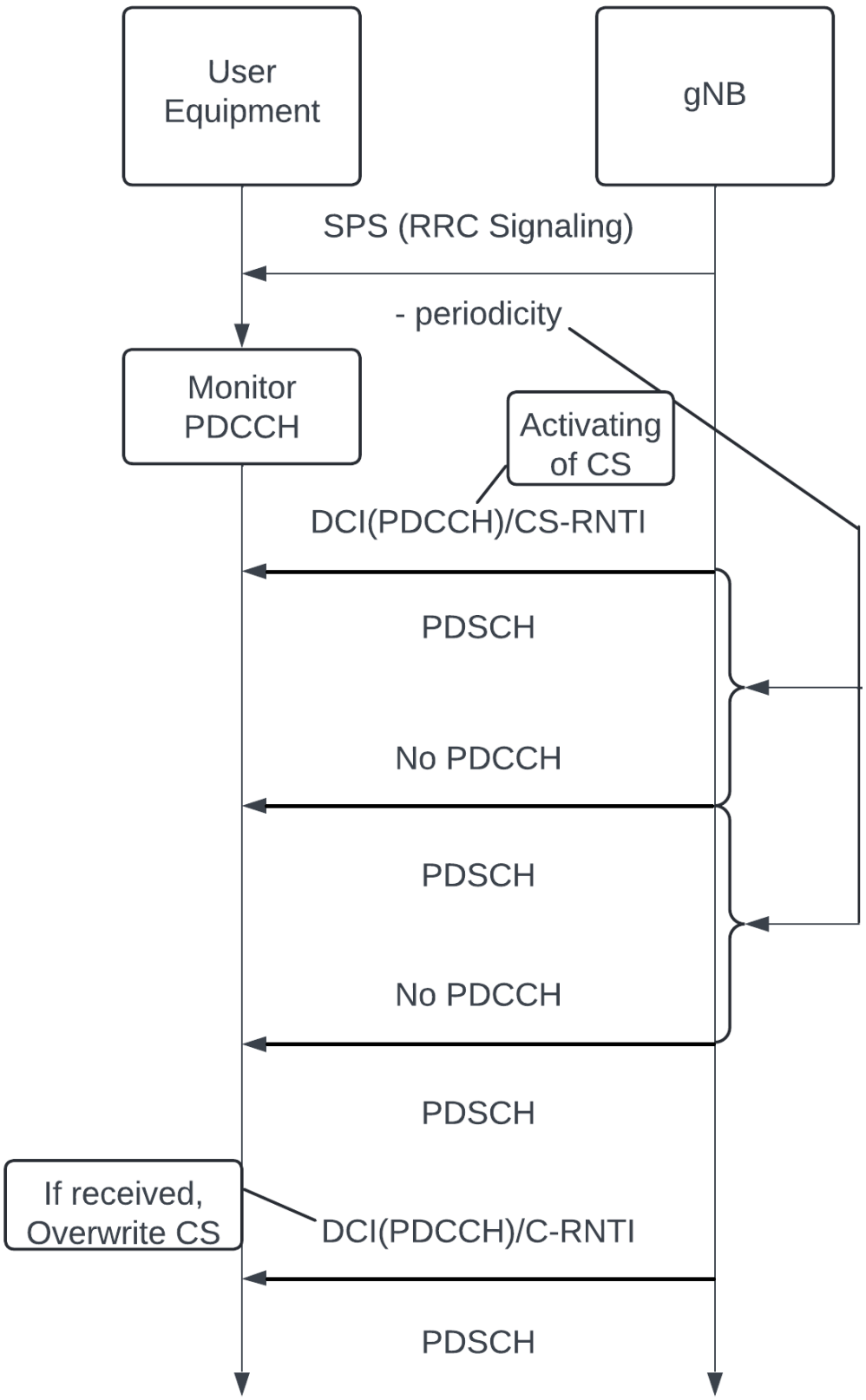

Figure 3.3: Semi-Persistent Scheduling

4 SPS Implementation

4.1 SPS Implementation Overview

The changes made to the SRSRAN protocol stack were done all according to the technical specifications[13]. The process included reading the technicaly specification requirements for all the layers of 5G communications to support SPS and then trying to implement those in the stack. Understanding what the SRSRan project has implemented as part of their protocol stack, and finally extending the current protocol stack to support semi-persistent(non-dynamic) scheduling. Examples of such changes include, but are not limited to, the definition of SPS Grant struct in the serving_cell_config class or the implementation of a method for converting the serving_cell_struct to an ASN1 message for both CS and SPS.

At the moment, our implementation of SPS only supports up to one UE. Further additions need to be made to the SRSran project code stack for it to support multiple UEs. Furthermore, the current implementation of SPS was made for one UE, without consideration to periodicity or resource allocation. Should a second UE attempt to connect in our implementation , there would be conflicts when attempting to allocate resource since we the same Resource block and periodicity for every UE. Of course in a real world scenario, a properly designed scheduler, would care to allocate different set of RBs to prevent such collisions. Moreover, the periodicity of each UE would be decided by the scheduler to match the UE's traffic pattern, as closely as possible, in an effort to increase the total utilization of resources. While our implementation supports any of the spec defined periodicity, it does so in a pre-defined manner. Specifically the periodicity value is pre-configured at the gNB application level and it isn't defined based on current needs.

4.2 gNB SPS Modifications

To make the gNB compatible with the 3GPP specifications of SPS, additions had to be made to multiple layers of the code stack to actually support SPS as a feature. Specifically intervention was necessary in the RRC, MAC and PHY layers.

Working in a top to bottom fashion we present the changes made in each layer, challenges faced on implementation and discuss issues faced when developing and implementing those changes.

4.2.1 RRC Layer Modifications

Despite the fact that SPS is technically a scheduling mechanism, one would expect its mechanism to work on the MAC layer, its semi-static nature however, necessitates it involving the RRC layer. Work in the RRC included the enhancement of the already existing RRCReconfiguration procedure to include the SPS-Config field. To do so, we defined an SPS-config struct which included all the spec defined values for fields such as periodicity, MCS table or nrofHARQ-Processes the necessary. Then, we make sure this struct was included in the RRCReconfiguration message sent to the UE during RRC Reconfiguration procedure. As it is already mentioned, in our implementation the periodicity value, which will dictate the rate at which those pre-configured resources reoccur, isn't chosen to match the traffic rate which will not lead to the most best utilization of SPS. Different periodicity values, are however, supported. Later Releases of 5G could also support multiple SPS Configurations per UE, each with different values for time and frequency resources. Our solution is limited to one SPS-Config per UE. Additionally, we had to guarantee that during the RRC Reconfiguration process, the UE gets a different type of RNTI. Messages sent on the PDCCH get scramble with the RNTI of each UE. The aim of this scrambling is to make sure that each UE can tell apart which message is intended for it. SPS directly makes use of RNTI, specifically CS-RNTI. It was thus critical that the RRC message through which the UE gets assigned a CS-RNTI value was correctly implemented and added to the RRC-Reconfiguration message.

4.2.2 MAC Layer Modifications

Expanding upon the pre-existing MAC layer functionalities, we attempt to implement SPS as part of the code stack. The main challenges faced with expanding the code stack to that direction was the complete lack of a framework intended to support SPS.

In a general sense, changes were made mostly focused around scheduler_time_rr, which is part of the overall scheduling strategy of the gNB and ue_cell_grid_allocator whose main functionality is correctly allocating the scheduling grant to the resource grid. Finally, the result of the scheduler is the DCI message, which is then sent to the UE to indicate the grid allocation of the scheduled data. To this end, inspired by the method SRSRAN already implemented to schedule downlink data dynamically we implement a similar methods for SPS.

