Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Thesis in ICT for internet and multimedia

Implementation, Design and Performance Evaluation of Scheduling Protocols for Non-Terrestrial Networks (NTN)

Master Candidate

Mattia Figaro

Student ID 2089060

Supervisor

Prof. Marco Giordani

University of Padova

Co-supervisors

Prof. Michele Zorzi

University of Padova

Dr. Matteo Pagin

University of Padova

Academic Year
2023/2024

A mamma e papà, per avermi sempre sostenuto in tutto.
Alle nonne e alle zie, per aver sempre saputo
consigliarmi nel momento giusto.
A tutta la mia famiglia, per avermi fatto sentire a casa.
A Giada, per essere riuscita a tirar fuori il meglio di me.
A Ross, Sara, Prince, Auro e Fede, per aver
condiviso anni indimenticabili.
A Giovanni, Marco, Manuel, Valentina e tutti i miei amici,
per essere cresciuti assieme.

Abstract

The consolidation of fifth generation cellular system (5G) and the imminent arrival of the sixth generation mobile systems (6G) have improved internet coverage, especially in the urban areas. These advancements promise to revolutionize various sectors, including healthcare, transportation, and entertainment, by enabling faster data transmission and more reliable connections. On the other hand, this will lead to new challenges to be faced, like extending coverage to remote areas and enhancing disaster response and recovery. In addressing this, the research community is focusing on Non-Terrestrial Networks (NTN), which utilize Unmanned Aerial Vehicles (UAVs), High Altitude Platforms (HAPs), and satellites as aerial or space gateways. These platforms are positioned to play a crucial role in achieving the ambitious goals of 5G and beyond, especially in areas where terrestrial infrastructure is lacking or impractical. Despite their potential benefits, NTN present unique engineering challenges: factors such as the high mobility and long propagation delays inherent to satellite communications, and the unpredictability of intermittent connectivity necessitate innovative solutions in network design and management. Efficient scheduling protocols are particularly critical in these networks, as they directly impact the quality of service and overall network performance. Acknowledging these challenges, this thesis adopts a comprehensive approach into the design and optimization of NTN. Notably, we explore the performance of 3rd Generation Partnership Project (3GPP)-like terrestrial scheduling, and propose new solutions specifically tailored to the NTN satellite scenario. To evaluate the performance of our models, we use the ns-3 full-stack end-to-end network simulator, which guarantees that numerical results are realistic. Specifically, we extend the *ns3-ntn* module, that currently implements the NTN channel model and physical layer, to simulate scheduling. First, the work focuses on the weaknesses of User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) in scenarios with large propagation delay. Then, disadvantages of Grant Based Scheduler (GBS) approach will be highlighted and discussed. For this reason, a dedicated Semi Persistent Scheduling (SPS) approach will be implemented and tested in periodic and aperiodic traffic scenarios.

Contents

List of Figures

List of Acronyms

NTN Non-Terrestrial Networks

4G fourth generation mobile systems

5G fifth generation cellular system

6G sixth generation mobile systems

3GPP 3rd Generation Partnership Project

Gbps Gigabits per second

VSAT Very Small Aperture Terminals

2D two-dimensional

3D three-dimensional

UAVs Unmanned Aerial Vehicles

HAPs High Altitude Platforms

HAP High Altitude Platform

UE User Equipment

NYU New York University

UDP User Datagram Protocol

TCP Transmission Control Protocol

IoT Internet of Things

GEO Geostationary Earth Orbit

MEO Medium Earth Orbit

LEO Low Earth Orbit

RTT Round Trip Time

NR New Radio

BS Base Station

MEC Mobile Edge Computing

Tx transmitter

Rx receiver

UL Uplink

DL Downlink

RRH Remote Radio Heads

ICI Inter Carrier Interference

SPS Semi Persistent Scheduling

TA Timing Advance

DMRS Demodulation Reference Signal

HARQ Hybrid Automatic Repeat Request

MAC Media Access Control

RLC Radio Link Control

RACH Random Access Channel

LOS Line-of-Sight

TDD Time Division Duplexing

FDD Frequency Division Duplexing

PAPR Peak-to-Average Power Ratio

PTRS Phase Tracking Reference Signal

RO RACH occasion

RAR Random Access Response

RRC Radio Resource Control

gNB gNodeB

GNSS Global Navigation Satellite System

DRX Discontinuous reception

PDCCH Physical Downlink Control Channel

SR Scheduling Request

DCI Downlink Control Information

LTE Long Term Evolution

PDSCH Physical Downlink Shared Channel

PUSCH Physical Uplink Shared Channel

RETX retransmission

ACK acknowledgment

SYN synchronization

RTO retransmission time-out

TO time-out

QoS Quality of Service

ULSCH Uplink Shared Channel

BSR Buffer Status Report

PDU Protocol Data Unit

AM Acknowledged Mode

UM Unacknowledged Mode

PDCP Packet Data Convergence Protocol

SDU Service Data Unit

MC multi-connectivity

V2X Vehicle-to-Everything

PDR Packet Delivery Ratio

PHY physical

SAP Service Access Point

SINR Signal to Interference plus Noise Ratio

TDMA Time Division Multiple Access

OFDM Orthogonal Frequency Division Multiplexing

TB Transport Block

SNR Signal to Noise Ratio

RH Remote Host

APP Application

CQI Channel Quality Indicator

TTI Transmission Time Interval

SI Send Interval

GBS Grant Based Scheduler

TR Technical Report

1
Introduction

Nowadays we are living in the middle of the 5G era, and the technological advancement it brings is increasingly becoming part of everyday activities. 5G was developed by the 3GPP, while its global launch began in 2018. It represents a significant advancement in cellular network technologies with respect to its old brother fourth generation mobile systems (4G), since it introduces a new set of standards for devices and applications [1]. These latter includes enhancement to speed connection and bandwidth, achieving data transmission up to to 10 or even 20 Gigabits per second (Gbps), which is more than 100 times faster than 4G, latency reduction, meaning data can travel from one location to another much faster, and also enhancement to connectivity [2]. This is just the latest step in a longer development progress of cellular communication systems, as shown in Figure 1.1.

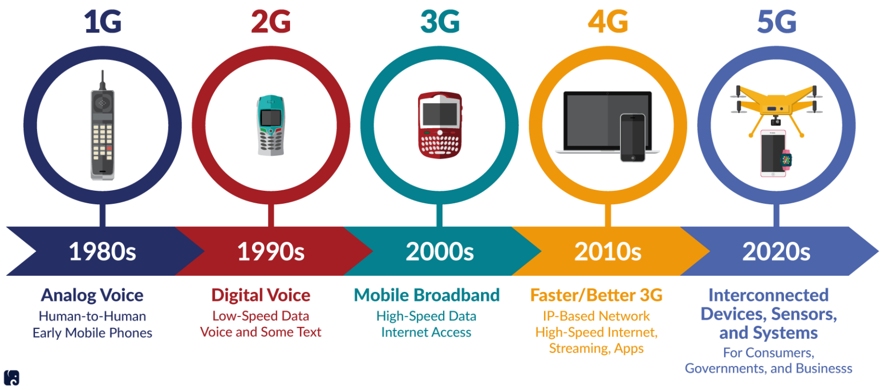


Figure 1.1: Development of cellular systems over the years. Figure from [3].

5G networks aim to deliver 1000 times higher mobile data volume per area, 100 times the number of connected devices, 100 times higher user data rate, ten times longer battery life for low-power massive-machine communications, five times reduced end-to-end latency, and ten times longer battery life for low-power massive-machine communications [4]. Although this development already seems to be yielding great results, researchers around the world are already working on the definition and development of what will be the future generation of cellular systems: the 6G. As shown in Figure 1.2, this is necessary not only to increase the comfort and technological services of everyday use, but also to keep pace with the explosive growth of data worldwide [5].

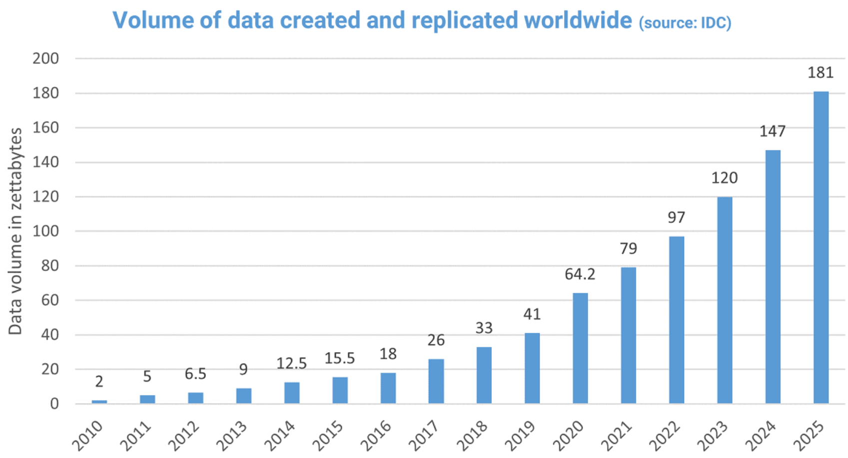


Figure 1.2: Volume of data created and replicated worldwide. Figure from [6].

However, this growth is not evenly distributed across all parts of the world. The *digital divide* represents a significant challenge in today's interconnected world, highlighting the inequity in access to high speed internet among different populations [7]-[8]. This gap does not only affects individuals' ability to participate in the global economy but also exacerbates existing societal inequities, such as the wealth gap. New and better ways to communicate and work efficiently are essential for addressing the digital divide, particularly in the context of cellular communication systems. NTN could act as a promising solutions to this problem [9]. In a nutshell, they represent a significant progress in the telecommunication systems, since they offer a novel approach to delivery connectivity beyond the limitations of traditional terrestrial networks. In particulars, satellite networks cover vast geographical areas, including remote rural locations and oceans, where traditional fiber-optic cables cannot reach [10].

Moreover, they are particular effective in remote or undeserved areas, filling the connectivity gaps left by traditional ground-based telecommunications infrastructure. This is made possible by their resilience and reliability, offering an alternative path for communication, in particular during natural disasters or other disruptions to terrestrial networks. This resilience is crucial for maintaining critical communications in times of need [11]. Furthermore, they play a crucial role in expanding the the range of cellular applications. The advent of 5G and upcoming 6G technologies opens up new possibilities for integrating NTN into the broader cellular communication infrastructure [12]. However, realizing the full potential of NTN requires addressing several challenges, including regulatory hurdles, the high costs associated with launching and maintaining satellites or High Altitude Platform (HAP)s, and the need for compatible devices and infrastructure on the ground. Collaboration between governments, private sector entities and researchers will be essential to overcome these barriers and ensure that NTN contribute effectively to bridging the digital divide.

The recent technological development has led to greater integration between terrestrial and non-terrestrial technologies to enable more advanced use cases. Among these, there are communication resilience and service continuity to help existing Base Station (BS)s, global satellite overlay, ubiquitous Internet of Things (IoT) broadcasting, advanced backhauling and energy-efficient hybrid multiplay. On the other hand, despite the current research in this field, there are still several open challenges to solve to achive a proper functioning of these systems. For instance the channel modeling, spectrum co-existence, physical (PHY) layer procedures, Hybrid Automatic Repeat Request (HARQ), synchronization, initial access, mobility and costellation managements, higher-layer design and architecture technologies. They are mainly due to the different characteristics that a non-terrestrial system has compared to a terrestrial one. Among these, there are for example the great lengthening of the distances that causes a large growth of the propagation delay, and the appearance of new disturbance phenomena, such as larger *path loss*, *atmospheric absorption* and *ionospheric/tropospheric scintillation*. The research in this area is therefore still open. The University of Padua, represented by the SIGNET research group, has been working on these issues for many years. Initially with the development of the *ns3-mmwave* [13] module for the simulation of 5G cellular networks operating at mmWaves. Continuing then with the expansion of the latter into a new version dedicated to non-terrestrial environments, namely the *ns3-ntn* module [14]. Currently, a dedicated channel model has been created following the indications in the 3GPP Technical Report (TR)s. However, much more work is needed to obtain a module that can make the 5G New Radio (NR) stack [15] work properly in a non-terrestrial environment. The work of this thesis is therefore focus on the study, design and implementation of scheduling protocols for NTN. In particular, trying to understand what is the best way to schedule transmissions based on the traffic type of the application. The goal is in fact to understand how to adapt the protocols of the terrestrial 5G NR stack, for an environment that is not, characterized by much higher distances between the nodes of the system. Indeed, this leads to having propagation delays that are extremely higher than classic cellular systems, causing the need to develop more advanced mechanisms than the simple Timing Advance (TA).

1.1 History of NTN

The history of NTN starts in the 1990s, when Globalstar and Iridium Communications companies chose to invest time and resources on the ambitious project to deploy a global satellite communication network, capable of supporting low-bandwidth communication directly to specialized handsets [16]. On the other hand, the rapid expansion of terrestrial networks and the difficulty to achieve good performance in a short period of time, in particular on signal penetration and voice quality, hindered their commercial success [17].
The evolution towards the modern NTN starts from 3GPP: the famous global consortium responsible for defining the standards for mobile telecommunications, actually initiated its study on NTN in 2017. This initiative greatly shifted the focus towards the integration of satellites, aerial platforms and terrestrial networks within the broader cellular telecommunications framework. The 3GPP standard for NTN aims to provide wireless connectivity beyond the Earth's surface, encompassing various scenarios with different terminal types, frequency bands, services, and orbits. The key features that 3GPP wants to guarantee are broad coverage and versatility, integration with existing infrastructure, enhancements and updates, in particular on NTN-IoT and NTN-NR, and finally support for emerging technologies. Future 3GPP releases are expected to include specifications for Very Small Aperture Terminals (VSAT), which will enable more efficient satellite-to-user connections.

However, besides 3GPP, other companies are also engaging in research in this sector. In fact, in May 2019, SpaceX's Starlink project launched its initial 60 satellites, starting a new era in satellite communication. Currently, it is basically the world's first and largest satellite constellation using a Low Earth Orbit (LEO) to deliver broadband internet capable of supporting streaming, online gaming, video calls and more [18]. Nowadays, the satellite count has risen to 12000, with the aim to provide global internet coverage always better. Moreover, recently the need for seamless integration and orchestration across NTN has become increasingly present. For that reason, several collaboration and partnerships are being created, such as the one announced between SpaceX and T-Mobile in August 2022, with the aim to facilitate the direct-to-device connectivity, allowing smartphones to connect to satellites outside the reach of terrestrial networks [19].

For what regards the future of NTN, they will play a crucial role on the development of 6G, offering enhanced connectivity and addressing the limitations of terrestrial counterpart [20]. Further, more and more companies and research organisations are starting to invest in NTN, searching the better way to integrate them with the standard terrestrial network.

1.2 The thesis structure

This thesis focuses on the study, design and implementation of scheduling protocols for NTN, in particular, trying to understand what is the best way to schedule transmissions based on the traffic type of the application. To do so, an initial review of the *ns3-ntn* module will be done, trying to understand what are the limitations and improvements needed to the native GBS of the module, using UDP and TCP as transport protocols. In particular, the work will focus on understanding how long distances, and consequently large delays, present in satellite communications, compromise the proper functioning of these protocols. For the purpose, the ns-3 end-to-end full stack network simulator will be adopted, since its ability to offer a highly controlled and reproducible environment for studying system behavior under various conditions. Moreover, ns-3 supports both C++ and Python, allowing researchers to implement the simulation at low level for greater precision and efficiency and, at the same time, manage the results nimbly with Python. After this initial introduction to contextualise the entire work, in Chapter 2 there will be a generic overview on NTN, explaining what they are, their issues and possible hints for future works. Chapter 3, on the other hand, will be dedicated to a review of today's literature on scheduling protocols, in which the critical points and possible resolving approaches will be illustrated. Chapters 5 and 6 are instead dedicated to the implementation, design and evaluation of results. In particular, a dedicated SPS will be implemented and compared to the native GBS, in periodic traffic scenario, showing how the latter is extremely less convenient in terms of overhead. Furthermore, a new scheduling protocol, called *Smart Scheduler*, will be implemented for cases where the application may vary its traffic over time. The adaptability of the *Smart Scheduler* approach showed a fair balance with GBS in terms of latency and large advantages in terms of overhead, managing to decrease the PHY-throughput. Specifically in some cases, it was possible to achieve a 10% saving on PHY-throughput consumption. Finally, all the results achieved will be summarised in Chapter 7 with any further comments.

2

Non-Terrestrial Networks

2.1 NTN Overview

While we are still living in 5G era, many organizations and researchers are already investigating the enabling technologies for the 6th generation of cellular systems. NTN is considered one of these technologies, since they can play a kaey role in providing cost-effective ubiquotious connectivity [21]. Among all the challenges to be faced, we can highlight the difficulty of providing good coverage to rural regions, particularly in terms of reliability, availability and promptness. For instance, outages experienced in the ground connectivity infrastructure during a natural disaster can prevent a prompt intervention, in turn leading to the endangering of the impacted population and exacerbated economic impact. Increasing the density of cellular sites would only be an ineffective and expensive solution to the problem. In fact, for many researchers, the best way forward is investigating on [22]. The biggest difference with respect to previous wireless generation networks is that NTN aim to work in a full three-dimensional (3D) space, instead of a quasi bi-dimensional one. This is made possible by the use of non-terrestrial nodes, such as UAVs, HAPs and satellites [23]. This is about wireless communication systems that operate beyond Earth's surface, and so they do not rely solely on terrestrial infrastructure for connectivity, making them crucial for extending global communication coverage, especially in remote or undeserved areas.