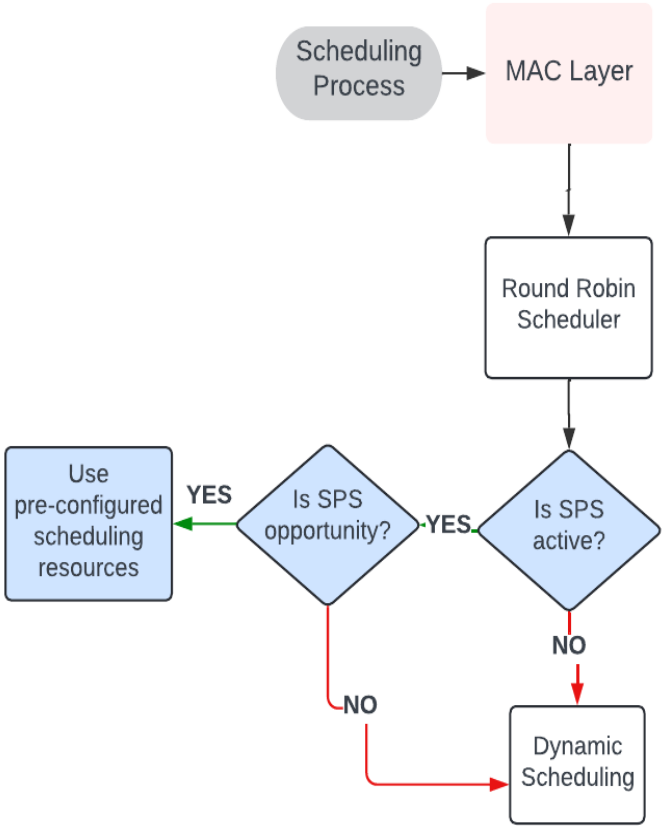

Figure 4.1: Additions made to the SRSRAN stack, shown in blue

Scheduler_Time_RR

The Scheduler_Time_RR source mostly includes functions and classes related to the allocation strategy used by the gNB. In this case round robin is used in an attempt to provide fairer resource allocation. For the scope of this paper we are mostly interested in dl_sched and alloc_sps_ue along with their constructors. To enhance the scheduler design with SPS capabilities we first had to grasp a general overview of its functionality. Being a sub-module of the MAC layer the scheduler has distinct "implementations" for different types of traffic needed to be sent to each UE. There is the "common" scheduling which concerns types of data that can be shared by multiple UEs (like paging or the System Information Block scheduling) and there is the UE scheduling associated class where UE-specific data is scheduled. The scheduler runs at every slot and the results of the scheduler are then communicated to the PHY layer. By examining the scheduler class design and since, in theory, SPS is used to transmit user-data traffic it was decided that the best place to implement the scheduler was the UE scheduling sub-class. There it was understood that the gNB attempts scheduling in a round-robin strategy, split into two methods, one for uplink: ul_sched and one for downlink: dl_sched. Upon reviewing the dl_sched method it was noticed that it actually implements dynamic scheduling carrying through

the alloc_dl_ue method for both initial transmissions of data as well as re-transmissions.

The consideration here was where exactly to implement SPS. Should it be included within the alloc_dl_ue or should we have constructed an entire new method instead? It was decided then, that since whenever SPS fails to schedule, re-transmission instead occurs with dynamic scheduling, we constructed a new method to handle SPS specifically based upon the pre-existing methods and making the necessary changes.

```
1
2  alloc_outcome ue_cell_grid_allocator::allocate_sps_grant(const
       ue_sps_grant& grant){
3
4    //
5    logger.info("Activating and allocating grant");
6    //Important checks for allocation
7    srsran_assert(ues.contains(grant.user->ue_index), "Invalid UE
       candidate index={}", grant.user->ue_index);
8    srsran_assert(has_cell(grant.cell_index), "Invalid UE candidate
       cell_index={}", grant.cell_index);
9
10   if (dl_attempts_count++ >= expert_cfg.
       max_pdcch_alloc_attempts_per_slot) {
11     logger.info("Stopping DL allocations. Cause: Max number of DL PDCCH
         allocation attempts {} reached.",
12                 expert_cfg.max_pdcch_alloc_attempts_per_slot);
13     return alloc_outcome::skip_slot;
14   }
15
16   ue& u = ues[grant.user->ue_index];
17   // Verify UE carrier is active.
18   ue_cell* ue_cc = u.find_cell(grant.cell_index);
19   if (ue_cc == nullptr or not ue_cc->is_active()) {
20     logger.info("PDSCH allocation failed. Cause: The ue={} carrier with
         cell_index={} is inactive",
21                 u.ue_index,
22                 grant.cell_index);
23     return alloc_outcome::skip_ue;
24   }
```

The new method had to account for for both the activation of SPS and the functionality of SPS after activation. The method initially checks if SPS is activated branching into two different options based on the SPS status. If SPS is inactive, it is activated. This only needs to occur once for the whole transmission. The activation is configured to occur at a specific TTI. Afterwards the scheduler proceeds to schedule a DCI in PDCCH but without a PDSCH following that. The activation DCI includes the time and frequency resources that will be used in a periodic manner. To allocate the DCI the allocate_sps_grant method is called in the ue_cell_grid_allocator class. This method is responsible for allocating the DCI to the grid in a collision-free manner. After activation the second path is always called which ends with the allocation on the resource grid of the sps activated grant in a similar method as before. The difference here is that PDSCH must be also allocated since

it is there that the SPS transmission will happen. The SPS allocation only occurs if the TTI for which the scheduler operates matches the periodicity value of SPS. If not the UE is skipped.

```
alloc_outcome ue_cell_grid_allocator::allocate_sps_activated_grant(
    const ue_sps_activated_grant& grant){
  logger.info("Allocating grant for subsequent SPS");
  //Important checks for allocation
  srsran_assert(ues.contains(grant.user->ue_index), "Invalid UE
    candidate index={}", grant.user->ue_index);
  srsran_assert(has_cell(grant.cell_index), "Invalid UE candidate
    cell_index={}", grant.cell_index);

  //Extracting UE
  ue& u = ues[grant.user->ue_index];

 // Verify UE carrier is active.
  ue_cell* ue_cc = u.find_cell(grant.cell_index);
  if (ue_cc == nullptr or not ue_cc->is_active()) {
    logger.warning("PDSCH allocation failed. Cause: The ue={} carrier
    with cell_index={} is inactive",
                   u.ue_index,
                   grant.cell_index);
    return alloc_outcome::skip_ue;
  }

  const ue_cell_configuration& ue_cell_cfg = ue_cc->cfg();
  const cell_configuration&    cell_cfg    = ue_cell_cfg.
    cell_cfg_common;
  search_space_id                      ss_id =  MIN_SEARCH_SPACE_ID;
  const search_space_info&     ss_info    = ue_cell_cfg.search_space(
    ss_id);
  const coreset_configuration& cs_cfg     = *ss_info.coreset;
  dl_harq_process&             h_dl        = ue_cc->harqs.dl_harq(grant
    .h_id);
  ...
  return alloc_outcome::skip_ue;
```

4.2.3 ue_cell_grid_allocator

The SPS implementation of the UE_cell_grid_allocator was done based on the already existing dynamic downlink allocate grant function.

As already discussed, two different methods were created at this class. One for allocating the SPS activation grant and one for allocating the SPS resources after activation. What follows is the general framework of the alloc_grant method in SRSRAN. Based on it with some modifications the 2 new methods were implmented.

The overall aim of the function is to allocate the received scheduling grant to the resource grid, while aiming for the most efficient utilization.

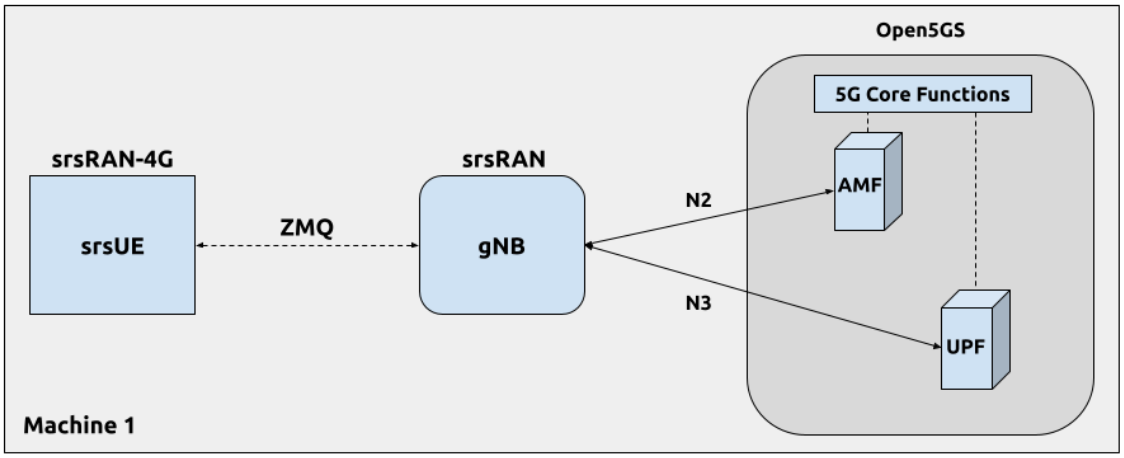

Figure 4.2: ZMQ-based RF driver[4]

The preliminary setup includes initializing the ue_cell_grid_allocator object and adding a new cell using the add_cell function along with PDCCH, UCI and cell resources. Finally the DL and UL counters are reset.

After the allocate grant function verifies that the UE candidate is valid for grant allocation, while also checking whether the UE and cell are present and active, it assigns the UE to that cell. It also checks whether the maximum number of PDCCH allocation attempts have been reached or not, and appropriately stops the allocation for this slot

The next step is to fetch the UE and cell configurations. It then searches for a valid SearchSpace candidate. Next it also determines the type of DCI message. If the space is valid new checks are performed to confirm that sufficient PDSCH resources are available to be allocated. If that's the case, it allocates resources to the PDCCH, which is mostly used for transmitting control information to the UE.

Next the function allocates the PDCCH and UCI, computes both MCS and TBS for DL transmission, fills the DL PDDCH DDCI PDU and the PDSCH PDU and finally saves the allocated parameters in a HARQ process.

The most important modification according to the already implemented method of SRS's gNB was the use of different RNTI types for the activation DCI message. Since SPS uses RNTI type of CS-RNTI and not C-RNTI a separate method for constructing the DCI message was implemented that used the CS-RNTI.

4.3 UE SPS implementation overview

Currently the 4G SRSRAN suite includes three different applications a core network, an eNodeB, and a UE implementation. Those apps, were developed in C++ and support LinuxOS as an operating system. They were developed to run using Software Defined Radio(SDR) devices for over-the-air transmissions. The suite also includes a virtual radio which is implemented using ZeroMQ, as an alternative networking library to transfer radio samples between applications.

In our specific niche, ZeroMQ played a critical role in the development, debugging

and testing of SPS. In particular, srsUE which is "a full-stack SDR 4G UE application with 5G features" is the app used in the development of this project, with multiple additions being made to support SPS.

4.3.1 RRC Layer modifications

One of the first changes actually implemented was configuring the UE RRC layer to acknowledge the existence of configured scheduling. Parameters such as periodicity & nrofHARQ-Processes etc. are directly used by the SPS algorithm and needed to be included in the UE code stack.

The RRC layer of UE receives an RRC reconfiguration message from gNB which includes the SPS config. It will then unwraps the message and then pass the information to the MAC layer using an RRC to MAC interface. Then the NDS class located in the MAC layer will receive the message and mainly use it to configure UE periodicity. The MAC layer then needs to handle transmission and reception without dynamic scheduling(through the NDS class created) as per TS 138321 5.8[13].

However the SRSRAN-4G UE code stack didnt include such functionality and needed to extend the stack to support it.

For this reason we created the apply_sp_cell_init_dl_sps boolean function. The way it works is as follows:

- First it checks whether or not the downlink HARQ configuration was created. If not it will return false
- Next it checks for the existence of the MCS table within sps_cfg. If true the value the PDSCH MCS table is qam64LowSE as per spec
- If the MCS table is not present our addition was to create the 64qam table

The next step is creating a switch statement that will assign the periodicity. Both the assigned value and default value are 10ms.

Finally the periodicity is passed to set_sps_grant and a true value is returned to the function

```
/// apply sps_config
//Read sps config and notify of content in rest of the stack
bool rrc_nr::apply_sp_cell_init_dl_sps(const asn1::rrc_nr::sps_cfg_s&
    sps_cfg){

 logger.info("In apply init dl sps");
    dl_harq_cfg_nr_t dl_harq_cfg_nr;
        if (make_mac_dl_harq_cfg_nr_t(sps_cfg, &dl_harq_cfg_nr) ==
            false) {
          logger.warning("Failed to make dl_harq_cfg_nr config");
          return false;
        }
  logger.info("Before infprming the mac about the harq");
```

```
    mac->set_config(dl_harq_cfg_nr);
       if (sps_cfg.mcs_table_present) {
       phy_cfg.pdsch.mcs_table = srsran_mcs_table_qam64LowSE;
    }

  else{
    //Our thought here!!!
    phy_cfg.pdsch.mcs_table = srsran_mcs_table_64qam;
  }

  switch (sps_cfg.periodicity)
  {
  case sps_cfg_s::periodicity_opts::ms10:
    phy_cfg.pdsch.periodicity_ms = 10;
    break;
  case sps_cfg_s::periodicity_opts::nulltype:
    logger.warning("Warning while selecting sps periodicity");
    return false;
  default :
      phy_cfg.pdsch.periodicity_ms = 10;

  }

 logger.info("Setting the grant toi the mac mainly the periodicity");
  mac->set_sps_grant(sps_cfg.periodicity);

  return true;
  //TODO::
}
```

Per 3GPP TS 38.331[14], the UE requires the following:

- periodicity ENUMERATED ms10, ms20, ms32, ms40, ms64, ms80, ms128, ms160, ms320, ms640,spare6, spare5, spare4, spare3, spare2, spare1,
- nrofHARQ-Processes INTEGER (1..8)
- n1PUCCH-AN PUCCH-ResourceId OPTIONAL, – Need M
- mcs-Table ENUMERATED qam64LowSE OPTIONAL, – Need S

CS-RNTI or Configured Scheduling - Radio Network Temporary Identifier is, as the name suggests, an identification number used to differentiate one UE's traffic to the eNB/gNB from another's. There is a large number of different RNTIs each with its own use case but in our specific scenario we will exclusively use CS-RNTI.

Going more in depth regarding CS-RNTI, it is actually used in conjuncture with DCI messages. One can think of CS-RNTI as a CRC mask that is used to decode the DCI message a UE receives from the gNB. For every DCI message a unique CS-RNTI ID is needed to actually decode it. This implies that the gNB has a list of all available CS-RNTIs thus "knowing" the amount of users needed to be scheduled

4.3.2 MAC Layer modifications

The main class which we worked on for the MAC layer is proc_nds_nr.cc. It is used in order to handle the scheduling functionalities of non-dynamic scheduling, particularly configured grant and SPS.