As illustrated in Figure 2.1, the architecture of a generic non-terrestrial net-

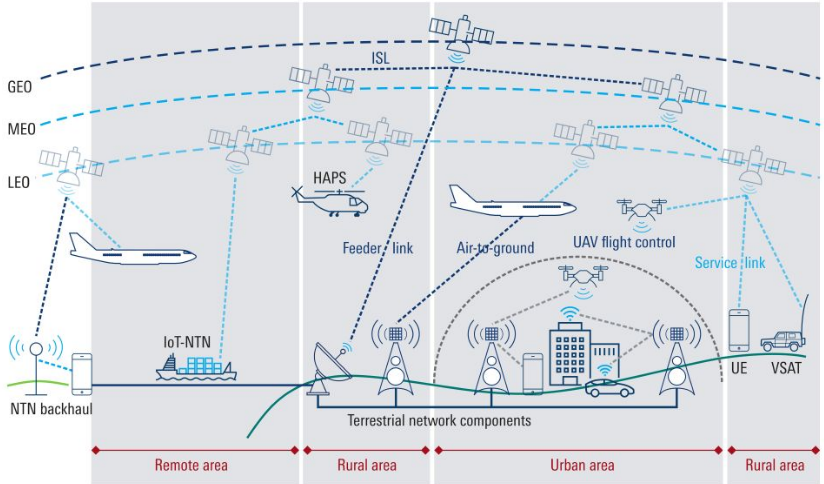


Figure 2.1: A general overview of NTN architecture. Figure from [24].

work may appear very complex. Let's take a look in more detail at each component of the system.

- **Terrestrial components:** there are several parts that constitute this part of the system. Two key points, are represented by the *NTN Backhaul* and *VSAT Systems*: they are the infrastructure that connects the non-terrestrial nodes to the core network on the ground through a feeder link [25]. This includes also terrestrial base stations, routers, and switches that facilitate the transfer of data between the non-terrestrial elements and the broader internet or local networks. In addition, there are also *User Equipment (UE)* and *IoT sensors* (Internet of Things) representing the end-users of the system. They can in fact send or receive data more effectively and reliably, thanks to the 3D network to which they are connected.

- **UAVs**: they are aircrafts that operate at low altitude, approximately below 300 meters, and without any human pilot, crew, or passengers on board. UAVs can be categorized by their payload capacity, which determines the weight of cargo they can carry and by their wing types. These nodes are very useful in providing broadband wireless connectivity, in particular during disasters or temporary events, thanks to their great movement flexibility. UAVs promote energy efficiency data collection, in particular for IoT devices, and they also can be deployed on-demand. On the other hand, these aircrafts are limited by energy constraints due to high flight propulsion power and their coverage capabilities are not very extensive since the low altitude [26].

- **HAPs**: they operate in the stratosphere, i.e., at an altitude of approximately 20 km. HAPs are characterized by a large geographical coverage due to their high altitude respect to UAVs, and by a low energy consumption, thanks to the installation of on board solar panels, which guarantees a good energy autonomy. Some drawbacks are the difficulty to perform refueling operations when needed and several challenges in terms of flying stabilization [27].

- **Satellites**: they can be classified based on their orbit characteristics. *Geostationary Earth Orbit (GEO)* satellites are collocated on the Earth's equatorial plane at an altitude around 30000-40000 km [28]. Due to their altitude, they have a rotation period of 24 hours, thereby, they are always visible from the ground and they do not need tracking. Moreover, to guarantee a complete coverage only 3 to 6 satellites are necessary. Due to their high altitude, their main disadvantage is the propagation delay that the signal takes to travel from ground to satellite and vice versa. Other alternatives are *LEO* and *Medium Earth Orbit (MEO)* satellites. They are collocated respectively at 200-500 km and 10000-20000 km, thus exhibiting a rotation period of 1.5 and 5 to 6 hours. Unlike GEO satellites, they need a slow speed tracking system since their positions change over time with respect to the Earth surface. Constellations of 30 to 60 satellites for LEO and 10 to 20 for MEO are necessary to guarantee a global coverage. On the other hand, they are much more convenient than GEO in terms of propagation delay. Typical Round Trip Time (RTT) values, i.e. the time required for the signal to travel from the Earth to the satellite and to came back again to the ground, are of 20-25 ms for LEO, 110-130 ms for MEO and 250-280 ms for GEO [29].

All these components do not necessarily have to be used as stand-alone deployments. Instead they can be combined in larger and more complex satellite networks, with the goal of promoting ubiquitous and ultrahigh-capacity global connectivity. These 3D satellite layers systems are called *Multilayered Hierarchical NTN* and they represent an important tool for the imminent arrival of 6G era [20]. Their name derives from the usage of multiple NTN platforms, with a layered subdivision due to their different altitudes. This orchestration among different aerial/space platforms represent one of the most promising proposals to solve coverage and latency constraints associated with individual NTN paradigms

[30]. For instance, the authros of [31] introduce these multi-layered NTN architectures, describing the advantages and downsides of various space nodes combinations. In particular, the authors have comapred, in terms of capacity and outage probability, a stand-alone GEO satellite and different cooperative architectures, such as GEO-to-Earth, GEO-to-LEO-to-Earth, GEO-to-HAP-to-Earth and GEO-to-LEO-to-HAP-to-Earth (Figure 2.2). They discovered that HAP relays are superior in connecting satellite signals to the ground, boasting a capacity that is up to six times greater than what is achievable through direct point-to-point GEO transmissions.

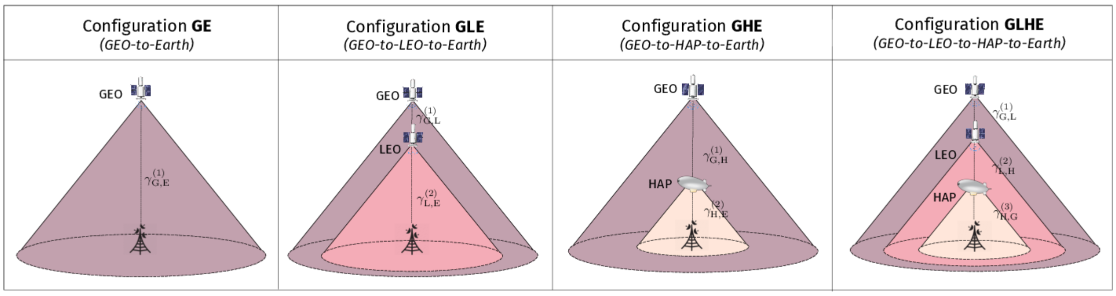


Figure 2.2: Possible multi-layered NTN configurations. Figure from [31].

2.2 Use Cases and Applications

For many years, the role of non-terrestrial devices was muchlimited to specific applications, such as navigation, television broad-casting, meteorology, and so on. However, the research community is increasingly focusing on NTN, in order to promote new and more complex services,whose were not possible before. In general, these research efforts target two goals: *NTN-IoT* and *NTN-NR*, each addressing different use cases and markets [32]-[33]. The NTN market has started establishing itself with the introduction of NTN-IoT, which broadens the scope of IoT by providing comprehensive global coverage across land, sea, and air, with a marked focus on GEO-based services. As NTN technology advances, NTN-NR will become more significant. Indeed, it facilitates direct connections between smartphones and other 5G devices, supporting low data services, voice, and messaging for various applications, mainly operating at the LEO altitude [34]. Over the next years, the NTN-IoT based services will have a growth in particular for SOS and two-way messaging, farming, asset tracking, disaster recovery, remote monitoring and autonomous driving. On the other hand, also

NTN-NR will have a rapid development in the next years. The scope of this thesis is more related to the latter type. Recent technological advancements in the aerial and space industry have paved the way for integrating terrestrial and non-terrestrial technologies, enabling more advanced use cases. As illustrated in [21], the main ones are:

- **Communication resilience and Service Continuity:** NTN enable the transition from a two-dimensional (2D) to a 3D connectivity layer, thus exploiting Earth's two-dimensional space. The deployment of NTN is particularly effective in rural and remote regions, when terrestrial towers are out of service or after a natural disaster, or even when the main path is unavailable, providing a secondary route to maintain the connection. At the same time, their deployment can be useful in assistance of already existing BS, providing high-capacity wireless coverage and preventing the overloading of terrestrial BS. Furthermore, these non-terrestrial platforms can guarantee Mobile Edge Computing (MEC) services, offering on-the-ground terminals additional computing and storage capabilities [35]. Moreover, they can promote fairness to all the users of the network, providing on-demand extra capacity to the most resource-constrained network entities: the cell-edge users.

- **Global Satellite Overlay:** achieving ubiquitous global coverage, using only wired devices, like optical fibers, represents a difficult objective for economic reasons and also for the impossibility of deployment in adverse regions [36]. For these reasons, whenever deploying ground base stations proves infeasible, a constellation of satellites, or other non-terrestrial platforms, can provide high-capacity access connectivity to on-the-ground devices, by forwarding the data signal trough an overlay space mesh network. This latter could be represented by the a multilayered hierarchical network or by a constellation of various satellite at the same altitude or, of course, a combination of them.

- **Ubiquitous IoT Broadcasting:** NTN platforms can also provide the service of moving aggregators for IoT traffic, or, in other words, offer global continuity of service for applications that rely on sensors [37]. This makes possible to convey multimedia and entertainment contests to a huge number of UEs, including also in-motion terminals that cannot benefit from terrestrial coverage. This is guaranteed by the wide geographical coverage and the broadcast nature of NTN platforms.

- **Advanced Backhauling:** Non-terrestrial terminals can wirelessly handle on-the-ground backhaul requests, especially in areas lacking wired backhaul solutions. This approach preserves terrestrial resources for access traffic and prevents the expense of traditional fiber-like deployments. Additionally, satellites and other aerial platforms can supplement terrestrial backhaul in densely populated regions with high peak traffic demands, thereby achieving load balancing.

- **Energy-Efficient Hybrid Multiplay:** Air-borne and space-borne platforms can offer high-speed connectivity while enhancing energy efficiency. Despite their high energy consumption for hovering, aerial platforms, such as UAVs can be deployed as needed with smart duty cycle control mechanisms, reducing the management costs associated with always-on fixed terrestrial infrastructures. Conversely, space platforms like satellites can be powered by solar panels, providing efficient, clean, and renewable energy compared to the conventional energy sources used by terrestrial devices [38].

Overall, there are several use cases and applications associated with NTN, differentiated by their purpose and methods. In summary, NTN offer a range of advantages, including reduced network costs, expanded coverage, and new revenue opportunities across various sectors. They are poised to play a crucial role in future telecommunication networks, supporting the evolution towards 6G and enhancing global connectivity. All this represents a natural starting point, which we can already experience in our present with numerous services already made available by various companies, but we will only truly discover the potential of NTN with the research that will take place in the coming years.

2.3 Challenges and Limitations

Along with the research community, the 3GPP, an international collaboration among seven telecommunications standard development organizations aimed at creating and maintaining the technical specifications for mobile telecommunications [39], is focusing on describing what are the main issues related to NTNs and where to intervene in order to better integrate them into the NR protocol stack. In this section we will delve into the potential key impact areas on NR to support NTN, and the Radio Layer 1 and Radio protocols issues and related possible solutions.

2.3.1 Potentially key impact areas on NR for supporting NTN

This subsection refers to TR 38.811 [40], and aims to describe the main NTN design issues that need to be addressed to correctly deploy NTN in real cellular scenarios. In particular, 3GPP identifies the following key impact areas:

- **Propagation channel:** the main differences with respect to standard NR is related to multi path and Doppler spectrum model, since they generally depend on the time evolution of the channel due to the transmitter (Tx) and the receiver (Rx) movements, or scatterer ones [41]. On the other hand, we usually consider outdoor and line-of-sight conditions for communications via satellite, instead indoor and non-line-of-sight conditions for HAPs system. Despite propagation problems, a great focus is required in order to maximize the throughput for Uplink (UL) and Downlink (DL) communications, and to maximize the availability of the service under deep fading situations [42].

- **Frequency and channel bandwidth:** the allocated spectrum for UL and DL satellite systems is 30 MHz at S band and 5000 MHz at Ka band, mostly using circular polarizations. Moreover, for efficient spectrum usage, it is recommended to minimise the risk of inter cell interference.

- **Cell pattern generation:** when the positions of UEs are unknown by the network, the contention based access channel can be put under stress. This is due to the fact that aerial and space systems typically feature larger cells compared to standard cellular networks. Moreover, if we consider non-GEOs, their positions may move over the time. All this could cause the increase of differential propagation delay between a UE at cell centre and UE at cell edge, especially at low operational elevation angles.

- **Propagation Delay:** the propagation delay is one of the most difficult challenges to deal with, and its value varies widely depending on which NTN system we are considering. This larger delay will in particular impact the data transfer levels, with access and transport procedures.

- **Mobility of the transmission equipment:** in cellular networks, when we talk about transmission equipment we refer to BS or Remote Radio

Heads (RRH). Except for GEO satellites, the transmission equipment in NTN are in motion, and this leads to Doppler effects. They depend on the relative BS velocity with respect to the UE, and on the frequency band. In turn, this leads to modification of the carrier frequency, of the phase and of the spacing, and may create Inter Carrier Interference (ICI) [43].

- **Service continuity:** this point focuses in particular on the service continuity between land-based 5G access and non-terrestrial based access networks, guaranteed by a hand-over procedure to or from satellite/HAPs system [44]. All this should take into account the support of both non-transparent air/spaceborne and bent-pipe architectures, handover preparation and failure handling, the time synchronization, lossless handover support and specifics related to intra-NTN mobility, as well as between non-terrestrial and cellular networks.

- **Radio resource management:** in satellite systems, access control is located at satellite BS, and gateway or hub level which may prevent optimal response time for access control. In order to support different traffic demands while also taking UE mobility requirements into account, requires minimal response time of the access control functionality, adapting it to the considered network topology. Moreover, pre-grants, SPS and/or grant free access scheme could represent possible turning points for this scope [45].

These differences between the above techniques highlight how each of the NR features may require adaptations to support NR operation via NTN. The motion of space/aerial vehicles, especially when considering non-GEO based access network, may cause the movement of cell patterns. The delay variation and all the issues related to the Doppler effect, causing problems to the hand-over/paging procedure, TA adjustments [46], initialization synchro-downlink procedure and Demodulation Reference Signal (DMRS) time density [47]. The huge altitude instead entails a high latency, which in turn affecting the HARQ [48], the Media Access Control (MAC), in particular the Radio Link Control (RLC) procedures, and the Physical layer procedures, specifically the Adaptive Modulation and Coding and the power control. On the other hand, the huge cell size can cause high differential delay, due to the fact that two different UEs may be far away within the same cell, damaging the TA in random access response message and the Random Access Channel (RACH) procedures. Moreover, the propagation channel of HAPs and satellites is mostly respectively direct Line-of-Sight (LOS), with slow fading and characterized by significant multipath effects. Thus necessitating adaptations to DMRS and to cyclic prefix, to compensate for the delay spread and the jitter/phase [49]. Another important aspect is the duplex mode, in which the usage of Time Division Duplexing (TDD) mode becomes difficult, as the necessary guard time, needed to prevent UEs to simultaneously transmit and receive, would lead to a very inefficient radio interface, due to the fact that it should commensurate the huge RTT delay [50]. This results in prefer the use of Frequency Division Duplexing (FDD) for most NR based NTN access network, with the possibility to consider TDD for HAPs and LEO satellites where the RTT still has low values [51]. From a practical point of view, the applicability of the duplexing mode TDD or FDD depends on the regulations (ITU-R and/or national) associated to the targeted spectrum. Satellite and aerial payload performance have to be considered too, since they cause phase noise impairment and back-off effects, bringing issues to Phase Tracking Reference Signal (PTRS) and to the Peak-to-Average Power Ratio (PAPR) [52]-[53]. Finally, the general architecture of the system will change, leading to modification of the Radio Access Network mapping and the need to develop new protocols specifically for these new scenarios [40].

2.3.2 Radio protocol issues and related possible solutions

In this subsection, the most important radio protocol issues and the related proposed solutions will be presented and discussed, starting from the description in the TR 38.821 [54].

- **Preamble detection:** in the context of NTN, differential delay could be experienced by two UEs within the same cell. This can lead to receiving the preambles associated with the same RACH occasion (RO) at different time. To avoid this situation, the preamble receiving window should start from $RO + RTT_{min}$ and end at $RO + RTT_{max}$, where RTT_{min} is the minimum RTT and RTT_{max} maximum one. Another important factor to take into account is the RO periodicity, since if it is not long enough then the preamble receiving windows, for two consecutive ROs, may be overlapped with each other, preventing the network to know the correct association between ROs and preambles, and ruining the estimation of the specific timing advance. The two proposed solutions are set the interval between two consecutive ROs larger than two times the maximum delay difference within the cell, and divide the preambles into groups and map them to different ROs, such that assign different groups of preambles to ROs with timing separation less than the time limit [55].

- **Random access response window:** in standard terrestrial communications, the Random Access Response (RAR), the response from gNB to the Random Access Preamble of an UE, is expected to be received within few milliseconds, otherwise the UE will send a new preamble. To account for on-terrestrial delays, an offset could be introduce for the start of the waiting window of the UE, taking into account the different possible non-terrestrial platform characteristics. Moreover, this process have to be done considering the specific RTT value, in particular for UEs that know their location, for which this value could be easily computed, deriving the associated offset too. Similar considerations can be done for the Radio Resource Control (RRC) connection request and the correspondent response from the gNB, for which introducing a dedicated time offset is mandatory to guarantee the correct working of the process. Moreover, an offset should be introduced also for the contention resolution timer between RRC Connection Request and the correspondent response from gNB, because, even if its maximum configurable value is larger enough to cover the RTT in NTN, it would be useful to avoid UE waste of power.

- **Timing Advance:** TA is a technique to adjust the uplink frame timing with respect to the downlink one, and it is specific for each UE, since it is used to advance or delay the transmission instant so as to compensate for propagation delay, achieving a perfect alignment within the receiver window of the gNodeB (gNB). It is derived from the UL received timing and sent back to the correspondent UE by the gNB. It can be computed during the random access procedure, measuring the received random access preamble and notifying the UE via the Timing Advance Command field in MAC RAR, or measuring the UL transmission and sending back the correspondent value to the UE. The possible solutions developed so far are different for the case of regenerative payload and bent-pipe payload, and they should take advantage of the knowledge of users positions. For UE without location information, the easiest things to do are broadcasting a common TA or extending the value range of the existing offset, and then the specific TA will be compensated via Timing Advance Command field in random access response. Instead, for UE with location information can be applied a specific TA correction during the 4-step random access procedure, or directly compensate applying the correct offset from UE side, when sending the random access preamble, broadcasting the delays in case of movements. A new challenge that can be found is therefore deal with a scenario in which there are both UEs with Global Navigation Satellite System (GNSS) and non-GNSS capabilities, and so the random access scheme, the resource allocation and control access for these might be different and difficult to apply equally. One possible solution is that the network configures separate resources and differentiate these based on GNSS capabilities and on each specific case being considering.

- **Discontinuous reception:** Discontinuous reception (DRX) is a mechanism that allows UE to save power by entering a low-power state for periods when it does not expect to receive data [56]. For this topic, adjustments are needed to the timers associated to UL and DL HARQ and retransmission processes. If HARQ is enabled, the solution could be to add a dedicated offset to these values, instead if it is disabled or active only for a certain number of processes, then the UE might be forced to monitor the Physical Downlink Control Channel (PDCCH) for retransmission opportunities that never will happen, and thus waste energy that reduces the battery lifetime [57]. Therefore, it is be essentially to confirm that the implementation does not start the DL and UL HARQ timer in this case. Furthermore, UE should not always monitor the PDCCH when the system is characterized by huge delay, since it will not receive something until at least one RTT, avoiding, in this way, waste of energy. It should switch from an active mode to a sleeping one cleverly, and this is mainly controlled by the network configurations. On the other hand, sometimes may occur some situations in which the UE enters in active mode without the network control, for example when it sends a Scheduling Request (SR) or when he has to reply to the RAR in Contention-Free Random Access. These are tricky situations since the UE would have to monitor the PDCCH for at least one RTT before receive any response. This could be fixed allowing the UE to discontinuously monitor the PDCCH during this amount of time, paying attention to the possible case in which the network schedules the UE directly after one of the previous cases. Finally, from UE point of view, after one RTT the network is allowed to reuse HARQ process IDs and it could send Downlink Control Information (DCI) allocations. However, in this period the UE is almost always in sleeping mode and this would represent a waste of time and resources. For this purpose, a possible solution could be to introduce an extra delay to any transmission in such a way to fall into the first available active period, or to allow the UE to leave its DRX state when there is the possibility to receive DCI by PDCCH.

We have summarised most of the issues that characterized non-terrestrial communications, proposing also possible implementations that could allow to fix them. We have also discussed which are the key impact areas of NR to support NTN, highlighting the main difference with the standard cellular systems.

3

Scheduling Protocols for NTN 5G NR

This chapter describes the role of scheduling processes in managing the allocation of resource for the data transmissions between UEs and the network in 5G NR, with a focus on its limiations and inefficiences when used in a non-terrestrial context. non-terrestrial extension. These processes are of fundamental importance for optimizing the utilization of available spectrum, ensuring efficient data transmission and avoiding waste of resources.

For what regards scheduling mechanism in standard 5G NR, it is quite similar to those in Long Term Evolution (LTE) [58], but with some enhancements, particularly in terms of granularity. In both cases, the network dictates the scheduling decisions to the UEs which must comply with the instructions. This centralized approach allows optimized resource management based on real-time network conditions and user demands. Depending on the scenario, on the predictability of the traffic patterns and on the need of flexibility in resource allocation, we can distinguish four main types of scheduling approaches.

- **Dynamic Scheduling:** with this approach, each transmission opportunity is individually scheduled by sending a DCI message from the gNB to UE. Therefore, each Physical Downlink Shared Channel (PDSCH) and Physical Uplink Shared Channel (PUSCH) message is scheduled based on current network conditions and UE requirements. It is very well suited to cases where the traffic varies over the time and frequent adjustments are necessary to optimize resource utilization [59].

- **Semi-Persistent Scheduling:** this technique allocates resource for a predefined period, typically for services with predictable traffic patterns and stable data rates, such as machine-to-machine communications or certain IoT devices. This approach significantly reduces the signaling overhead associated with constant adjustments, conserving network resources and improving overall system efficiency [60].

- **Grant-Based Scheduling:** with this approach, before trasmitting any data, the UE sends a SR, in line with the amout of packets in its trasmission buffer, to the BS, and this latter responds with a grant message allocating specific resources for data transmission. One of the primary drawbacks of GBS is the inherent latency introduced by the exchange of grant messages. This latency can be significant in scenarios requiring low-latency communication, such as ultra-reliable low-latency communications services [61].

- **Configured Scheduling:** this approach is less flexible than previous ones, since it assigns resources to UEs based on pre-configured parameters, either by the network or via higher-layer signaling. On the other hand, this approach makes it simpler to manage for predictable traffic flows. Furthermore, it fits well in scenario in which the traffic patterns is known ahead of time and does not change frequently, minimizing the need for multiple signaling between the network and UEs [62].

- **Predictive Scheduling:** all the above approaches are well suited for the specific cases for which they are been designed for, but, at the same time, it is also possible a combination of them in order to optimize resource usage based on predictions. Indeed, predictive scheduling aims to anticipate future traffic demands and allocate resource accordingly. It is particularly efficient for networks with mixed traffic types, allowing for more flexible management of resources based on prediction of future needs.

Apart from that, it is not only important to have an in-depth knowledge of scheduling mechanisms in order to be able to optimise them properly, but in the same way, it is essential to invest some time in a more general study of all the related subjects. For this reason, the role that the UDP and TCP transport protocols play in our non-terrestrial system will be discussed in the next section.

3.1 Transport Protocol Overview

Layer 4 of the ISO/OSI protocol stack, referred as Transport Layer, is where end-to-end logical communication between application processes running on different hosts are provided. The transport layer effectively provides to the upper layers, in a trasparent manner, an end-to-end connection between any two endpoints, even if they are located in different parts of the globe. All this is possible since transport layer protocols are implemented in the end systems and not in the network routers used to communicate. At this level, from sender point of view, messages are fragmented into segments and passed to lower layer, instead from receiver spot, the reverse process occurs, where segments are reassembled into messages and passed to higher level. For this reason, Transport Layer represents a junction between Network and App Layers. The main services that it offers are: process-to-process communication, addressing, encapsulation and decapsulation, multiplexing and demultiplexing, and also error, flow and congestion control, if it is implemented a reliable protocol, such as TCP.
Starting from that, it is possible to identity two different protocols within Transport Layer: UDP, that is unreliable connectionless, dedicated to simple and efficient applications, and TCP, characterized by a reliable connection-oriented approach and used in scenarios where reliability is one of the key points.

3.1.1 UDP - User Datagram Protocol

UDP is a connectionless and unreliable transport protocol that provides process-to-process communication using socket addresses [63]. In particular, each user datagram is independent, not numbered and can travel on different path. Moreover, UDP does not implemented any flow control or window mechanism, leading to possible receiver overflow due to receiving too many messages. Knowing all these characteristics, it may seem strange that UDP could be really useful as a transport protocol. However, just think of the case where a process wants to send a small message and does not care much about reliability: using UDP takes much less interactions between the sender and receiver than using more sophisticated protocols. In addition, UDP does not introduce latency delays to establish the connection, and does not have to wait for acknowledgment (ACK)s, so it is also used for many other applications.Furthermore, UDP can perform checksum calculation including a pseudoheader, the UDP header and the data coming from the application layer. This can guarantee a useful additional level of protection, with the price of violating the layering principles, and for that reason, it can be activated or deactivated as desired. Apart from that, there is no any error control mechanism in UDP, meaning that the sender has no way to know if a message has been correctly received or lost. In addiction, it does not use any congestion control mechanism, starting from the hypothesis that UDP should be used to transmit small and sporadic packets, therefore with low probability of congestion from the receiver side. Finally, it adopts encapsulation mechanism, as expected from a transport layer protocol, applying also multiplexing and demultiplexing procedures, since for each host running a TCP/IP framework, there is only one UDP but possibly several processes that may want to use the services of UDP.

3.1.2 TCP - Transmission Control Protocol

TCP is a connection-oriented and reliable protocol, which aims to provide guarantees on the packet reception, at the cost of additional complexity. To achieve this, TCP uses a combination of Go-Back-N and Selective Repeat, two sliding window protocols used for reliable data transmission and they differ mainly in how they handle the retransmission (RETX) of data frames when errors occur [64]. To do so, TCP adopts checksum for error detection, byte, sequence and ACK numbers, cumulative and selective acknowledgments, retransmission of lost or corrupted packets and reordering of arriving packets.
As UDP, also TCP implements services like process-to-process communication, multiplexing and demultiplexing. On the other hand, it offers a full-duplex service, where data can flow in both directions at the same time, in which each endpoint has its own sending and receiving buffer. As last thing, TCP explicitly defines three phases to perform a correct communication: a first one in which the two TCPs establish a connection between them, the next with the data exchange in both direction and finally the termination of the connection. In this way, it establish a logical, and not physical, connection, turning the unreliable and connectionless service of UDP into a reliable and connection-oriented one. Let's now focus on the three specific phases that it adopts.

- **Connection Establishment:** this phase is necessary to establish a single logical path between source and destination, in which all segments will be sent through it, facilitating also ACK and RETX processes. The connection establishment in TCP usually adopts the *three-way handshake* procedure: the first step is the synchronization (SYN), in which the client sends the first segment in which only the SYN flag is set and without any data. In the next step, the server replies with a SYN + ACK segment, in which it acknowledges the correct reception of the SYN and defines the receive window size for flow control. Finally, the client sends the third segment, containing just an ACK in the standard case, but with the possibility to piggyback some data in some situation, starting immediately the communication at this stage.

- **Data Transmission:** after the first step, the bidirectional data transfer can begin, starting from the possible piggyback procedure. What is interesting to analyse is the delay needed before starting to transmit data. With the standard connection establishment the delay amounts to at least three time the propagation delay (τ_p), since it includes SYN, SYN + ACK and final ACK processes, instead with piggyback it reduces to only $2\tau_p$. This is true as first approximation if we assume that the transmission time is much less than the propagation time, which is the exact case in NTN scenarios.
- **Connection Termination:** when there is no more data to transmit, the procedure for closing the connection is initiated. With the *three-way handshaking* version, the TCP client sends a FIN segment, the server replies with a FIN + ACK, and then the client replies itself with an ACK.

For what regard the flow control in TCP, it is managed by windows from client and server sides, necessary to accumulate the sending and received segments. From the client side we have the sending window: it is the amount of data that can be in transit between two endpoints at any given time. Instead from the server side, the receive window represents the amount of data in bytes that the receiver is willing to accept and buffer during a connection. In particular, it is the difference between the buffer size and the number of waiting bytes to be pulled up by the process. Moreover, the window size is communicated from the receiver to the sender through the TCP segment header. In this way, the sender can dynamically adjust the sending window according to the receiver's requirements. From this perspective, there is also a window for congestion control: it defines how many bytes can be transmitted before stopping to wait for ACKs. In particular, the sending window is defined as the minimum between the congestion and receive counterparts, since the first one is related to the congestion

in the network and the second to the congestion at the end point. The TCP's policy for handling congestion is based on 3 algorithms: the *slow start*, *congestion avoidance* and *fast recovery*. The first one allows to start slowly but increase the congestion window exponentially up to a certain point, in order to reach a good throughput fast. Then, when the size of congestion window reaches a slow-start threshold, the congestion avoidance phase begins. It is characterized by a linear increase to avoid congestion before it happens, since slow start can rapidly saturate the link. In conclusion, we have the fast recovery phase that starts when three duplicate ACKs arrive, which is interpreted as light congestion in the network. It increases the size of the congestion widow linearly and it is only an optional algorithm in TCP.
Of fundamental importance is also the TCP error control system, which is characterized by a retransmission time-out (RTO), that can dynamically adjust based on the RTT value, and by a fast retransmission, for which if three duplicate ACKs arrive for a segment, then the next segment is retransmitted without waiting for the time-out (TO).

3.2 Scheduling in 5G NR

Scheduling processes at the MAC layer in telecommunications involves managing how data packets are transmitted over a shared communication medium. This process is crucial for ensuring efficient use of resources,such as energy efficiency and bandwidth, and maintaining Quality of Service (QoS) across networks. However, there are several problem relating to scheduling processes in NTN communications. With respect to the terrestrial NR system, factors such as propagation delay and relative positions between entities in the network can be orders of magnitude higher, and exhibit higher variability as well. This can lead to several issues specific to the NTN scenario, and the research community is focused on finding solutions to address this aspects. This section describes which are the answers proposed by the 3GPP and to summer up the current state of the art.

3.2.1 Solutions of 3GPP for NR to support NTN

In this subsection, we will analyze the most important radio protocol issues and related solutions on NR to support NTN, which we have not yet dealt with, with reference to [54].

- **Scheduling Request:** there is a timer between when a UE sends a request for Uplink Shared Channel (ULSCH) resources from the gNB to perform a new transmission. RRC configures this process: the timer at latest expires after 128ms and then initiates a SR. In this situation the main problem is due to the propagation delay, in particular for GEO satellites, where the the value range is not sufficient because the RTT is larger and the timer value should be changed, according to the specific use case.

- **PHY/MAC HARQ:** it is the main procedure to support error correction and retransmission in NR systems and it ensures delivery between entities in the network. In the specific case of NTN, the critical and dynamically conditions, in particular related to propagation delay, make it difficult to perform HARQ correctly. For that reason, disable HARQ or increase the number of processes could represent one possible solutions. Enabling or disabling of HARQ feedback is a network decision signalled semi-statically to the UE by RRC messages.

- **Uplink Scheduling:** typically when a UE has data in the buffer to be transmitted, a Buffer Status Report (BSR) is triggered and, if the UE does not have any uplink resources for transmitting the BSR, it will send a SR to ask for them. On the other hand, the network does not know the size of resources required to schedule the UE, since the SR is only an needed indication. Thus, first the network may typically schedule the UE with a grant large enough to send a BSR so that the network will schedule the UE more accordingly in the following transmissions, as illustrated in Figure 3.1.

  The drawback of this procedure is that it takes at least two RTTs from data arriving in the UE buffer to correctly receive them from the gNB side. Due to large propagation delay, this process can become extremely hard to be carried out correctly. In order to mitigate this problem, there are many possible solutions, each with its pros and drawbacks. The already described standard one is the SR-BSR procedure, in which low resource overhead

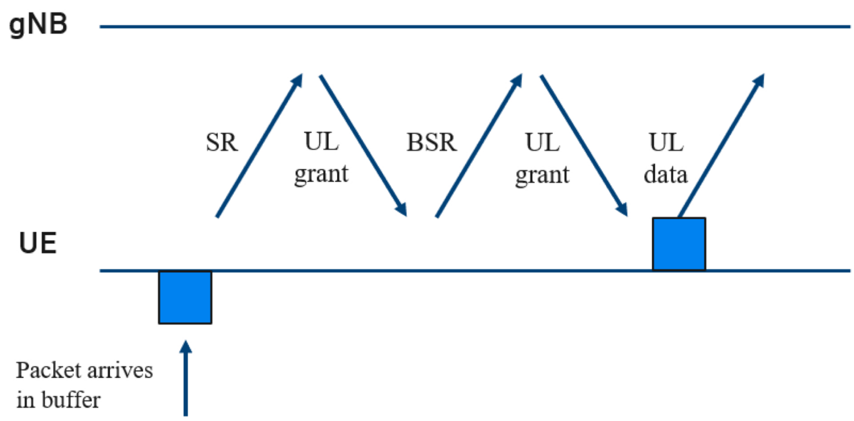


Figure 3.1: Scheduling for UE transmission. Figure from [54].

is required but it takes at least two RTTs of delay. Another possibility is to send large grant in response, with potentially low resource overhead, but it still takes two RTTs and it might be a waste of resources since the network is not aware of the UE buffer status. Also configured grant and BSR-indication can constitute an alternative, since they are characterized by a low latency, equal to one RTT with the right configuration, but it has larger overhead, so more research is needed to find the best trade-off. Finally, with BSR over 2-step random access can guarantee low latency and low overhead but it requires several RACH resources.