As presented before, one of the main qualifiers of whether a transmission is viable for SPS is the periodicity and whether it's a transmission opportunity or not.

```
void  non_dynamic_scheduling_nr::handle_downlink_grant(const uint32_t
    periodicity){
    logger.info("handling the sps grant in the MAC layer");
std::lock_guard<std::mutex> lock(mutex);

sps_grant.periodicity = periodicity;
sps_initiated=true;

}
void non_dynamic_scheduling_nr::activate_sps(){
if (sps_is_activated) {
    sps_is_activated=false;
}else{
    sps_is_activated=true;
}
bool non_dynamic_scheduling_nr::is_sps_transmission_opportunity(
    uint32_t tti){
    //dummy
    return (tti % 7 == 0);

    //return (tti % sps_grant.periodicity == 0);

    //TODO:
    // add the actual equation

}
```

As shown in the code snippet above, we had to make some concessions pertaining to the transmission opportunity. In the current implementation we have used a pre-configured value, but in reality, if going by the technical specifications, transmission opportunities are calculated using the following formula[13]:

$$(numberOfSlotsPerFrame * SFN + slotnumberintheframe) = [(numberOfSlotsPerFrame * SFNstarttime + slotstarttime) + N * periodicity * numberOfSlotsPerFrame/10] modulo (1024 * numberOfSlotsPerFrame)$$

While the proc_nds class is able to handle both configured grant and SPS, we will focus solely on the additions needed for SPS.

After configuring the HARQ process and ensuring that they are given a HARQ process

ID, the UE begins looking for an appropriate transmission opportunity.

If an appropriate transmission opportunity is found, a method is provided which checks the time the grant arrived and provides the value for the HARQ. Finally the HARQ process ID is calculated, based off of the current TTI and periodicity.

4.3.3 PHY layer modifications

Additional work was needed in the PHY layer of the UE code stack, in order for the UE to both identify the SPS activation grant as well as any following SPS transmissions. To this end, modifications were done to the state class which is responsible for managing the pending grant list of the UE along with the cc_worker class which handles the reception and decoding of both DL and UL PHY channels of the UE. Particularly, the decode_pdcch_dl method was updated to search not only for PDCCH scrambled with C-RNTI but also with CS-RNTI. Furthermore, updates were done to the decode_pdsch_dl() method by adding the capability to also search for an SPS grant and not just exclusively dynamic grants. Special consideration was needed for the specific case of SPS resources being unutilized. As these resources are available in a periodic manner the gNB might or might not use them to send an amount of user data. Despite these resources not being used the UE still attempted to decoded them and sent an ACK message even though no data was sent. Extra attention to detail was needed here so that the UE could tell apart an empty transmission from a non empty one and not proceed with an acknowledgment in case of an empty transmission being detected.

5 Simulations and Results

In this chapter the results of simulation experiments will be presented. Overall there was a substantial reduction of DCI messages requirements achieving the primary goal of this implementation, which is to reduce the scheduling overhead that is inherent with transmissions such as VOIP.

The simulation environment, consists of 4 different components.

- Dockerized Open5GS: A containerized version of the Open5GS core network, providing the necessary 5G core network functions.
- gNodeB (gNB): Our modified version of the srsRAN gNB implementation, incorporating the SPS functionality.
- User Equipment (UE): The modified srsRAN UE implementation, capable of operating under SPS.
- Traffic Generator Console: Used to simulate realistic network traffic patterns between the gNB and UE.

This setup allows for a controlled scenario of packet traffic to occur. The main key metric that is presented in the graphs is the average number of required transmissions needed to send a specific amount of packets.

Our primary goal of reducing the signaling overhead was successfully achieved. Figures 5.1, 5.2, 5.3 and 5.4 illustrate the different scenarios. In total there were two scenarios, a 2000 packet and a 5000 packet one. As evident from the graphs, SPS significantly reduced the signaling overhead, with an average reduction of 30% compared to dynamic scheduling.

One thing to mention however is the inconsistent latency presented when transmitting packages. Whilst SPS successfully reduced the overhead requirements, there wasn't a similar reduction in latency between packets. In specific cases concerning the hardware the simulation was ran on latency was slightly increased when using SPS. This was due to SPS opportunities colliding with the Channel State Information-Reference Signal which are also periodically transmitted. However, even in cases of no such collisions the latency reduction was not so apparent. This could be explained due to limitation presented by 3GPP, particularly the requirement that a UE must continuously look for scheduling grants which leads to similar latency to dynamic scheduling. The UE has to search for PDCCH in every TTI because the gNB might either decide to deactivate the SPS or schedule the

UE with a dynamic grant for this specific occasion. One could see a substantial latency reduction in the case of UL configured scheduling where the Scheduling Request is not needed.

The simulation results showcase the potential of our SPS implementation to significantly reduce control signaling overhead while at the minimum maintaining other key metrics. The reduction in signaling overhead aligns with our primary objective and signifies that SPS can effectively address challenges with large scale 5G networks. Whilst there were no latency improvements, the increased throughput and efficient resource utilization could potentially translate to future real-world deployments.

In the graphs presented below, the bars showcase the number of transmissions on each slot, whilst in red is the average number of PDCCH transmissions per second. For the 3rd and 4th graphs, the sampling rate is 10 seconds due to the large amount of data that needed to be processed from the log files.

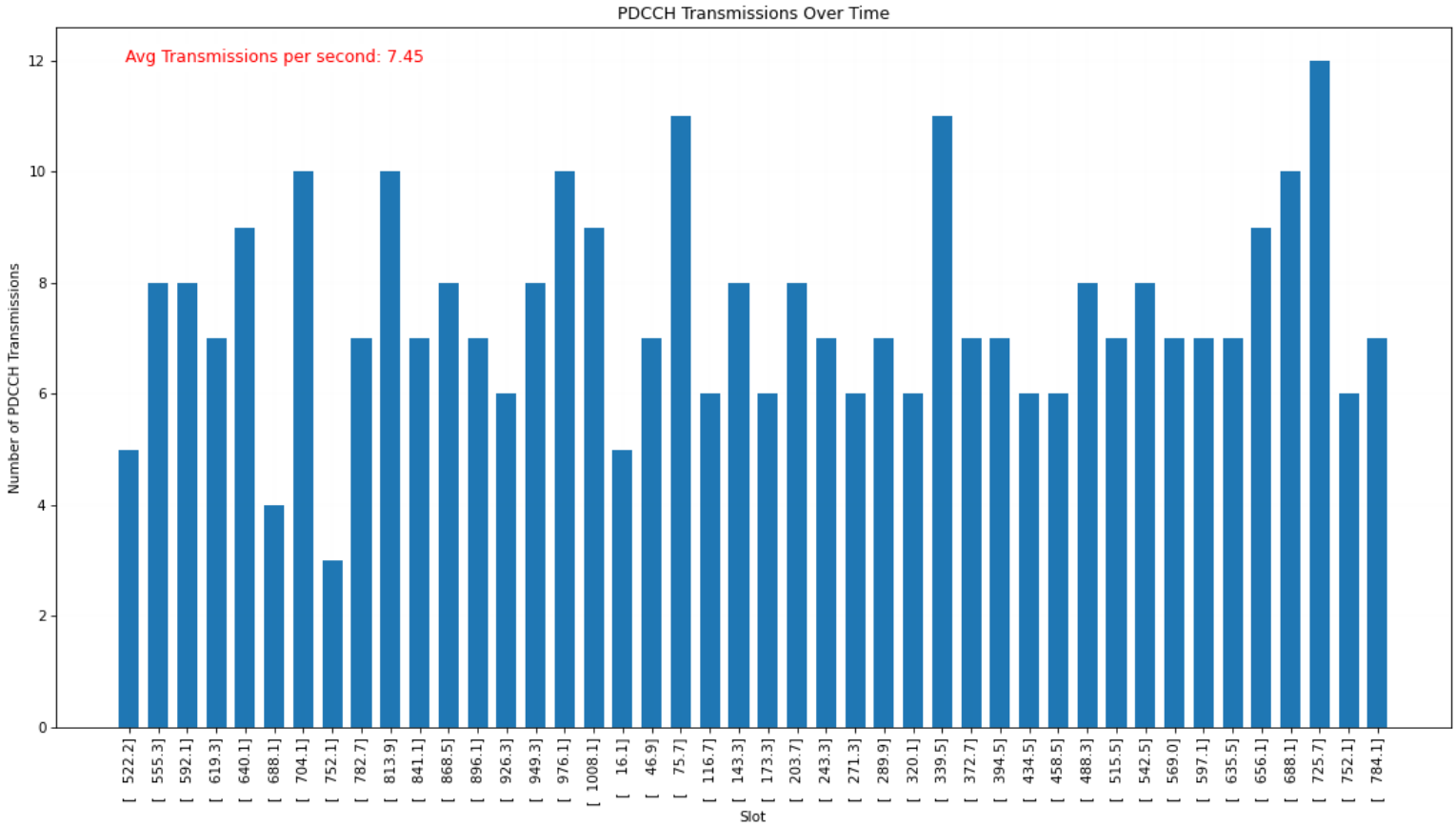

Figure 5.1: 2000 Packets SPS experiment with an Average Number of transmissions per second equal to 7.45