- **RLC HARQ:** the RLC status report is triggered by the polling procedure or by error detection in reception of an Protocol Data Unit (PDU) segment, indicated by the expiration of t-Reassembly timer. This procedure is used in both RLC Acknowledged Mode (AM) and RLC Unacknowledged Mode (UM). Its value can be configured between 0 and 200ms, since it has to cover the largest time interval in which an individual segment arrives out of order at the receiver side due to segmentation and/or HARQ processes before a status report and consequently an retransmission are triggered. For that reason, if HARQ is supported by NTN, the value of t-Reassembly should be modified to compensate the specific case scenario, according to RTT, HARQ retransmissions and scheduling offset values.

- **Packet Data Convergence Protocol:** the QoS requirements are mainly reflected by the discard timer of Packet Data Convergence Protocol (PDCP) transmissions. A PDCP Service Data Unit (SDU) is discarded when the correspondent timer expires or when a status report confirms the correct delivery. Its value can be configured between 10ms or 1500ms, based on the RTT value, number of retransmissions and on the presence of HARQ,with the possibility to switch it off, setting it to infinite. For that purpose, it is necessary to change the reordering delivery timer, PDCP sequence number and window size values, according to NTN scenario characteristics.

3.2.2 State of the Art

Several universities and research groups around the world are currently investing time and resource on NTN research, looking for new solutions able to efficiently and reliably enhance NTN performance, given the previously discuss problems in the above sections.

General Scheduling techniques

In [65], the authors investigate on the extend RTT and increased path loss in NTN scenarios, resulting in a degraded network throughput. The researchers showed that the network throughput could be improved by efficiently scheduling the HARQ processes, instead of requesting additional resources and increasing the transmit power. Of course, better performance can be achieved by the well know solutions of disable the use of HARQ feedback at MAC layer or increasing the number of HARQ processes. However, the winning idea was to design a smarter transport blocks scheduling methods, able to increase the efficiency of resource allocation and utilization. They developed flexible downlink data-to-acknowledgment delays and uplink grant-to-data delays for more efficient NTN signaling, ensuring that the proposed solutions works for different satellite orbits and even with varied number of HARQ processes. With this work, they demonstrated noticeable gains in throughput, adopting suitable coding rates for each specific scenario, enabling NTN system to serve an increased number of UE, which is one of the most critical points for these network, since their extended cell-size due to the height of satellites.

A multi-connectivity (MC) scheduler of uplink multi-layer NTN was proposed in [66], where UEs can be served by more than one satellite to achieve higher peak throughput. This technique has been introduced by 3GPP originally for terrestrial 4G and 5G system, but it has the possibility to show gains also for non-terrestrial ones. MC has the ability of enhance the offered system capacity and simultaneously improve the spectral efficiency in heterogeneous networks. Moreover, it achieves higher-per-user data rate, providing mobility robustness and improving the resilience of wireless communication,that are two of the most critical points in NTN. All of these advantages require the price of increasing power consumption at user terminal for UL transmission, that may be critical in some cases. For that reason, researchers focused on design an efficient terminal aware multi-connectivity scheduling algorithm, that takes into account the available radio resources and propagation information to intelligently define a dynamic resource allocation pattern. In particular, it optimally routes traffic so as to maximize UL data rate while minimizing energy consumption. They were able to achieve excellent results by realising a multi-layer NTN resource scheduling architecture, aggregating the capabilities of LEO, MEO and GEO satellites. Moreover, radio link parameters are taking into account, such as the energy per bit to noise power spectral density ratio (E_b/N_0) and the carrier-to-noise ratio (C/N), along with the UL information.

The impact of large differential delay and Doppler shifts within a NTN cell on the 5G-NR resource allocation and MAC protocol are investigated in [67]. The UL performance degradation due to Doppler shift can be compensated if each individual UE can estimate and consequently correct each shift appropriately before transmitting the signal. Moreover for UL transmissions the UE must have the time to prepare the data to send, and therefore the gNB takes the UL scheduling decision in advance and then sends the UL grant accordingly. This corrections are of course different for each users, but the offset is usually computed considering the worst case scenario. Therefore, the latency for low propagation delay users would be too severe and this would impact the general system performance. To resolve this, in [67] the focus is on the design of an UL zone-based scheduling technique to address the high differential delay and Doppler shifts for the specific case of LEO satellites. Specifically, different scheduling offsets can be considered for different zones within the same satellite beam, radically improving the latency of UEs without increasing the complexity at the UE-side, or required of any additional signalling overhead.

Semi-Persistent Scheduling

Another possible path to follow for optimizing the scheduling procedures is the Semi-Persistent approach. It was originally designed for constant bit-rate voice applications. However, the very low control overhead makes it a potential latency reduction technique in 4G and 5G systems. In [68], a comparison between SPS and dynamic scheduling was made in order to analyze the impact of resource allocation methods on UL latency. The research had the specific focus on narrowband 4G networks and the conducted simulations were performed through ns-3. The mainly difference between GBS, referred to as dynamic in this case, and the semi-persistent one is illustrated in Figure 3.2.

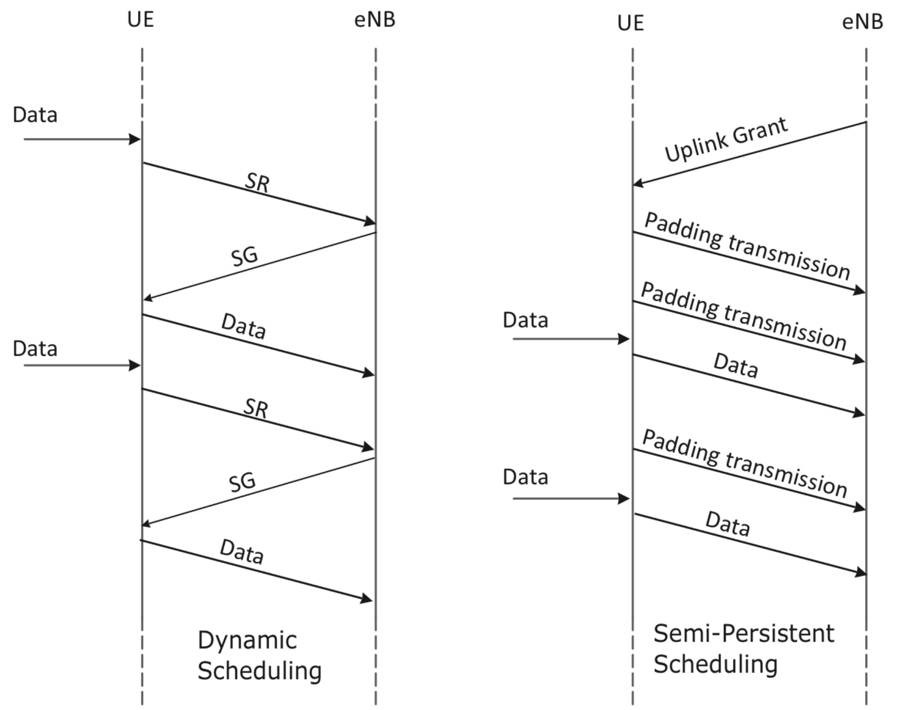


Figure 3.2: SPS and GBS schemes. Figure from [68].

In the first case, the UE requires resource allocation each time it has something new to transmit, and it does so sending a SR to the BS. This latter will schedule the user sending to it a grant, that will be used by the UE to send the data contained in transmitting buffer. On the other hand, SPS periodically allocates resource to the user, regardless if it has something to transmit or not. Consequently, it will perform padding or data transmission depending on its buffer status. Pay attention about the fact that SPS introduces spectral inefficiencies and increased energy consumption. The first one happens when the UE has nothing to send, while the latter is due to the undesired transmissions from the user even when there is no available data. Moreover, since the scheduling decisions are taken without any user request, it may happen that the UE has several data to transmit between two UL grants, and this will lead to undesired latency inefficiency. However, GBS introduces a transmission delay as well: the minimum time it takes from the arrival of a new packet in the buffer and a scheduling grant is correctly received by the UE is at least of one RTT. For these reasons, it is necessary to adapt the desire scheduling approach to each specific use case scenario. The results emerged in [68] show that SPS has the potential to improve system performance in terms of UL latency with a factor of more than two. Furthermore, it turned out that for a high number of simulations data uploads, there is only a small difference between the two types of scheduling, and also the system bandwidth limits the device density for keeping the latency under the threshold defined by different ultra reliable low latency communications.

On top of NTN, the SPS is identified by 3GPP as one of the most promising technique even for Vehicle-to-Everything (V2X) systems, and the third generation partnership project introduces it into the MAC layer of LTE-V2X Mode 4 in release 14. A mathematical model for SPS reliability in this field can be found in [69], in which a physical layer and MAC layer representation of sensing-based SPS are presented, and scheduling reliability and channel busy rate are given. The simulations results show that the theoretical curve can successfully represent the scheduler reliability, and that this latter has a quadratic function relationship with the channel busy rate, which provide theoretical support for possible future work.
Starting from the well known fact that SPS works fine for periodic transmissions comparatively to aperiodic messages, in [70], researchers proposed an enhanced semi-persistent scheduling method for resource reservation for aperiodic traffic. This will overcome the already reserved unutilized resource and the increase of resource contention, due to the fact the aperiodic messages compelled the vehicle to select new resources for its transmission based on the latency associated with the packets generation. The proposed algorithm works with the reinforcement learning mechanism where each vehicle acts as an agent. Looking on the traffic density and speed of vehicle, the protocol dynamically adjusts the sensing window size and, at the same time, re-evaluation mechanism is

introduced to confirm the available resources by performing the sensing again while selecting them. This approach decreases the resource contention and assists in conflict-free resource assignment for aperiodic transmission, resulting in improved network performance in terms of Packet Delivery Ratio (PDR).

4 End-to-end simulation of NTN

4.1 The Role of ns-3 in Networking Research

Ns-3 is a free and open-source discrete-event network simulator, primarily dedicated to research, development and educational purposes. It implements several specific simulation environments, in line with the evolving need of modern networking research, and it is supported by a strong and numerous communities that maintaining and updating it over the years. The choice of using ns-3 throughout this thesis is motivated by its key features that distinguish it from other available network simulators. Its modular design is composed as a set of libraries that can be combined with external software, offering a high degree of flexibility and customization. Ns-3 supports both C++ and Python, allowing researchers to implement the simulation at low level for greater precision and efficiency and, at the same time, manage the results nimbly with Python. Moreover, it provides models of packet transmissions over communication channels with bit-level realism, supporting various emulation modes of operation. Finally, it is characterized by a core support for running simulations across clusters of workstations, thus enhancing scalability. In conclusion, the importance of ns-3 lies in its ability to offer a highly controlled and reproducible environment for studying system behavior under various conditions. Its emphasis on modeling Internet protocols and network topologies (Figure 4.1) make it as a fundamental tool for understanding and innovating within the realm of Internet systems. Considering all these reasons, the choice of ns-3 as the main simulation tool comes naturally, especially for the specific scope of this thesis, where the simulation of communications in extreme environments and the need to obtain results that are accurate and truthful as much as possible form the basis of the work.

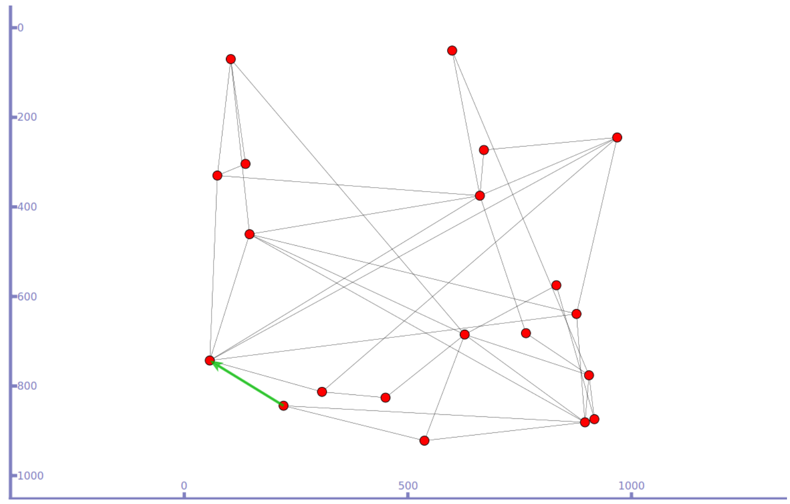


Figure 4.1: A network graph using ns-3. Figure from [71].

4.2 Ns3-ntn module

The collaboration of SIGNET research group and New York University (NYU) gave birth to the *ns3-mmwave* module [13]. It is a end-to-end full stack module, able to reproduce mmWave communications into the open-source ns-3 simulator. The module includes a variety of detailed statistical channel models, alongside the capacity to integrate real measurements or employ ray-tracing methodologies. Additionally, the module integrates with the core network of the ns–3 LTE module, facilitating comprehensive simulations of end-to-end connectivity across the stack. Furthermore, it incorporates advanced architectural functionalities, including dual-connectivity, among other features.

The goal of this thesis is to expand the above work into a new version specific non-terrestrial case: the *ns3-ntn* module. The implementation has already started from the work of other researchers. The 3GPP channel model for NTN has been implemented, introducing an ad-hoc characterization of the attenuation of the signal in the space scenario, including also new challenges associated with latency and coverage constraints [72]. The combination of the *ns3-mmwave* module and the NTN channel leads to an incorrect functioning of the cellular communication system. The reason is due to the use of the terrestrial protocol stack, which is obviously not adapted for space environments. In this regard, the current challenge is to make the *ns3-ntn* module efficient enough so that its performance can be compared with those of the terrestrial counterpart, although all the non-terrestrial system challenges. For example, in the baseline *ns3-mmwave* module, the propagation delay is basically negligible since it is associated to terrestrial links, and, for that reasons, it is not taken into account. On the other hand, this approach is not possible for NTN. For that reason, modifications and adaptations are necessary to to ensure proper functioning.

4.2.1 Ns3-mmWave Module

The starting idea was to have a simulation program to model the mmWave technology and beyond, since their central role for the current and future generation of cellular systems. The focus of this module is related not only to the well studied PHY layer, but it is also extended to all the above layers. In particular, the choice to use an end-to-end full stack network simulator, such as ns-3, represents the best way forward. In order to make it easy to integrate new future protocols and algorithms, the PHY and MAC layers are modular and highly customizable. Moreover, the full-stack simulation allows to model all layers of the protocols stack as well as applications running over the network. On the other hand, in order to obtain results as accurate as possible, it is of fundamental importance to model in detail the behaviour of all the different components that interact together in a cellular systems. For this specific case, the most important elements are: the channel model, the users mobility, the network deployment and the level of detail when modelling the protocol stack of communication links and of end devices.

As it is possible to notice in Fig. 4.2, the structure of the module is very complex, since it includes all the specific mechanisms in a general cellular system. From the device application to RRC, RLC and MAC layers, from HARQ and scheduling processes to PHY layer, including error model, interference, spectrum channel, and so on. All of these aspects are modelled individually and connected by means of a modular approach, with the possibility of expanding them with new algorithms or specialising them for different cases.

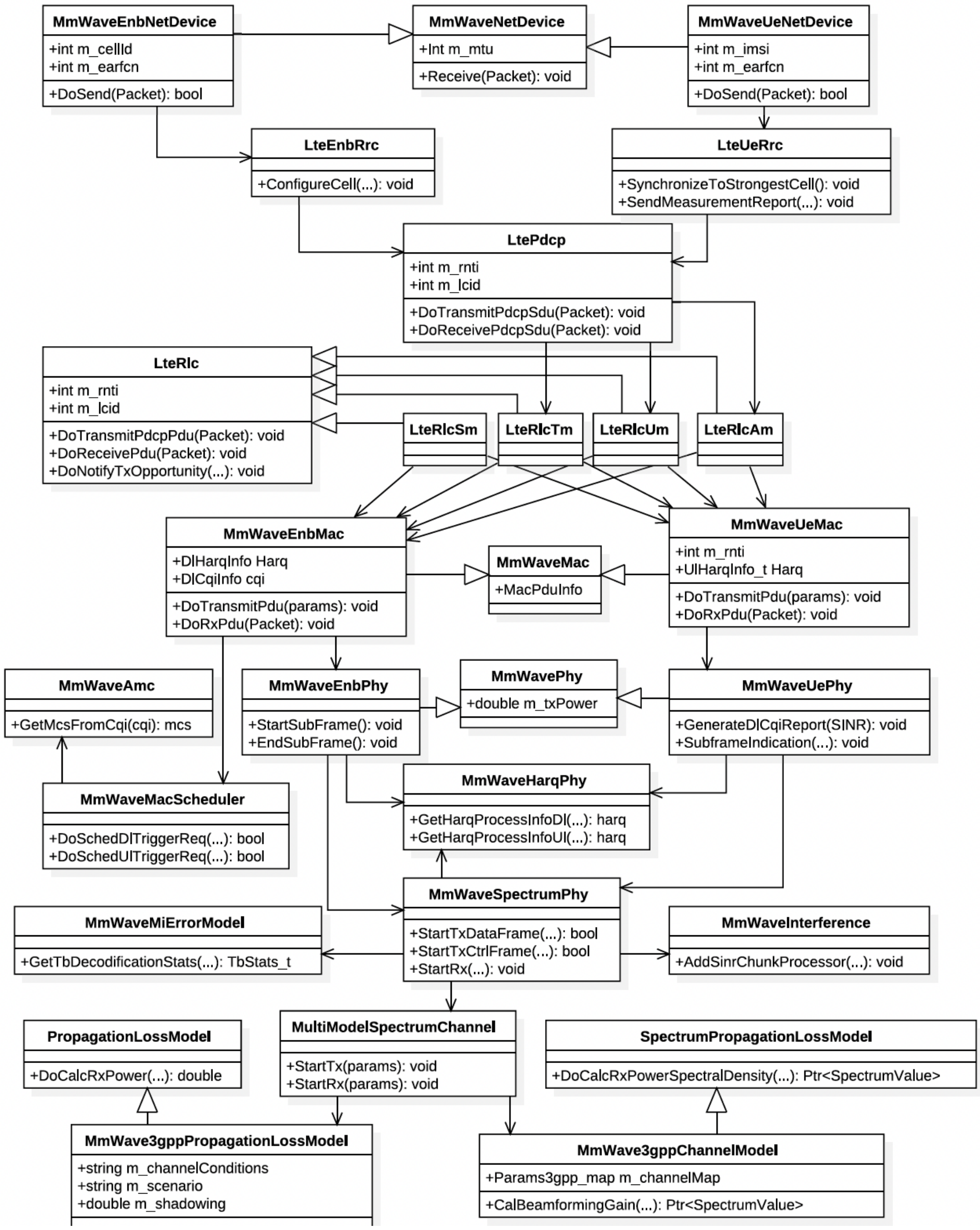


Figure 4.2: Class diagram of mmWave module. Figure from [13].

The classic scenario adopted to obtain numerical results for this purpose, and which will also be used for following chapters on the design and implementation of new scheduling algorithms, is represented in Figure 4.3.

In Figure 4.3, a end-to-end simulations of 3GPP-style cellular network sys-

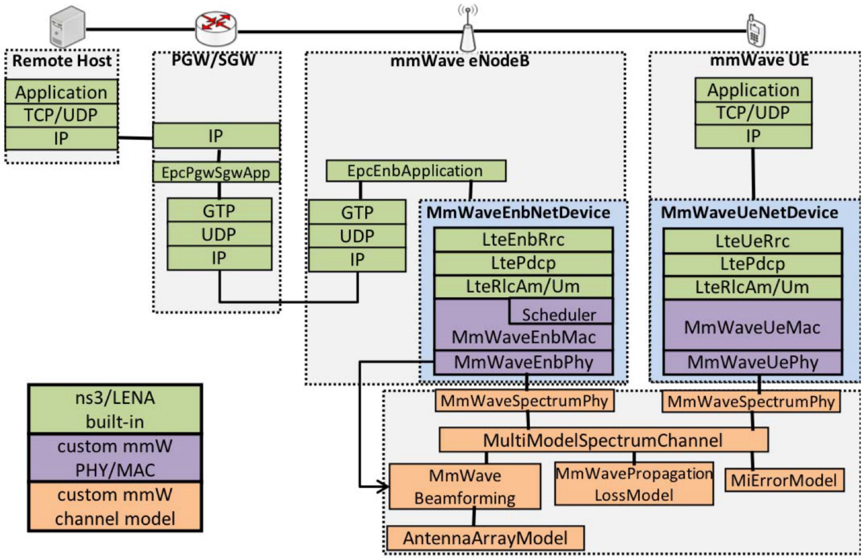


Figure 4.3: Class diagram of mmWave system. Figure from [13].

tem is represented. In particular, it shows a high-level composition of the `MmWaveEnbNetDevice` and `MmWaveUeNetDevice` classes, which represent the mmWave gNB and UE radio stacks, respectively. Furthermore, the `MmWaveEnbMac` and `MmWaveUeMac` MAC layer classes implement the LTE module Service Access Point (SAP) provider and user interfaces, which enable the inter-operation with the LTE RLC layer. The MAC scheduler also implements `LteEnbRrc`: a SAP for configuration at the LTE RRC layer. The `MmWavePhy` classes handle UL and DL data transmissions and control channels based on MAC layer control messages. The class `MmWaveSpectrumPhy`, which is shared between UL and DL, is used for PHY instance communications over the channel, i.e. `SpectrumChannel`, mainly using TDD techinques. Moreover, this class includes all PHY layer modules like Signal to Interference plus Noise Ratio (SINR) calculation by `MmWaveSinrChunkProcessor`, the mutual information-based error model by `MmWaveMiErrorMode`, which computes the packet error probability, as well as the HARQ PHY-layer entity to perform soft combining by `MmWaveHarqPhy`.

4.2.2 Scheduling process

As already said, the *ns3-ntn* module derives directly from *ns3-mmwave* baseline, and for that reasons it inherits some limitations of its precursor, such as the necessity of the system to operate in TDD and based on Time Division Multiple Access (TDMA). Additionally, the association between UE and gNB is static during the simulation, as UEs cannot perform handovers between different BSs. Nevertheless, it is possible to simulate satellite mobility based on pre-defined orbit.

In 5G NR, the scheduler located at the MAC layer dynamically allocates the UL and DL resources by assigning the so called resource blocks in the frequency domain and Orthogonal Frequency Division Multiplexing (OFDM) symbols in the time domain. These processes are computed at every slot and the scheduler conveys this information to UEs via the DCI. Taking into account the transmission delay in terrestrial scenarios, thus achieving perfect synchronisation, scheduling allocations are performed K_0 or K_2 slots in advance for DL or UL transmissions. Starting from the hypothesis that UEs and gNBs are perfectly synchronized in time, then there exists a complete match between: the slot reported in the DL DCIs, the slot during which the gNB performs the transmission and the slot in which the target UE will receive such data. Typically, for standard terrestrial systems the propagation delay does not exceed a few milliseconds, so in the worst scenario it can be taken into account using the TA mechanism, anticipating all operations by the time needed. If this corrections are not properly performed, possible mismatch may happen between DCIs information and actual instant of transmission and/or reception of data. On the other hand, this mechanism is clearly not enough for NTN, where the propagation delay could be several tens of milliseconds. To solve this problem, a new mechanism has been designed and integrated to the module. We start from the assumoption of knowing an estimate of the propagation delay t_{prop}, usually measured in slots, between UE and gNB, and from it we are able to perform the right correction even for NTN case.

Figure 4.4 illustrated how transmission scheduling works in the simulator. The two vertical rows basically represent respectively what the gNB and UE do as time passes. Moreover, each arrow represents a transmission, whether it is data or control message.

Specifically, DL and UL cases are managed as follows.

- **DL:** in Figure 4.4-(a), the gNB schedules DL transmissions K_0 slots in advance, and then propagates this decision to the target UE via the dedicated control channel, represented by the $G1$ segment. After a delay of t_{prop}, the UE receives the corresponding DCI, represented as U1, and so schedules a reception after K_0 slots. Please pay attention that, with respect to the absolute starting time, this corresponds to the slot indicated in the DCI shifted by t_{prop}. Then, the gNB will transmit the corresponding Transport Block (TB), as $G2$, which will be received by the UE after another t_{prop}, as $U2$.

- **UL:** in Figure 4.4-(b), the gNB schedules UL transmissions K_2 slots in advance, and then propagates this decision to the target UE via the dedicated control channel, as $G1$ segment. At the same time, starting from the knowledge of the propagation delay, the gNB schedules the reception of the corresponding TB after $2t_{prop} + K_2$ slots in the future. Then, the UE will receive the corresponding DCI, represented as $U1$ with a delay equal to t_{prop}, and then schedules a transmission after K_2 slots. Please notice that also in this case, this corresponds to the slot indicated in the DCI, shifted by another t_{prop}. Finally, the UE will transmit the UL TB after $t_{prop} + K_2$ slots respect to when the gNB sent the corresponding DCI, and this is represented as $U2$ segment. At the end, the gNB will receive the TB $2t_{prop} + K_2$ slots after scheduling it, as $G2$.

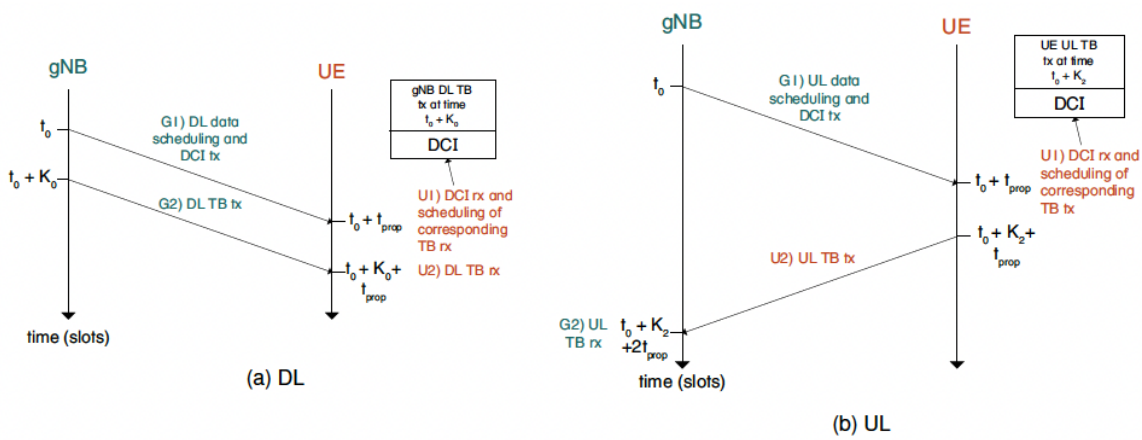


Figure 4.4: UL and DL scheduling in *ns3-ntn* module. Figure from [73].

5

Preliminary performance evaluation of 5G NR in NTN scenarios

The objective of this thesis is to bring a contribution in the proper functioning of the 5G NR protocol stack in the non-terrestrial environment, simulating these kind of systems via the *ns3-ntn* module. As described in Chapter 4, this latters is composed by the baseline *ns3-mmwave* module with the addition of a dedicted 3GPP channel model for NTN scenarios. However, this combination leads to an incorrect functioning of the cellular communications, due to the use of the terrestrial protocol stack in space environments, which it is not adapted for. The focus of this thesis is to bring improvements on the scheduling procedures at MAC layer. On the other hand, before this, a more detailed study on the current performance of the protocol stack is necessary. The work will start with an initial review on how the native GBS works using UDP and TCP as transport protocols. This will allow us to understand if some extra preliminary adjustments are necessary before implementing enhancements and new possbile solutions for the scheduler mechanism. A first step will be to understand how performance changes as propagation time or distance increases, with a cosequent impact on system timers and Signal to Noise Ratio (SNR). In particular, in this chapter the focus is on understanding whether the native GBS approach is efficient, even in scenarios with large distances between network nodes. Instead, in Chapter 6 will be study where is useful to take action, proposing and implementing new scheduling protocols. Among all, SPS will be developed, specifically for applications that generates periodic traffic, and then a new type of scheduler, called *Smart Scheduler*, able to understand the traffic type of the used application, whether it is periodic or aperiodic, and choose the best approach according to the needs.

5.1 Scenario and Parameters

The main scripts used are: `three-gpp-scen-10-delay-test.cc`, `test-ideal-delay.cc` and `test-leo-interval.cc`. They are test scripts to evaluate system performance. In particular, the last two have been developed specifically for the work of this thesis. In all of them, the scenario is basically the same: a wireless link between one terrestrial user and a LEO satellite, which itself features a backhaul link with a Remote Host (RH), as illustrated in Figure 4.3. Both UE and RH generate packets and so both UL and DL transmissions are possible.

Satellite antennas are modeled as a circular aperture antenna, i.e., a reflector antenna that offers circular polarization. Instead, the UE adopts a VSAT antenna radiation pattern model. Specifically, these parameters are taken from scenario 10 DL in TR 38.321 within 3GPP standards [74].

Satellite	**Value**	**User**	**Value**
EIRP Density	40 dBW/MHz	Noise Figure	1.2 dB
Antenna Gain	58.5 dB	Antenna Gain	39.7 dB
Antenna Diameter	5 m	Antenna Diameter	0.6 m

Table 5.1: Satellite and user antenna parameters.

Furthermore, frequency and bandwidth are set respectively to $20GHz$ e $400MHz$. The user is located in Padua, instead the satellite has the same (x,y) coordinates, but with a different altitude, increased by a variable `distance`, that can be fixed or varied according to the propagation delay as required. Both the node positions are fixed and do not change over the time. UE and BS establish an IPv4 connection, with the option of using UDP or TCP as transport protocol. The Path Loss Model and Propagation Loss Model are configured to work in a *NTN-rural* scenario, as described in the TR 38.811 [75].

Below some simulation parameters are presented. Those set to "configurable" change their values during the different simulations, while those with a fixed value were kept constant for all configurations. Unless some specific cases where the change is made explicit.

Simulation	**Value**	**Application**	**Value**
Propagation delay	configurable [ms]	Send interval	configurable [ns]
Distance	2000 m	Packet size	200 byte
SIM start time	0 sec	APP start time	0.5 sec
SIM stop time	16 sec	APP stop time	15.5 sec
RRC ideal	true	Nagle algorithm	disable
Error rate	0	HARQ	disable

Table 5.2: Simulations parameters.

Please note that error rate and Nagle algorithm are basically deactivated in order to well distinguish each packet path, also minimizing the presence of errors. Additionally, `sendInterval` variable has the same purpose, but it will be treated in a next section. The `distance` variable could be fixed to obtained a constant SNR or computed as a function of `propDelay`, while increasing the error probability. The application parameters can be configured to obtain the desired application behaviour. For this work, simulations were carried out for a long enough time and with no pause in the application. Moreover, the application starts after $500ms$ the beginnin of the simulations. This allows us to avoid the overlap between initials control message and data ones. In conclusion, as we will discuss later, we preferred to make the initial control RRC message ideal, by setting `rrcIdeal` to false.

Regarding the HARQ, it is usually disabled since its 5G NR version is not yet optimized for NTN cases and this also allows us to better isolate the scheduler performance. In addition, the simulation parameters were configured so that the value of SNR is basically always very high and the error rate is set to zero. This is made possible by configuring the constant `distance` value at $2000km$, ensuring an SNR value around $30dB$, and setting the `errorRate` variable to zero. Please remember that our goal is to understand what role propagation delay plays in these kind of systems.

Where TCP is used as transport protocol, our choice falls on the *Cubic* version with the following parameters [76]:

TCP Cubic	**Value**
Mininum reTx timeout	200 ms
Fragment expiration timeout	200 ms
Send and receive buffer size	$13 \cdot 10^6$ byte

Table 5.3: TCP Cubic parameters.

Furthermore, the TCP socket segment size is set equal to `packetSizeBytes` to keep packets intact and the `DelAckCount`, i.e. the number of packets to wait before sending an ACK, is set to one.
At RLC level, the choice fell on the RLC UM version [77]. Default buffer sizes have been increased to avoid data loss at upper layers, and the report buffer status and reordering timers have been increased respectively to three and two times the maximum propagation delay, to account for its non-negligible value.
The same treatment is also necessary for RRC timers [78]: in particular the values of `ConnectionRequestTimeoutDuration`, `ConnectionSetupTimeoutDuration`, `ConnectionRejectedTimeoutDuration` and `LteUeRrc::T300` have been increased to account for the non-negligible propagation delay. These parameters refer to specific timers and constants that manage the timing and behavior of connections and communication processes between the UE and the network at RRC layer. Specifically, they are responsable for establishing, maintaining, and releasing radio bearers, as well as for controlling the mobility of users.

5.1.1 Graphical result metrics

In this subsection, the metrics to produce the main result plots are explained. In particular, they are the outcome of several different simulations, characterized by different `RngRun` values, i.e., a mechanism for controlling the randomness and also ensuring reproducibility and statistical rigor in the simulations. Usually both average and median results are provided.

- **PHY-layer throughput:** it is computed as the ratio of the sum of all DL and UL packets correctly received during the simulations at PHY level, and the total time from the first transmitted and last received packets, looking from the produced file `RxPacketTrace.txt`. If first and last packets are received at the same time or there is only one packet in the data frame, the time difference would be zero and the throughput would be infinite, so the value is set to zero independently. The unit of measurement used is $Mbps$.

- **APP-layer throughput:** it is computed as the ratio of the sum of all Application (APP) packets correctly received during the simulations, and the total time from the first trasnmitted and last received packets, looking from the produced files `udp-app-trace.txt` or `tcp-app-trace.txt`, depending on the used transport protocol. If first and last packets are received at the same time or there is only one packet in the data frame, the time difference would be zero and the throughput would be infinite, so the value is set to zero independently. Sometimes in these kind of plots, the source rate may be represented too. However, it is computed with a different and bigger time interval: in fact, it is the time between the simulation start and stop instants. For that reason, in some plots it may appear that the APP-layer throughput is higher than the source rate, but this is only due to an unfair time comparison.

- **APP-layer latency:** the latency value of each APP packet is directly provided by `udp-app-trace.txt` or `tcp-app-trace.txt` files, depending on the used transport protocol. Both average and median are provided and expressed in ms.