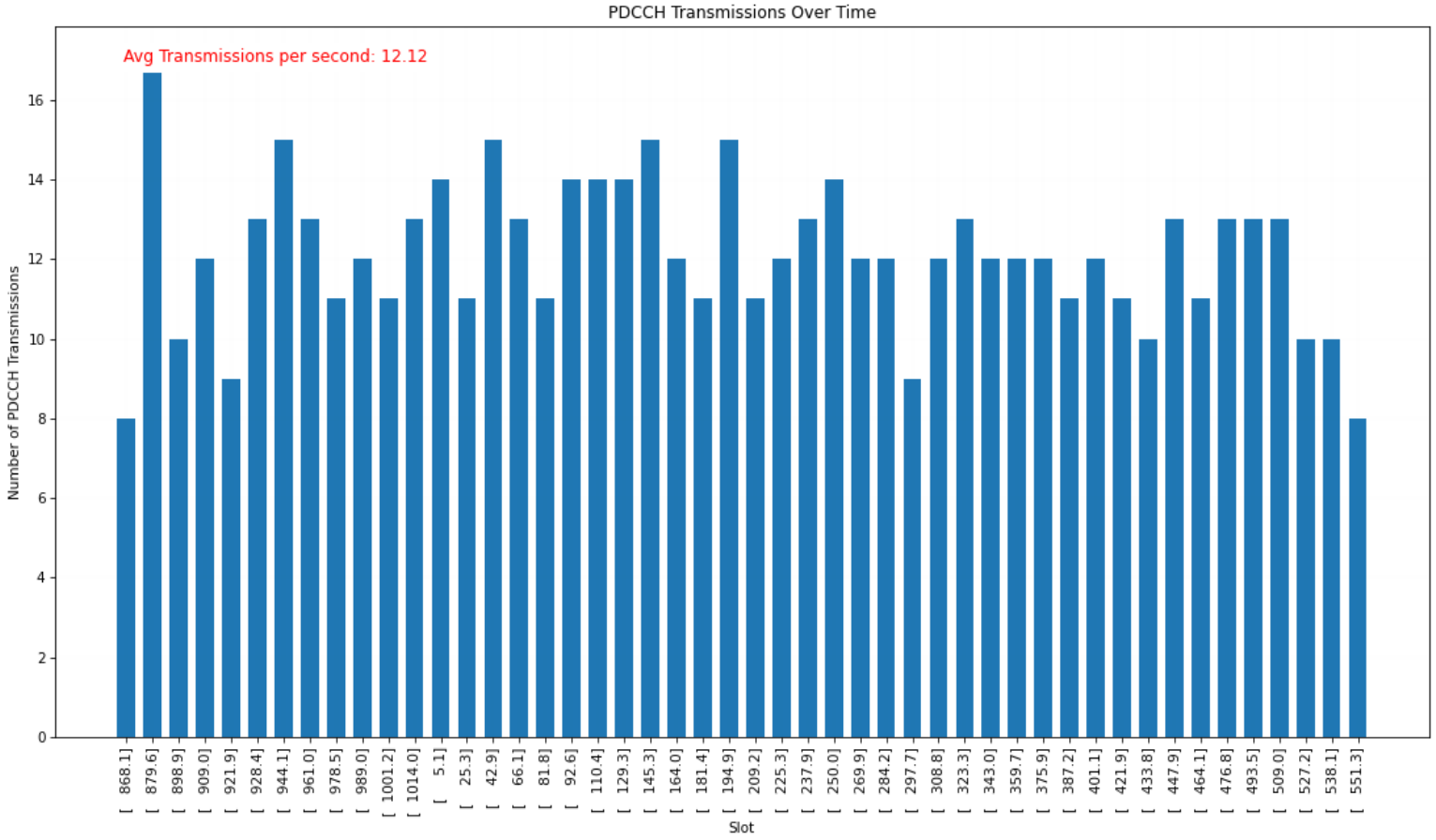

Figure 5.2: 2000 Packets Dynamic Scheduling with an Average Number of transmissions per second equal to 12.12

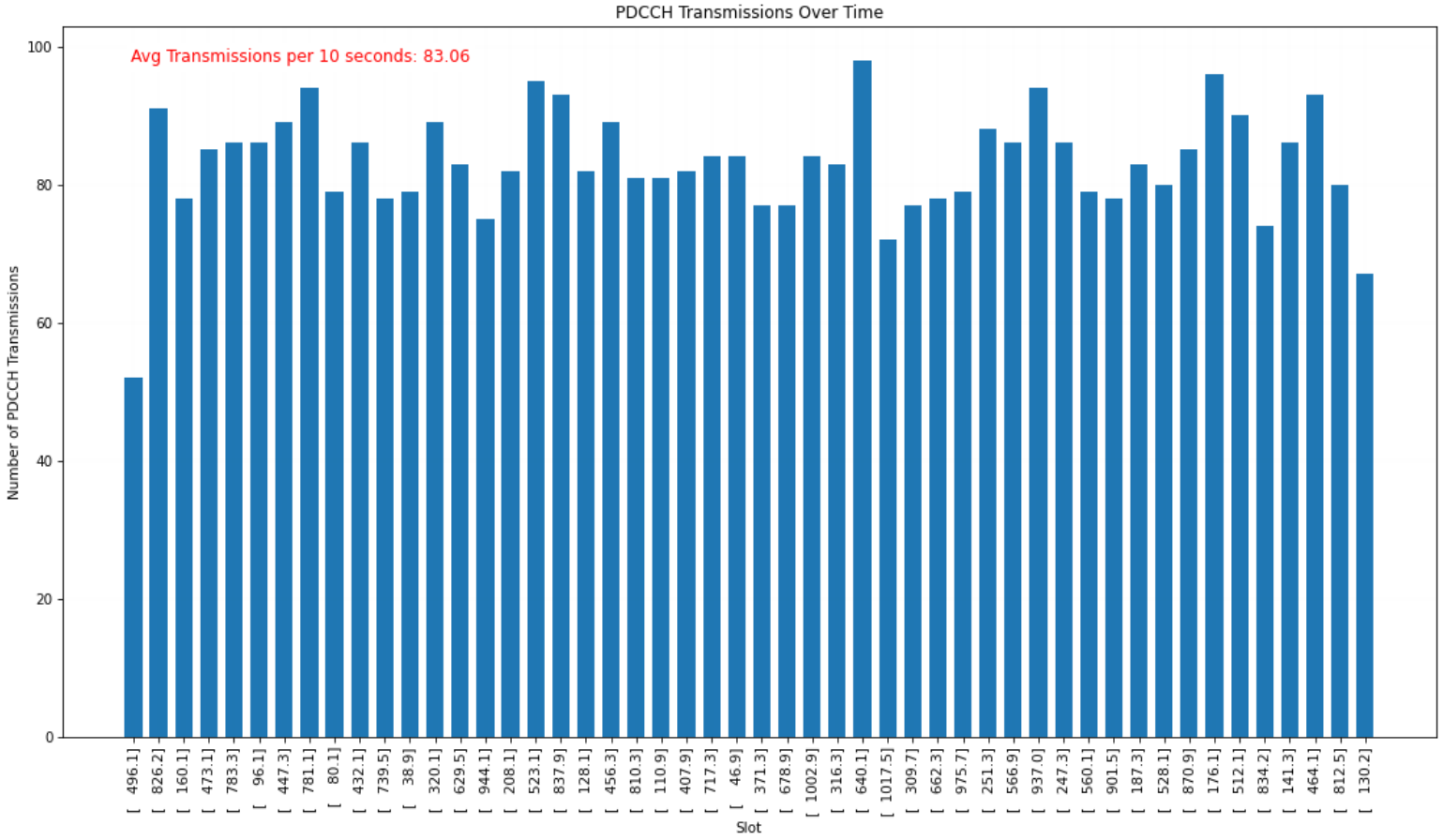

Figure 5.3: 5000 Packets SPS Average Number of transmissions per 10 seconds equal to 83.06

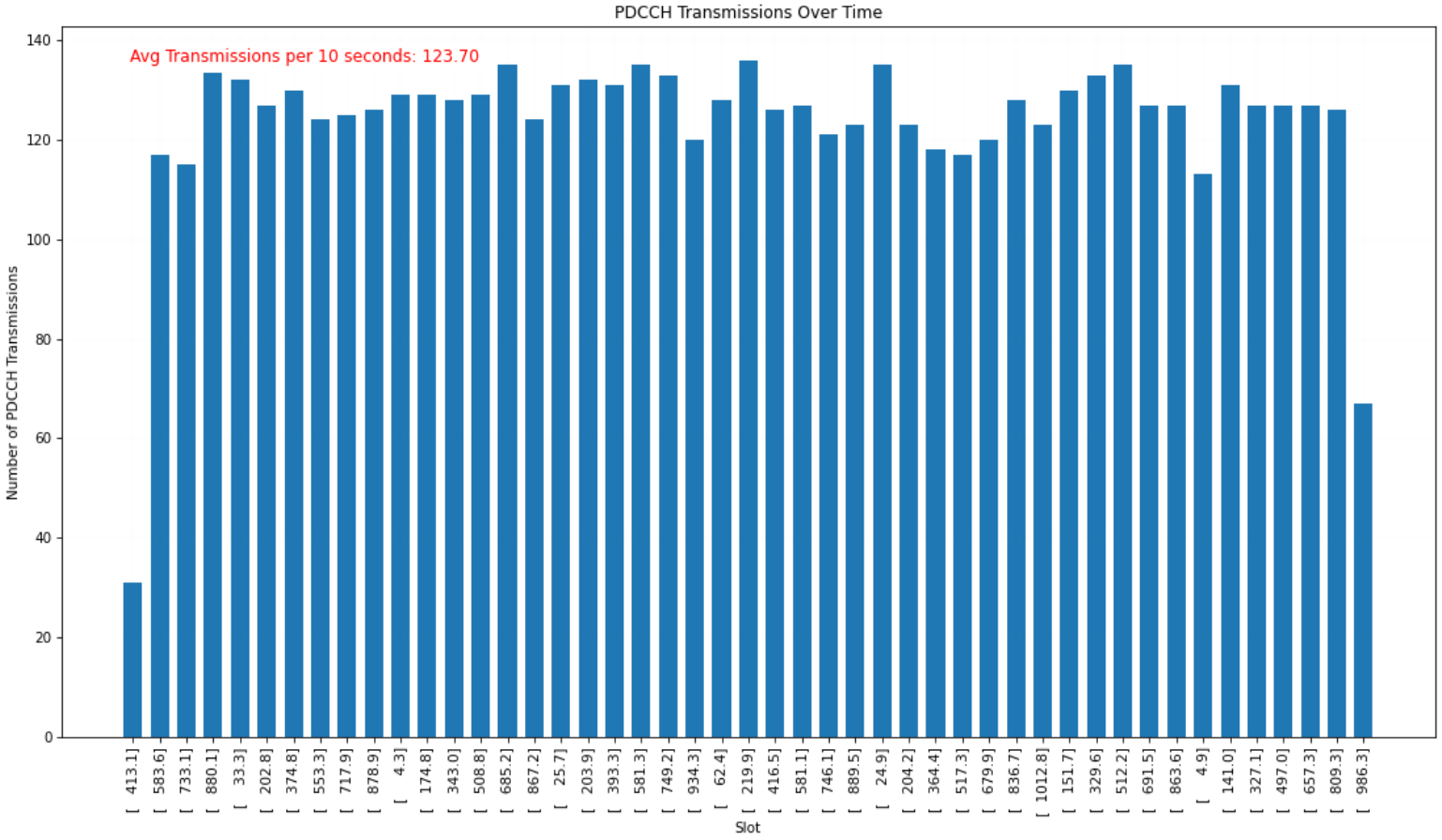

Figure 5.4: 5000 Packets Dynamic Scheduling Average Number of transmissions per 10 seconds equal to 123.70

6 Conclusion and future work

This thesis has presented a comprehensive overview and analysis of Semi-Persistent Scheduling for Beyond 5G networks, demonstrating the potential in resource allocation optimization and overhead reduction. Through changes to the overall project code stack, including the PHY, MAC and RRC layers we have developed a functional prototype of SPS protocol within the open-source srsRAN framework.

The primary objective which was the reduction of scheduling overhead was achieved successfully, with simulation results showing a significant decrease in the number of DCI messages required when compared to dynamic scheduling.

During the implementation several challenges arose, particularly in adapting the already existing srsRAN stack to support SPS functionality. While concessions had to be made, the complexity of this project showcased the difficulty of implementing a new protocol to an already existing code stack whilst providing insight to the importance of a well defined standard, such as the one provided by 3GPP.

Finally as mentioned, there are several limitations and concessions that had to be made in order to provide a working prototype. First and foremost, the fact that the current implementation is optimized for one UE, limits its potential for real world applications. Additionally, periodicity is statically defined rather than dynamically adapting to match current network traffic requirements.

Future improvements can focus in several fields such as:

1. Multiple UE support, each with different periodicities and resources requirements, necessitating the development of more complex scheduling algorithms to ensure fairer resources allocation

2. Dynamic Periodicity based on real world conditions such as traffic patterns and network conditions. This could lead to an improvement to resource utilization.

3. Integration with other network features: Exploring how SPS will be integrated with already existing network infrastructure as well as Ultra-Reliable Low-Latency Communications

4. Standardization and interoperability: Ensuring that the implementation is aligned with 3GPP specifications and interoperable with commercial network equipment.

In conclusion this paper provides a much needed foundation for future research into SPS for Beyond 5G networks. Ongoing research and development in the scheduling part of

cellular communications has the potential to provide significant advances to performance, efficiency and capabilities of next-generation networks.

Bibliography

[1] Ye Feng, Ampalavanapillai Nirmalathas, and Elaine Wong. A predictive semi-persistent scheduling scheme for low-latency applications in lte and nr networks. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019.

[2] 3GPP MCC Alain Sultan. Ultra reliable and low latency communications. https://www.3gpp.org/technologies/urlcc-2022.

[3] Dajie Jiang, Haiming Wang, Esa Malkamaki, and Esa Tuomaala. Principle and performance of semi-persistent scheduling for voip in lte system. In *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, 2007.

[4] L.L. Peterson, O. Sunay, and B.S. Davie. *Private 5G: A Systems Approach*. CC BY-NC-ND 4.0, 2023.

[5] srsRAN. srsran, 2022. https://www.srsran.com/5g.

[6] Qing He, György Dán, and Georgios P. Koudouridis. Semi-persistent scheduling for 5g downlink based on short-term traffic prediction. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, 2020.

[7] Suvra Sekhar Das, Priyangshu Ghosh, and Prabhu Chandhar. Estimation of effective radio resource usage for voip scheduling in ofdma cellular networks. In *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*, pages 1–6, May 2012.

[8] Syed Ghazanfar Salari. Control traffic overhead for voip over lte. 2012.