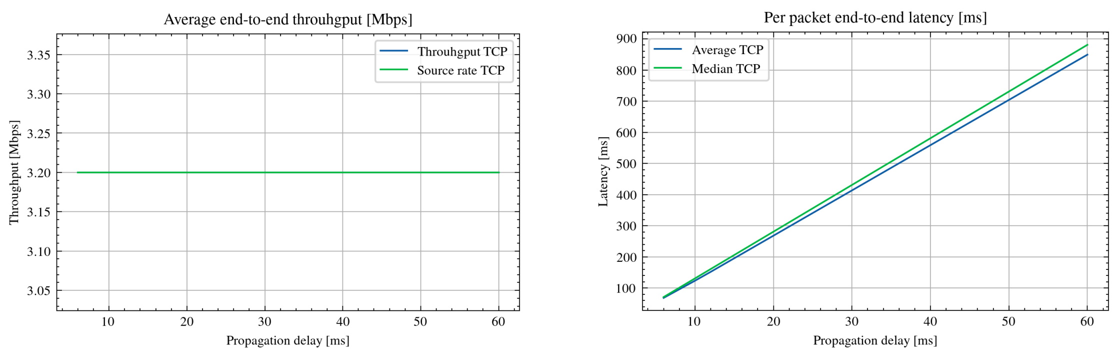
- **APP-layer PDR:** it is the ratio between the number of received and transmitted packets. The results are procuded looking from the produced files `udp-app-trace.txt` or `tcp-app-trace.txt`, and the range of possible values is from 0 to 1.

- **Congestion Window:** it is computed only for TCP case, looking from the produced file `tcp-app-trace.txt`. These plots are useful to analyze how our system performs for specific propagation delay values.

5.2 Comparison with ideal link

The first step of the work was to perform some simulation with the native script `three-gpp-scen-10-delay-test.cc`. It represents the starting point of the *ns3-ntn* module, after the combination between the *ns3-mmwave* baseline and the dedicated channel model developed for NTN communication. The simulated scenario is similar to the one in Figure 4.3 and the `distance` was set to $2000km$ (LEO satellite), in order to guarantee high values of SNR. However, the results did not meet the expectations: for UDP the PHY throughput and latency were totally meaningless, with the PDR fixed to one. Instead for TCP, similar results about PHY throughput, disproportionate latency values and rather low PDR values were obtained. Nevertheless, what we expect to find is that UDP starts to go wrong, in terms of throughput, latency and PDR, when the channel gets worse, while TCP should go wrong independently because its extreme sensitivity in terms of latency, which is being put to a severe stress in NTN systems. Trying to change the `distance` variable and make it as a function of the propagation delay, results get even worse, due to the fact that when the propagation delay grows then SNR decreases, introducing errors into the channel. For this reason the graphs for these specific cases have been omitted.

To better understand what the problems were, a new script was needed: exactly the same as the original one but which exploited an ideal link. For that purpose, `test-ideal-delay.cc` script has been developed. However, working with an ideal link, the APP throughput was equal to the source rate and the PDR fixed to one, as illustrated in Figure 5.1a. Still, there was a problem: the latency in UDP was linear with the propagation delay, as expected, but in TCP the scale factor was approximately about ten (Figure 5.1b).

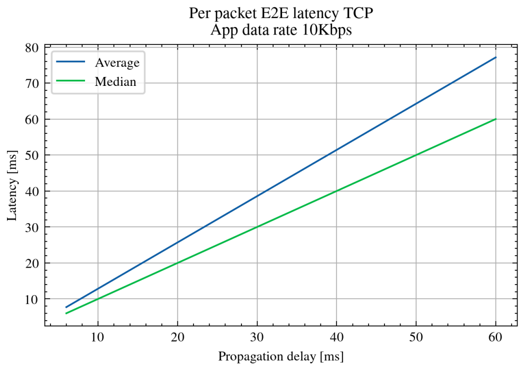


(a) `APP-throughput` (b) `Latency`

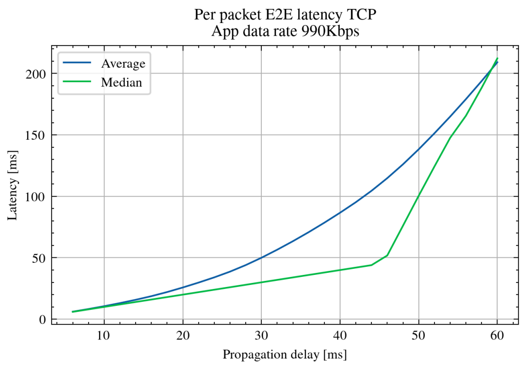
Figure 5.1: `test-ideal-delay.cc` results.

A first hypothesis was to justify the problem by some malfunctioning of the protocol stack. More investigations followed concerning the role of `appDataRate`, scheduler allocations, Channel Quality Indicator (CQI) computations, and so on.

Attempting to print TCP congestion windows for some specific propagation delay values was also unsuccessful. Then, a study on how the TCP latency varied as the `appDataRate` changes is illustrated in Figure 5.2. This problem of TCP, can be justified by the long waiting time before receiving ACKs. In fact, due to the long propagation delay, the value of RTT is much bigger than the application data rate. This causes the packet delay to grow more and more, when propagation delay and application data rate increase too. This is highlighted by the Figure 5.2, in which in 5.2a, the relationship between latency and propagation delay appears linear, whereas in 5.2b it appears as exponential. This is also highlighted by other studies in the literature, such as in [79].



(a) `appDataRate = 10 kbps`



(b) `appDataRate = 990 kbps`

Figure 5.2: TCP latency comparison with different `appDataRate`.

Analysing this issue in depth, it turned out that there were inconsistent delays between consecutive packets. After ensuring that the delays at the RLC level were constant once the propagation delay was fixed, it was necessary to develop an application that transmitted packets sporadically and with a not too large package size. This would allow us to focus more closely on the path of each individual packet. For this reason, `test-leo-interval.cc` script was developed.

5.3 Send Interval Application

The `test-leo-interval.cc` script was created to fulfill the requirement for an application capable of sending packets intermittently and with manageable packet sizes, enabling us to focus more closely on the path of each individual packet. The script is very similar to `three-gpp-scen-10-delay-test.cc`.

Moreover, it does not use the standard available ns-3 application, but new one has been developed: `app-intermittent-send.h`. This latter is able to send packets with a specific `sendInterval` value by `SendPacket` and `ScheduleTx` methods.

5.3.1 UL Scheduling problems

The focus of this subsection is checking the proper functioning of the scheduler. For that purpose, `app-intermittent-send.h` comes to our aid. Through it, we have the possibility to associate each packet with its correspondent SR, grant, transmission and reception. This approach has been useful in identifying a problem in the UL scheduling process. In particular, what appeared was that some grants were not exploited, even though there were certainly packets in the transmission buffer at that specific time instant. Furthermore, the PHY-level receptions of some UL packets appeared out of phase with what was found at the application level. The difference was in fact $100ms$, whereas we would expect at most a few ms, due to the application processing. The problem was related to the exclusion of propagation delay in the slot allocation computation. When a Tx is scheduled in UL, the mechanism to understand when the BS will receive the relevant packet is used. However, it was only used to find out whether it would be possible to schedule more data in that specific slot in the future, but not to apply the correction in the Rx instant by the BS side. Thus, the BS expected to receive the packet immediately after allocating the grant, not waiting for the required RTT (grant delivery + Tx). However, in some cases we were able to receive some packets thanks to some fortunate combinations with reception instances allocated for other packets . This was because the application traffic was periodic, due to the presence of `sendInterval`.
To correct this problem, a resource allocation algorithm, similar to the DL counterpart, was implemented in `src/mmwave/model/mmwave-enb-phy.cc`, in particular in the `StartSlot` method. The goal is to split each slot allocations into the ones used for sending DCIs, thus reporting all scheduled Transmission Time Interval (TTI)s and the ones used for transmission and/or reception in the current slot, which do not comprise UL DATA TTIs.
The first step is to get the current wrong allocations:

```
    m_lastSlotStart = Simulator::Now();
    m_currSlotAllocInfoForDcis = m_slotAllocInfo[m_slotNum];
```

```
 m_currSlotAllocInfoForTxAndRx = m_slotAllocInfo[m_slotNum];

 Time framePeriod = MilliSeconds(10.0);
 Time subframePeriod = MilliSeconds(1.0);
 uint32_t doubleDelayInSubframes = (2 * m_propDelay.
GetMilliSeconds()) / subframePeriod.GetMilliSeconds();
 uint32_t delayedUlDataSubframe = m_sfNum+doubleDelayInSubframes;
 uint32_t delayedUlDataFrame = m_frameNum;
 uint32_t rawFrameDelay = (m_sfNum + doubleDelayInSubframes) / 10;

```

Where all time quantities have been converted into frames/subframes, then we passed to estimate the UL DATA TTIs reception time using the `propDelay`:

```
 //If the delay moves us into the next frames, we need to account
for it and recompute the subframe delay accordingly
 if(rawFrameDelay != 0)
 {
     delayedUlDataSubframe = (m_sfNum+doubleDelayInSubframes)%10;
     delayedUlDataFrame += rawFrameDelay;
 }

 //  Struct holdings frame, subframe and slot
 SfnSf key = SfnSf(delayedUlDataFrame, delayedUlDataSubframe,
                   m_slotNum);

 // increment later the iterator since erase invalidates it
 for (auto tti = m_currSlotAllocInfoForTxAndRx.m_ttiAllocInfo
       .begin();
       tti != m_currSlotAllocInfoForTxAndRx.m_ttiAllocInfo.end();)
 {
     // If DATA and UL
     if((*tti).m_ttiType != TtiAllocInfo::CTRL &&
        (*tti).m_tddMode == TtiAllocInfo::UL_slotAllocInfo)
     {
         if (m_prevAllocatedTtiForUl.find(key) ==
             m_prevAllocatedTtiForUl.end())
         {
             m_prevAllocatedTtiForUl.insert(std::make_pair(key,
             std::vector<TtiAllocInfo> {}));
         }
         // Delaying Tx of UL DATA TTI
         m_prevAllocatedTtiForUl[key].push_back(*tti);
         // Remove UL DATA DCI from receptions expected during
```

```
            // this slot
            m_currSlotAllocInfoForTxAndRx.m_ttiAllocInfo.erase(tti);
        }
        else tti++;
    }

```

After that, finally fixing the previously TTIs:

```
    // Retrieve the previously stored UL DATA TTIs
    key = SfnSf(m_frameNum, m_sfNum, m_slotNum);
    auto prevStoredUlTtis = m_prevAllocatedTtiForUl.find(key);

    if (prevStoredUlTtis != m_prevAllocatedTtiForUl.end())
    {
        for (auto tti : prevStoredUlTtis->second)
        {
            // Look for the first TTI allocated after, and insert
            // before that
            auto nextInsertedTti = m_currSlotAllocInfoForTxAndRx.
                                   m_ttiAllocInfo.begin();
            while (nextInsertedTti->m_dci.m_symStart < tti.m_dci
                   .m_symStart &&
                   nextInsertedTti != m_currSlotAllocInfoForTxAndRx
                   .m_ttiAllocInfo.end())
            {
                nextInsertedTti++;
            }
            // Insert at the proper place to keep intact sorting
            // instead of at the beginning
            m_currSlotAllocInfoForTxAndRx.m_ttiAllocInfo.insert(
            nextInsertedTti, tti);
        }
    }

```

Listing 5.1: UL scheduling correct allocations.

The above algorithm takes into account the minimum delay that the BS has to wait before receiving messages from UE. This obviously works because it assumes that the BS knows exactly the `propDelay`. After this implementation, the study mainly focus on the UL case.

This work has allowed us to identify another problem that `test-ideal-delay` inherits from `three-gpp-scen-10-delay-test`. A few hundred *ms* after the application begins, SRs started appearing every $10ms$. As we might expect, the issue was related to the early expiry of a timer. The `ReportBufferStatusTimer` was in fact set to a value of $10ms$, triggering the mad sending of BSRs. By increasing its value to at least one RTT, the problem was fixed. In addition, this led us to discover a further error in the native *ns3-ntn* module example script: namely the use of `LteRlcUm` for the UE and the `LteRlcUmLowLat` version for the BS. Both were subsequently set to the `LteRlcUm` version. Furthermore at this point, the decision to work with `rrcIdeal` was taken. It allowed us to better distinguish the control packets from the data ones and to avoid some errors in the scheduler associated with high propagation delay values. Indeed, activating `rrcIdeal` enables for the initial control communications at the RLC level to be idealized, avoiding any problems before data transmission even begins.

5.3.2 GBS performance with UDP and TCP

All the previous adjustments allowed us to get a stable and working version of the simulation script. Now, a more precise and punctual study on the functioning of the native GBS will be carried out. In particular, its performance will be tested using different Send Interval (SI) values and adopting UDP and TCP as transport protocols. This will highlight the limitations that the terrestrial protocol stack has in NTN environments and emphasize the limitations of GBS in scenarios with large propagation delays.

GBS: UDP and SI = 100 ms

The first case considered is with UDP and `sendInterval` = 100ms, i.e. 10 pckt/s. A discrepancy between the values of UL receptions at PHY level and APP receptions emerged. Particularly towards the beginning of the simulation, where ULs of 547 bytes at the PHY level matched Rxs of 272 bytes at the APP level. This phenomenon appears only when the propagation delay was bigger than half of SI. This is mainly due to some features of the GBS approach implemented natively in *ns3-ntn*. In fact, with GBS every time a new packet is produced we send a BSR for all Tx buffers, regardless of what we have already sent an SR for. In the case for which the RTT is lower than the SI, then the UE receives each grant before a new packet is even generated, which is why a single packet is always sent at a time. Instead in the opposite case, the UE receives the grant after generating at least one new packet, with the consequent sending of a new BSR that duplicates the request for resources already made by the previous one. This allows the user to send more packets at the same time, due to receiving duplicate grants that communicate the allocation of more resources than necessary. Moreover, this phenomenon leads to a waste of resources at the PHY level.

To clarify this phenomenon, let us take as an example what would happen at the beginning in $SI = 100ms$ case, as illustrated in Figure 5.3.

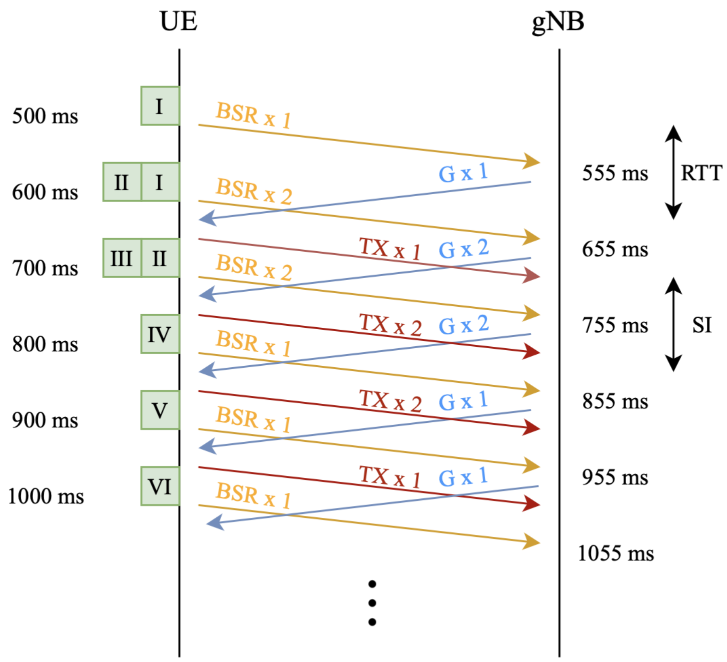
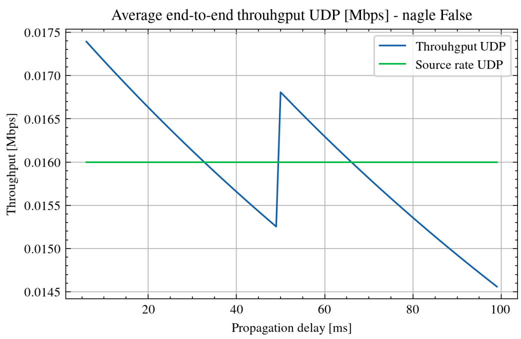


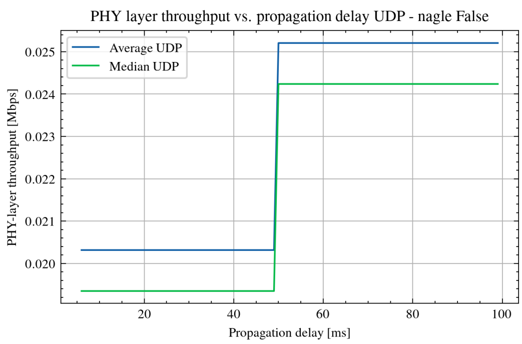
Figure 5.3: Scheduling processes for UDP, $SI = 100ms$ and $RTT > SI$.

When the application starts, an initial packet is generated and a subsequent BSR is transmitted. After $100ms$, a new packet is generated and a BSR for two packets is sent. This is accomplished before the first grant is received by the UE. However, this mechanism does not take into account that an SR has already been sent for the first generated packet, and ends up "duplicating" the request with the second BSR. Subsequently, upon the receipt of the first grant, the user will transmit the first packet. Instead, the reception of the second grant will occur after approximately $100ms + RTT$, i.e. when the UE has already generated the third packet. Still having second and third packets in the buffer and receiving a grant for two packets (associated with the second BSR), the user will send second and third packets together. All this not before the third packet has sent a new BSR. This then generates a snowball effect whereby duplicated BSRs are passed to BS, which by the time the corresponding grants will have expired, since the related packets will have already been transmitted thanks to previous grants. In addition, the current expired grant will not be wasted, because it will be used by newly generated packets. The effect of all this is highlighted by the following results. Please notice that, to achieve a fairer comparison in the various cases, the receiving application was set to close after some extra time, equal to $3t_p$. In this way, any transmitted packets still in transit or waiting for grant would be received in time.