[9] Aymeric de Javel. *5G RAN : implémentation de la couche physique et découpage du réseau*. PhD thesis, 2022. Thèse de doctorat dirigée par Rougier, Jean-Louis et Martins, Philippe Réseaux, informations et communications Institut polytechnique de Paris 2022.

[10] Douglas Morais. *Key 5G Physical Layer Technologies: Enabling Mobile and Fixed Wireless Access*. 01 2020.

[11] José Pedro Nogueira Rodrigues. Open source non-public 5g networking in licensed and unlicensed bands. Master's thesis, University of Porto, 2023.

[12] Jaeku Ryu. Sharetechnote. https://www.sharetechnote.com/html/5G/5G_ConfiguredScheduling.html.

[13] 3GPP. *Medium Access Control (MAC) protocol specification TS 38.321*. 3rd Generation Partnership Project (3GPP), 01 2021. Release 15, Version 15.3.0.

[14] 3GPP. *Radio Resource Control (RRC) protocol specification TS 38.331*. 3rd Generation Partnership Project (3GPP), 07 2020. Release 16, Version 16.1.0.

A Appendix

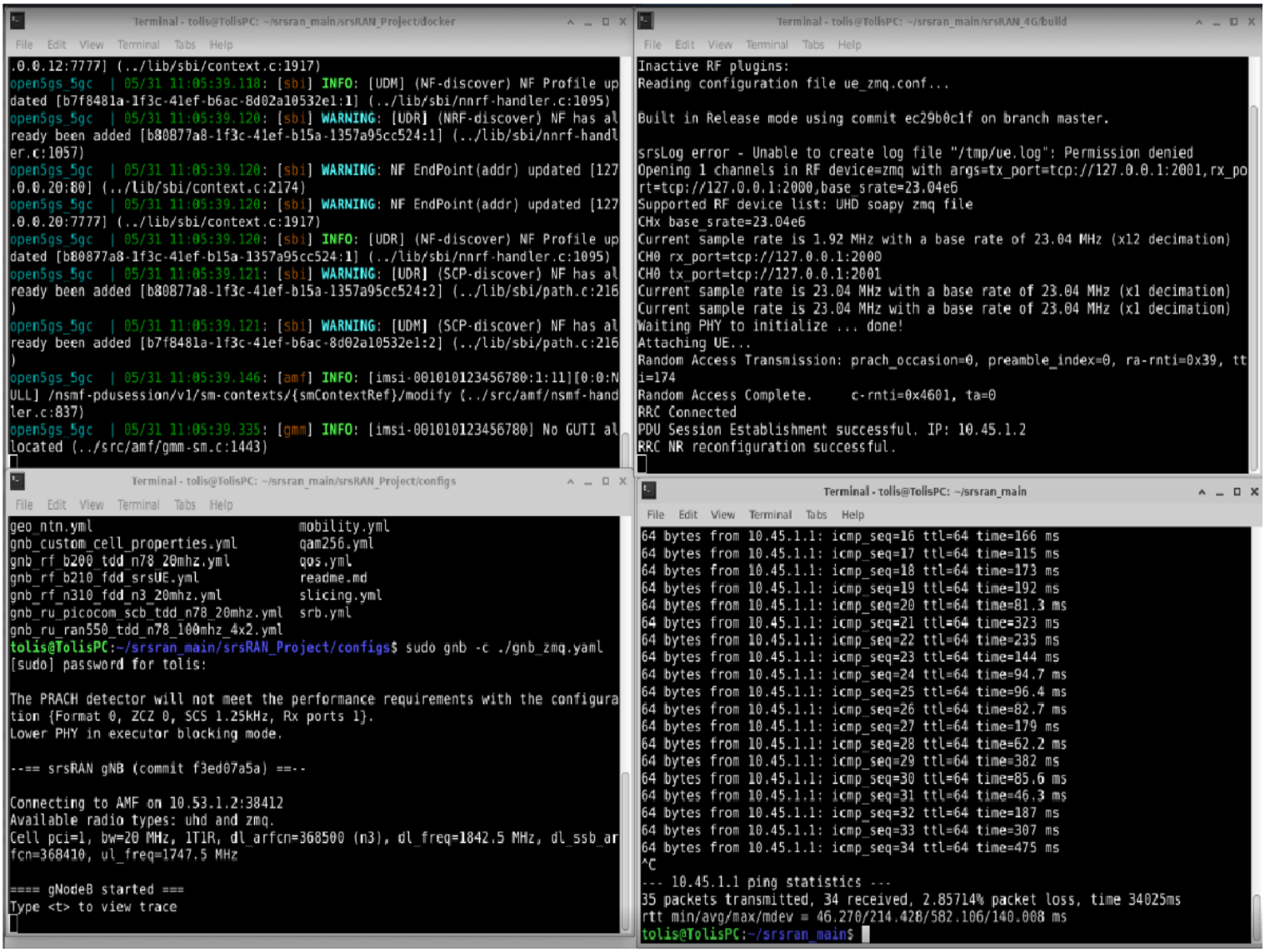

Figure A.1: Simulation Environment

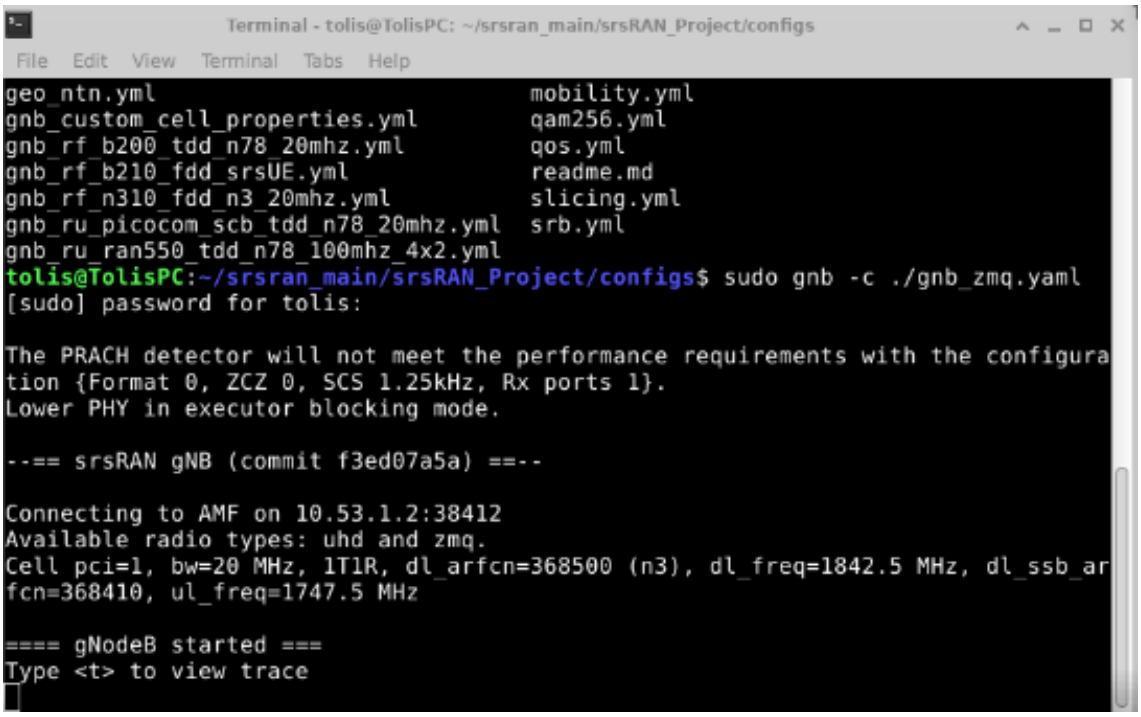

Figure A.2: GnB Console

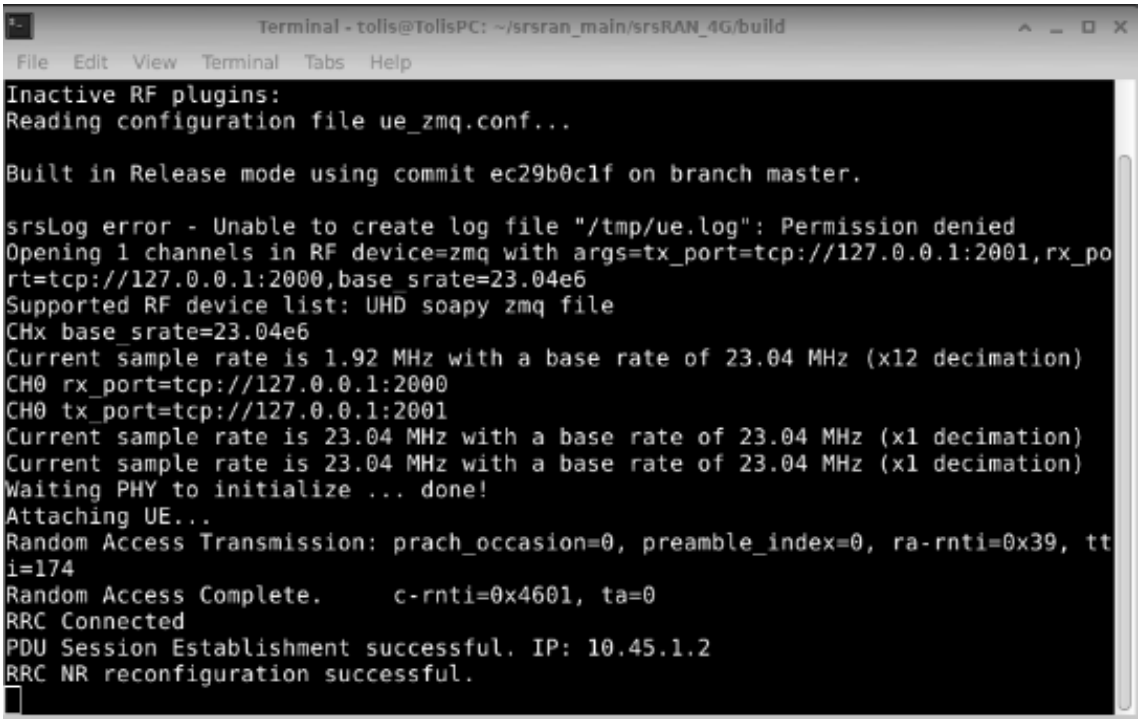

Figure A.3: UE Console

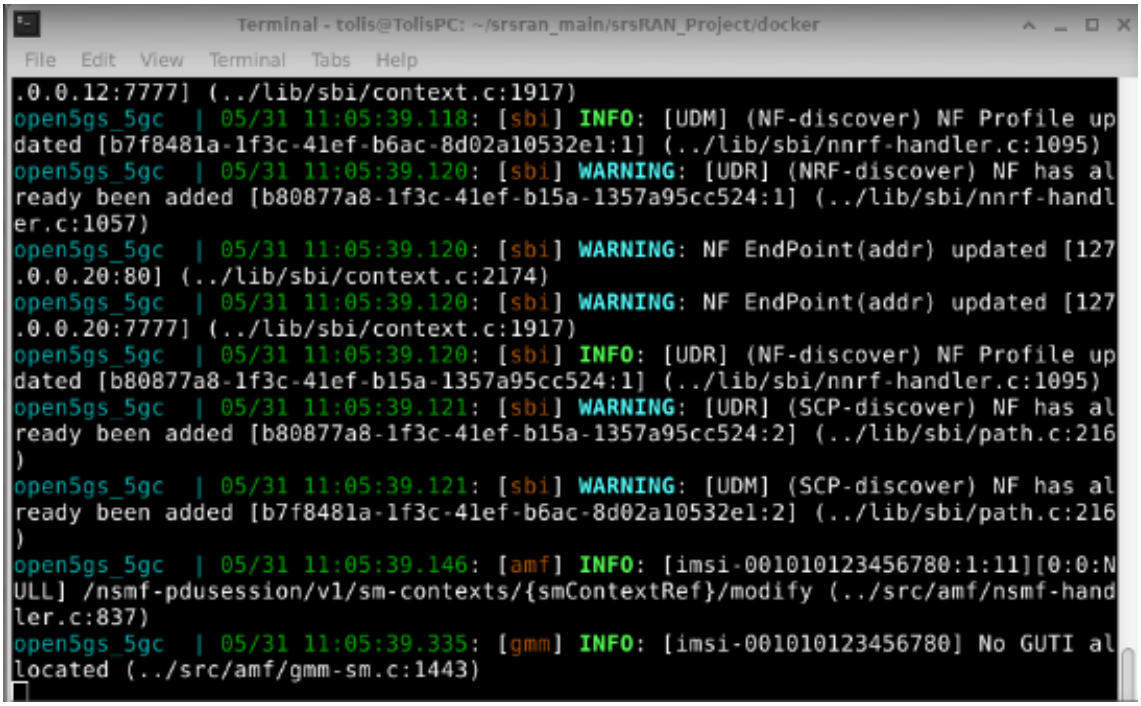

Figure A.4: Dockerized Open5GS

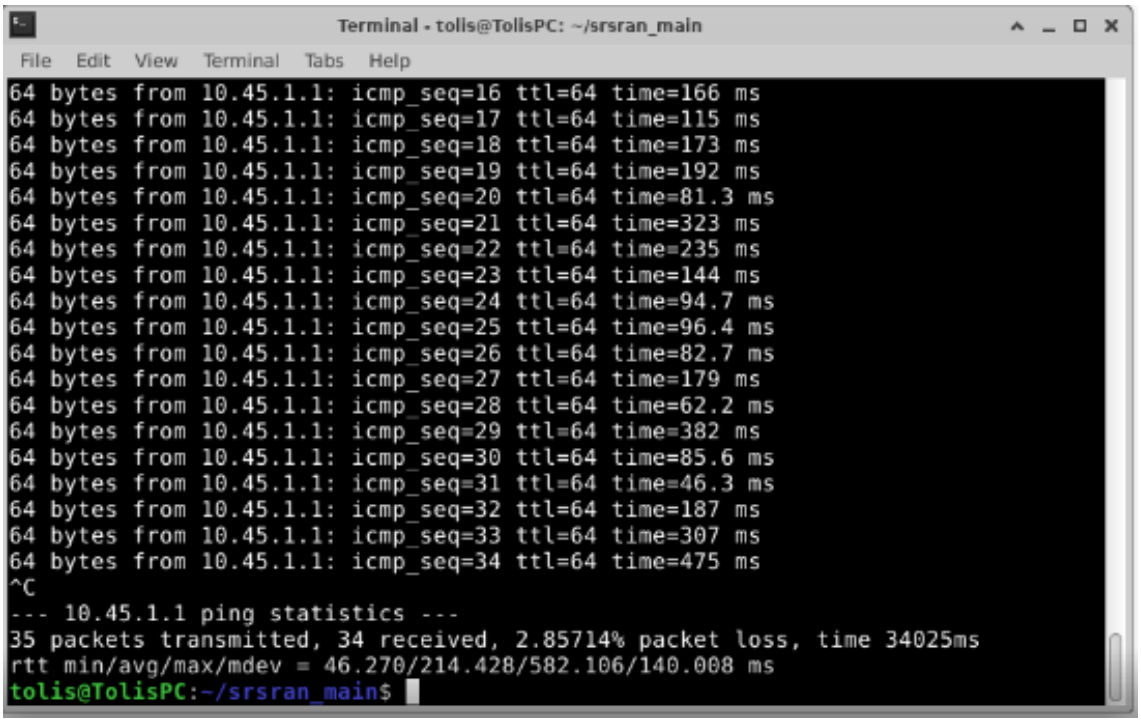

Figure A.5: Simulation Environment