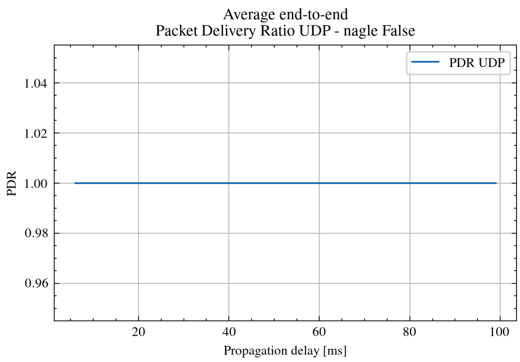
As we can see in Figure 5.4a, APP-throughput is exactly as we expect it to be: the trend is downwards as the same number of packets are transmitted but in a longer time, due to the increase in propagation delay.



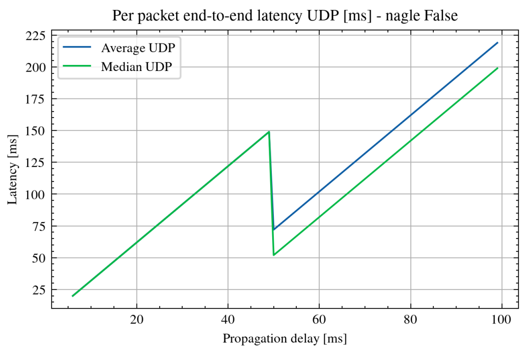
(a) `APP-throughput`



(b) `PHY-throughput`



(c) `PDR`



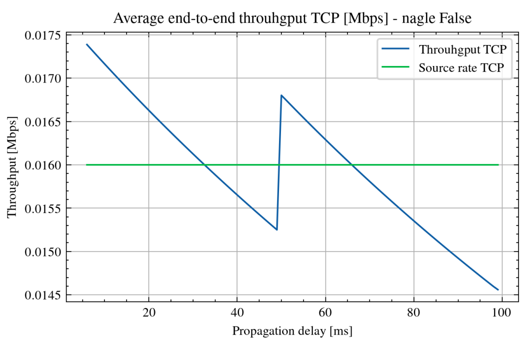
(d) `Latency`

Figure 5.4: GBS results with UDP and SI = 100 ms.

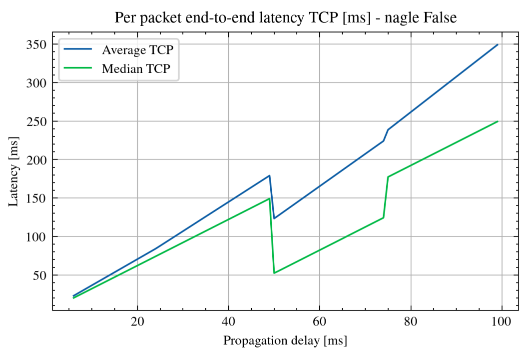
However, there is a spike approximately at $t_p = 50ms$. It corresponds to the case just discussed above, in which the UE is able to send two packets simultaneously, thanks to a "duplicate" grant. In fact, this happens exactly when the RTT exceeds the SI value. The same thing appears in Figure 5.4b and Figure 5.4d. Respectively, in the first the trend is costant and not downhill because in this case the difference in times between the first and last received packet is used, and not between the first transmission and the last reception, as specified in Section 5.1.1. This ensures that the time interval is practically the same, as t_p does not affect it. On the other hand, the spike corresponds to the phenomenon that occurs at 800ms in Figure 5.3: the UE then receives a grant for two packets, but only keeps one in the buffer. This leads to padding which implies waste of resources. Instead, the latency has a constant trend to $3t_p$ as we expect, with the consequent lowering at the same point for what has been discussed so far. After the lucky case at 800ms, subsequent packets no longer have to wait $3t_p$, but only a little more than one. This is because they are able to use expired grants not exploited by previous packets. In fact, each new generated packet is transmitted thanks to the grant associated with the previous packet, but, at the same time, it triggers the sending of a BSR whose grant will be exploited by the next packet. Finally , the PDR graph shows no errors during the simulation.

GBS: TCP AND SI = 100 MS

In the same case but with TCP, performance is almost identical. The PHY-throughput cannot be considered for this type of analysis, as it also counts the various ACKs and control messages used by TCP, so it would not give the right picture of what is going on.



(a) `APP-throughput`



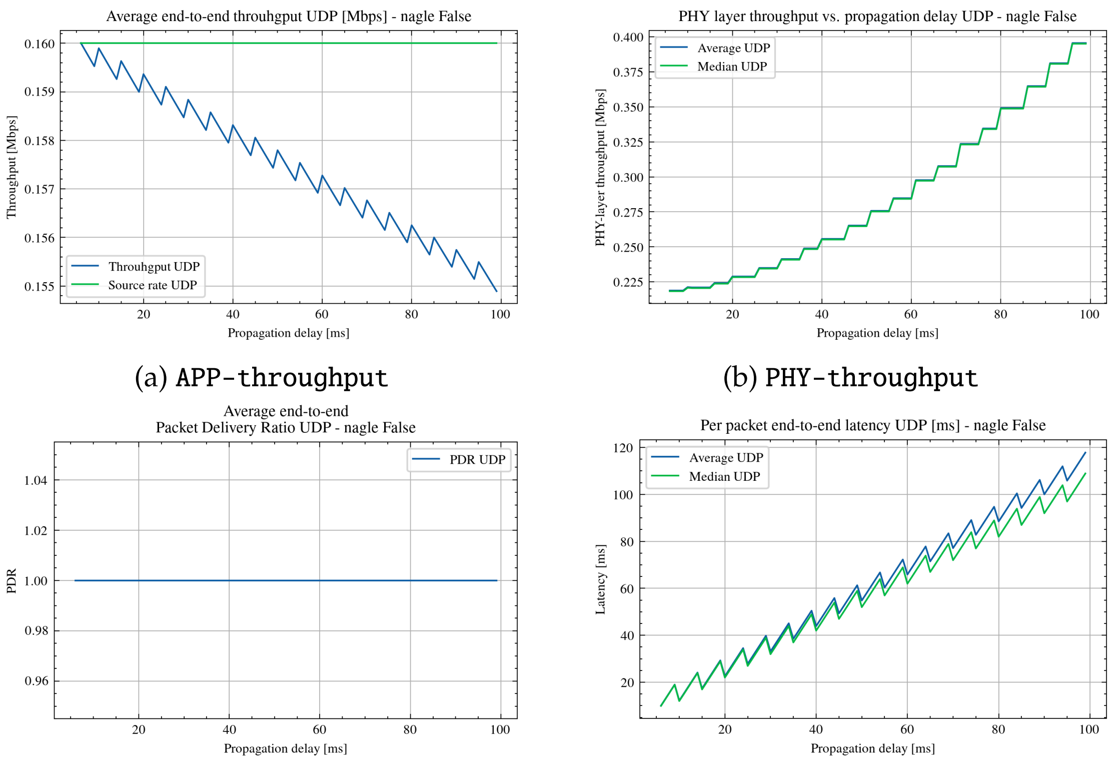
(b) `Latency`

Figure 5.5: GBS results with TCP and SI = 100 ms.

A slight overall drop in APP-throughput and a step up in latency around at $75ms$ are justified by the initial delays introduced by TCP with the three-way handshake procedure and the waiting time for all ACKs. The PDR graph is again fixed at one and is omitted for simplicity.

GBS: UDP and SI = 10 ms

As expected, the APP-throughput in Figure 5.6a is characterized by a decreasing trend as in the previous cases, but with values about 10 times higher due to the lowering of the SI, i.e., increasing of the data rate. Periodically steps up appear: this is the same phenomenon we had in the case with $SI = 100ms$, but with a higher frequency due to the lowering of the SI. In particular, we send more packets together at a time in which with a lower t_p we could not. This happens by sending multiple SRs before receiving each grant, because the RTT goes up as the t_p also goes up.



(a) `APP-throughput` (b) `PHY-throughput`

(c) `PDR` (d) `Latency`

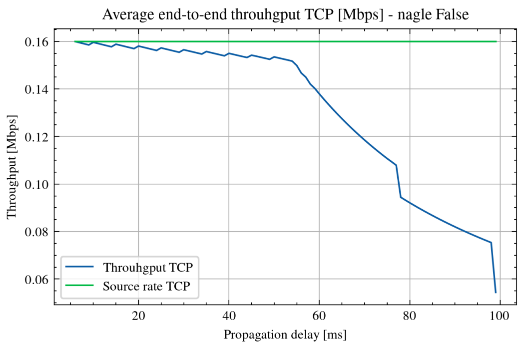
Figure 5.6: GBS results with UDP and SI = 10 ms

In Figure 5.6b, the PHY-throughput increases constantly as the propagation

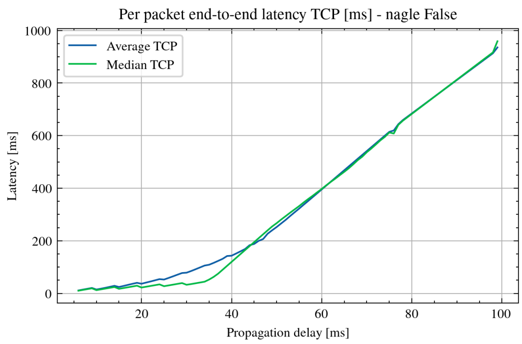
delay increases. This is due to the phenomenon that as we increase t_p, the buffer will get fuller and fuller, as packet generation is constant every $10ms$. Furthermore, with each new packet generated, we send a new BSR. This will allow us to receive very large grants, being able to send many more future packets together, but with a more aggressive use of padding. This is due to the fact that the user receives more transmission opportunities than available packets ready to be transmitted in the buffer. The step increase is due to the discrete and periodic way of sending the packets and how this fits in with the received grants. Each step up corresponds to the moment when we are able to include one more packet in a transmission where in the previous case we could not. This highlights the waste of resources we have as t_p increases using GBS.
Latency has a similar behaviour of SI=100ms, but with many more spikes (Figure 5.6d). These in fact correspond to the times when more packets can be sent together by decreasing the latency. The trend in the very first cases for low t_p is about $3t_p$ on average. However, it is then lowered by the sending of numerous packets with a latency of only one t_p, as they exploit extra BSRs generated by previous packets and never utilised. The periodic drops at $10ms$, $15ms$, $20ms$, and so on correspond to the upward steps in the APP-throughput, while the others always correspond to the cases where we send more packets together than in the case with slightly lower t_p.
Another important observation concerns the latency values obtained. In fact, they are lower than in the same case but with $SI = 100ms$. This appears to be a contradiction, as better performance is achieved in a case where traffic is more aggressive. However, this phenomenon is justified by analyzing the relationship between SI and propagation delay values. In the case with the slower data rate, in fact, the first packets were characterized by an initial transient of around $3t_p$ of latency, which then decreased by a propagation delay. This occurred because newly generated packets were able to exploit grants associated with BSRs of previous packets, thus reducing the delay. In the case where the $SI = 10ms$, the transmission opportunities of UE become ten times greater, due to the decrease of the SI, leading to more chances of lucky transmissions thanks to expired BSRs. All this leads to convergence of the delay values of a little more than one tp, thus improving the performance in terms of latency.
In conclusion, PDR plot in Figure 5.6c is basically perfect as always.

GBS: TCP and SI = 10 ms

Finally, in Figure 5.7, it is possible to analyse a case in which the propagation delay actually adversely affects the system performance. This happens using TCP as transport protocol, due to its high sensitivity in terms of latency. In Figure 5.7a, we can see as the APP-throughput initially performs similarly to the case with UDP, but for $t_p >= 55ms$ things get worse and worse. A similar phenomenon also occurs with regards to latency, in Figure 5.7b. The beginning is again similar to the UDP counterpart, but then a total explosion in transmission delays occurs.



(a) `APP-throughput`



(b) `Latency`

Figure 5.7: GBS results with TCP and SI = 10 ms.

In this case, the system does not work as we would like due to the TCP timers expiring, since the high value of the propagation delay. More work to adapt the TCP transport protocol to NTN scenarios is needed in the future. This behavior is also justified by the fact that the congestion window is reset several times, as we see in Figure 5.8 with $t_p = 99ms$.

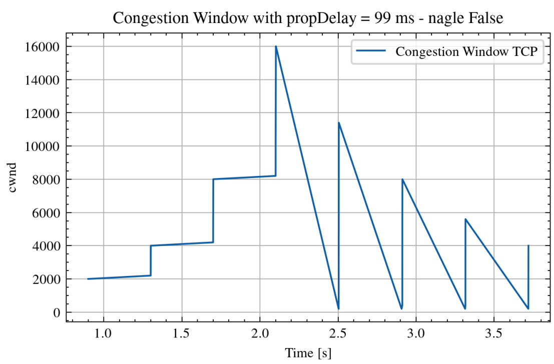


Figure 5.8: Congestion window with TCP, $t_p = 99ms$ and SI = 10 ms.

Enabling HARQ has brought some slight improvement in the graphs, avoiding some packets being lost, but without a big impact. This is because we are working at low rates and considering an almost ideal channel. For higher ones what we expect is that HARQ will even worsen things for its current implementation in the module. In fact, we know that it needs to be changed for non-terrestrial environments, proposing new solutions and enhancements.

All this work has allowed us to better understand the functioning of the GBS implemented in the *ns3-ntn* module. It also made us understand why, as the SI decreases, the PHY-throughput increases if t_p increases too. What happens is that by decreasing the SI, we send more and more BSRs that are increasingly larger before receiving a grant. On the BS side, these large grants are fully exploited, regardless of the number of packets in the buffer. This leads the user to send several packets and using padding in case we do not have enough packets to transmit, with consequent waste of resources. This phenomenon occurs in almost all cases, but obviously it is more evident with high t_p and few SI. In addition, these transmissions of many packets appear at the beginning of the simulation, when the transmission buffers are full of packets, reaching a convergence of BSR and Tx of one packet for the rest of the simulation. This is because little by little we are always able to empty the transmission buffer. Thus leading to transmit each packet with just over one t_p of latency by exploiting BSR of previous packets, but at the same time, requiring new transmission resources, which will be exploited by subsequent packets. What we can conclude is that the PHY-throughput increases when t_p grows up or SI decreases, leading to the advantage of significantly lowering the latency, but at the cost of wasting a lot of PHY resources, when using UDP. While in TCP case, more adaptation work is needed in the future to make it work properly. Finally, in the next chapter we will see other possible scheduling approaches, and we will test them against the GBS.

6

Scheduling with periodic and aperiodic traffic

This chapter focuses on the study of scheduling protocols in periodic and aperidic traffic scenarios. New scheduling solutions that can go beyond the GBS limitations are proposed. In particular, the SPS approach can indeed be useful in optimizing resources, especially in periodic traffic situations. The already discussed GBS problem concerns the sending of duplicate BSRs. The UE sends a BSR every time a packet is generated and the BS schedules grants based on these reports. The first BSR correctly reports the packets contained in the buffer from the user side. However in situations where $RTT > SI$, the second BSR already duplicates the request for the first packets, as the first grant will arrive after the transmission of this latter. This effect gets worse as the t_p increases and the SI decreases, i.e. as the data rate increases, leading to the minimization of packet latencies to one t_p, at the cost of a large waste of physical resources. Furthermore, this thing sabotages any traffic prediction or SPS mechanisms by the BS.

Currently, the $ns3 - ntn$ module does not have an SPS version implemented yet. For this reason, a dedicated version has been designed and implemented. In particular, it can be activated by enabling the `enableSPS` flag. Even though from 3GPP standards we expect that scheduling protocols are implemented on the BS side, this version is implemented from the UE one. In fact, it is designed for applications that can distinguish the type of their traffic. For instance, an environmental monitoring application for which when nothing happens only periodic control messages are sent and when something environmental phenomenon occurs the data rate increases and the traffic becomes aperiodic, due to all feedback signals. The implementation from the UE side also allows to choose the best scheduling approach according to the traffic type, without waiting for a propagation delay, as the application can realize it almost instantly.

Our goal is to test the performance of the two types of schedulers under periodic and aperiodic traffic conditions, to understand which one is more convenient to use in different cases. What we expect to do is to use the SPS in periodic traffic situations and GBS with the aperiodic one, using a higher overhead, but achieving greater precision and punctuality in this case.

6.1 Semi-Persistent Scheduler implementation

SPS has been implemented on the user side and it works by managing the sending of BSRs. Then, the BS simply allocates resources based on the requests it receives. For this purpose, `ReportBufferStatusTimer` and `ReordiningTimer` values at RLC level have been increased, to ensure they do not expire during the scheduling process. The main changes have been made in `lte-rlc-um.h` and `lte-rlc-um.cc`. The protocol works so that when the application starts, and more precisely when the first packet is produced, a BSR is sent including the packet itself and some future ones. The choice of the size is done according with RTT. In fact, how many packets will be produced in that time interval is calculated, and the request for resources will be made to ensure that they can all be transmitted together, after the reception of the first grant. The calculation is done in line with the following formula:

$$num_{pckt} = \left\lfloor \frac{RTT}{SI} \right\rfloor + 1 \tag{6.1}$$

In conclusion, the scheduling process occurs as illustrated in Figure 6.1. The mechanism is able to predict the number of initial packets generated in a RTT and then send single BSRs for each packet, starting from the knowledge of the propagation delay and the packet production period. The sending time of each individual BSR is chosen so that the corresponding grant is received just after the generation of a new packet. This allows to lower the latency of all packets, except for the initial ones, to one t_p and optimizes the use of physical resources used in UL.

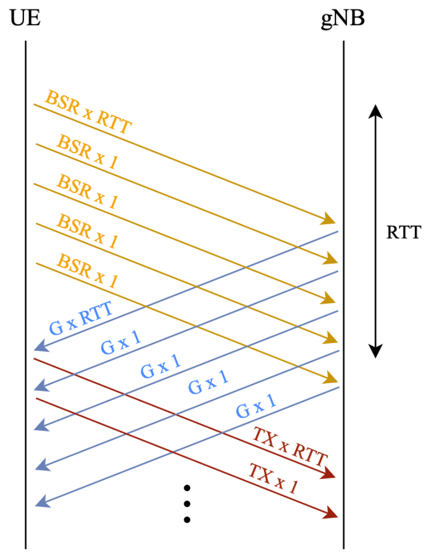


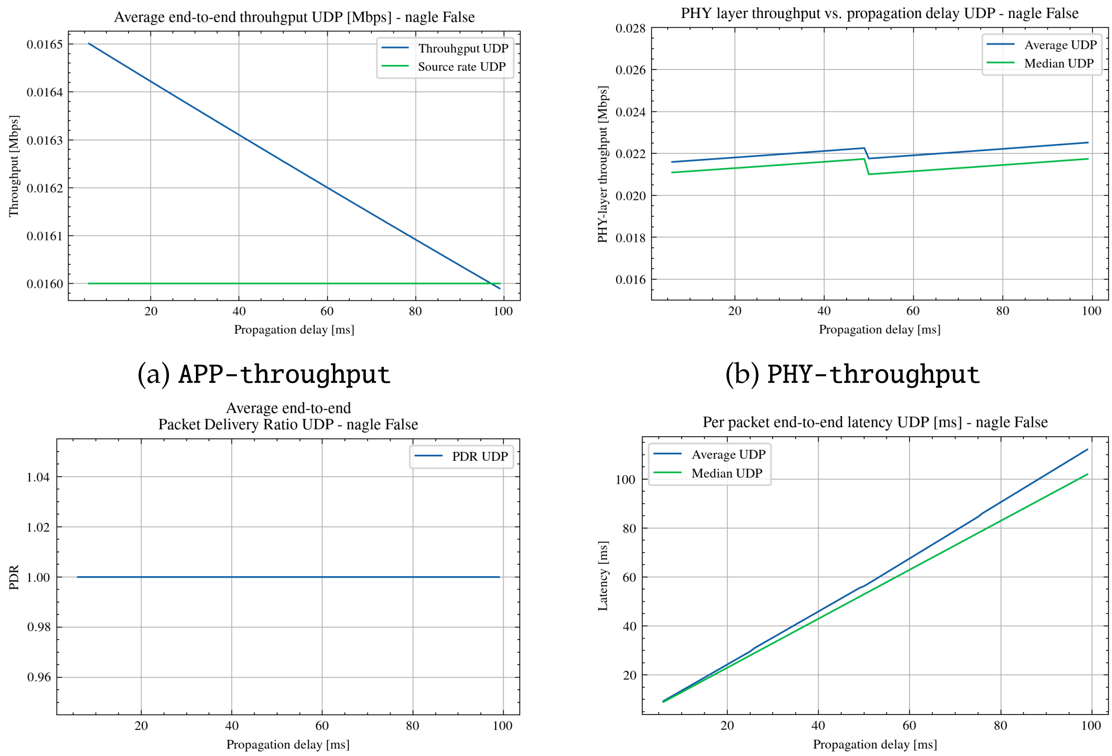
Figure 6.1: Semi-Persisten Scheduling process.

6.2 Periodic traffic

As first step, we analyze the performance of SPS in periodic traffic situation. In order to have a fair comparison, the same scenarios studied for the GBS case will be proposed, i.e. $SI = 100ms$ and $SI = 10ms$, as analyzed in Section 5.3.2. Furthermore, parameters such as simulation time, fixed distance and other script variables will be configured in the same way.

SPS: UDP and SI=100ms

At the APP-throughout level, the performances correspond to those we expect (Figure 6.2a). The trend is in fact of constant linear decreasing over time. This is because we no longer have the phenomenon that appeared in the GBS, whereby in some lucky combinations, it was possible to transmit packets faster than expected by exploiting duplicate BSRs, depending on the relationship between SI and RTT. Here each packet is transmitted in the shortest possible time, as a function of propagation delay. Please remember that the reason why throughput is higher than source rate is due to the different time intervals with which they are computed, as explained in Section 5.1.1.



(a) `APP-throughput` (b) `PHY-throughput`

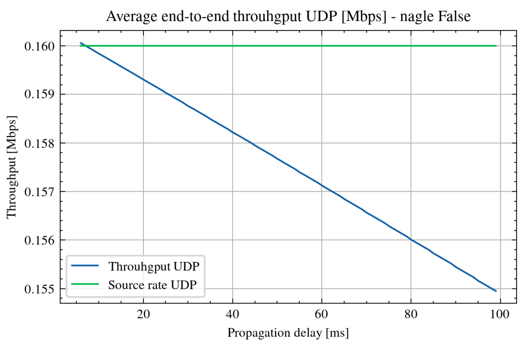
(c) `PDR` (d) `Latency`

Figure 6.2: SPS results with UDP and SI = 100 ms.

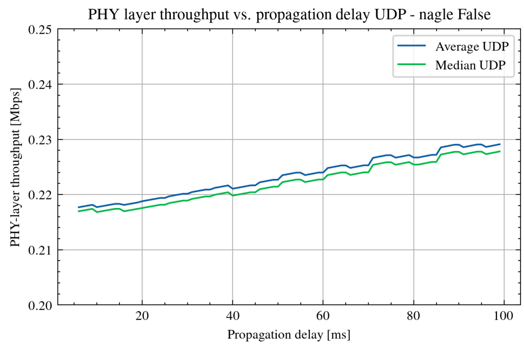
The big first difference is represented in 6.2b, where the PHY-throughput performance is practically constant over time. This is enabled by SPS's more intelligent scheduling technique, which never requires unnecessary extra resources, thus minimizing the overhead at the PHY level. The small step at $50ms$ corresponds to the size variation of the first BSR. In fact, when $SI = 100ms$, for $t_p < 50ms$ it includes only one packet, while in the opposite case it includes two. However, the trend is not horizontal as in Figure 5.4b, but it varies slightly as t_p increases, due to the implementation of SPS. In fact, it schedules Txs a few fractions of a millisecond after the generation of packets, in order to be sure to transmit them correctly. The PDR in Figure 6.2c is basically perfect and equivalent to the GBS case. Finally, the latency has a fixed linear growth trend to one t_p. This is due to the fact that, apart from the packets contained in the first BSR, all subsequent ones are characterized by a delay equal to a few ms more than the propagation delay.

SPS: UDP and SI=10ms

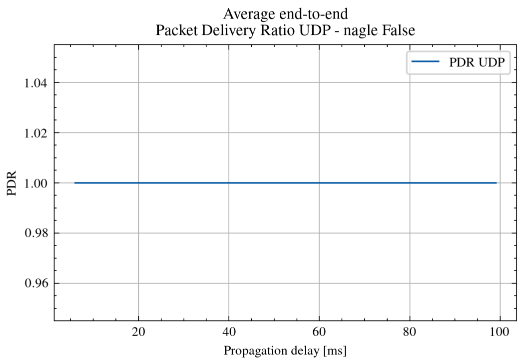
This case turns out to be very similar to the previous one just discussed, with some differences due to the variation of the SI. The APP-trhoughput has a linear downward trend, with values similar to the GBS counterpart (Figure 6.3a). As illustrated in Figure 6.3b, the PHY-throughput has a slight growth trend due to the size of the first BSR as a function of the RTT, but it remains pratically constant as the propagation delay increases. While in the GBS case the waste grew very quickly as the t_p increased, as huge grants were wasted mostly by padding procedures. The PDR is always perfect for all t_p values (Figure 6.3c). In conclusion, the latency trend also remains linearly growing proportional to just over one t_p, as illustrated in Figure 6.3d.



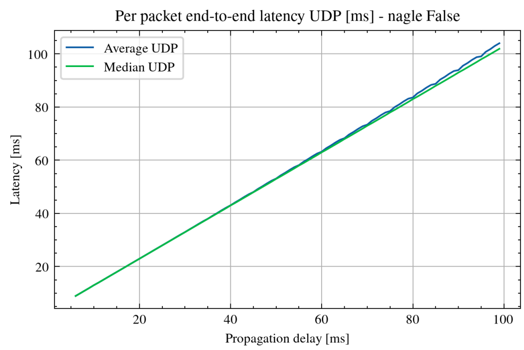
(a) `APP-throughput`



(b) `PHY-throughput`



(c) `PDR`



(d) `Latency`

Figure 6.3: SPS results with UDP and SI = 10 ms.

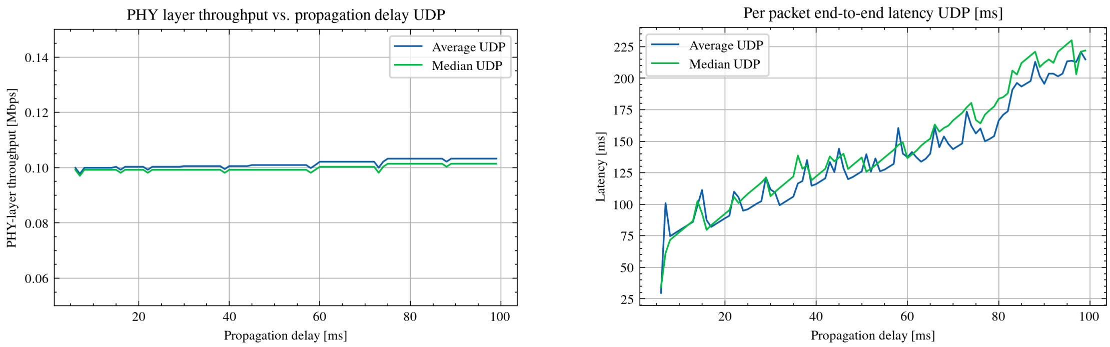
We can therefore conclude that the SPS approach allows us to obtain very similar performances (approximately of one t_p) to the GBS counterpart in terms of latency, but at the same time, it allows to optimize the use of physical resources and the consequent wastes. These results can be achieved only starting from the knowledge of the propagation delay between UE and BS, and assuming that the traffic is periodic. In the next section we will investigate what happens when this last assumption is missing, and we will test the performances of SPS and GBS with an application switching from periodic to aperiodic traffic and vice versa: *Crazy Application*.

6.3 Aperiodic Traffic

This section is dedicated to the study of the performance of the proposed scheduling protocols in aperiodic traffic situations. As we know, the GBS approach is well suited to any type of scenario, with all its advantages and limitations. On the other hand, the SPS approach would make no sense to be used in aperiodic traffic situations, so it needs to be adapted. For the purpose, it was necessary to implement a new type of application, called *Crazy Application*, which can be activated with the `enableCrazyApplication` flag. In fact, it is designed to simulate applications, such as environmental monitoring, in which most of the time only periodic control messages are transmitted, while when an environmental phenomenon occurs the traffic becomes aperiodic and characterized by a higher transmission rate. The chosen scenario for the simulations is the following: the application time is divided into three intervals. In the first and last one the traffic will be periodic with constant SI equal to $30ms$, while in the second one the traffic will be aperiodic with $SI = 15ms$ on average, but with the possibility of oscillating with a margin of 10% of the periodic SI, thus introducing some randomness in the traffic. GBS approach will be maintained for the entire simulation time and therefore it will adapt automatically to the type of traffic used. For SPS, we propose a new scheduler implementation, referred to as *Smart Scheduler*, with the assumption that the end user is able to understand the type of traffic (i.e., periodic or aperiodic) generated at the application with no additional delays. In this way, *Smart Scheduler* can act as SPS in situations of periodic traffic, and as GBS otherwise. This is also made possible by the implementation of the scheduling protocol on the UE side. If it had been implemented on the BS side, as expected from 3GPP standards, a traffic prediction algorithm on the BS side would have been needed and the whole process would have been delayed by at least one propagation delay, due to the long distance between user and satellite. UE is the first to realize that its traffic has changed so it makes sense to intervene on the user side by managing the sending of BSRs. Before moving on the comparison between the two cases, let's see why it is not convenient to use the SPS approach with aperiodic traffic, but instead it is necessary to develop the *Smart Scheduler*.

SPS: UDP and Crazy Application

In this case, the SPS approach is in no way able to predict, and consequently adapt, to variations in the type of traffic. For this reason, it works as if the application always generates packets with $SI = 30ms$, even in cases where the traffic becomes aperiodic. The results are illustrated in Figure 6.4



(a) `PHY-throughput` (b) `Latency`

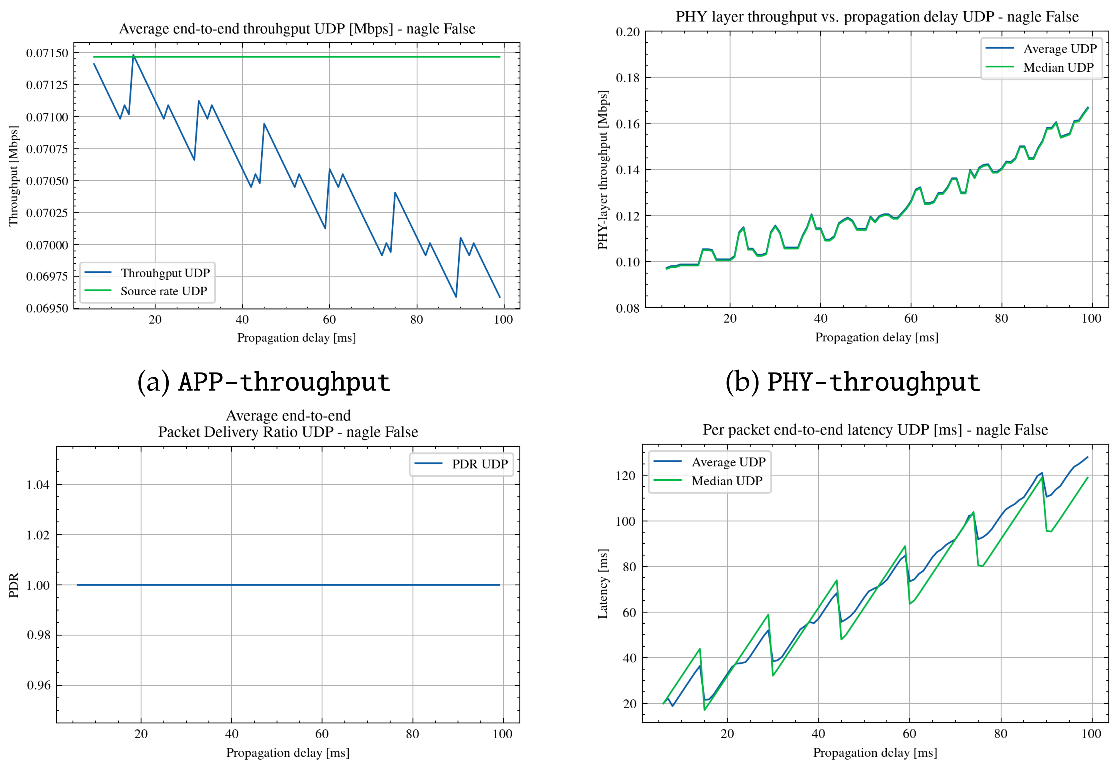
Figure 6.4: SPS results with UDP and Crazy Application.

As expected, Figure 6.4a shows the constant behavior of the PHY-throughput as the propagation delay varies. In fact, it coincides with the cases in which we tested the SPS approach in periodic traffic situations: the way in which transmissions are scheduled is completely constant in the simulation time, and decided at the beginning, starting from the knowledge of the propagation delay. Instead, Figure 6.4b illustrates the real reason why it is not convenient to use the standard SPS approach in aperiodic traffic conditions. What happens is that the initial packets are scheduled correctly, while when the type of traffic changes, the packets accumulate more and more delay leading to a sharp deterioration of performance in terms of latency. The reliability of the UDP transport protocol remains high in this case with a fixed `distance` of $2000km$, but large degradations are found in the time in which I can correctly receive each packet.

We can now move on to comparing the results obtained using GBS and *Smart Scheduler* approaches with the *Crazy Application*.

GBS: UDP and Crazy Application

In Figure 6.5a we can observe that the trend is exactly the same as already obtained in the GBS study in periodic traffic situations. On average, a linear decrease in throughput is obtained, characterized by some steps due to the relationship between SI and RTT. Furthermore, the presence of some noise in the function is due to the randomness introduced by aperiodic traffic situations.



(a) `APP-throughput`

(b) `PHY-throughput`

(c) `PDR`

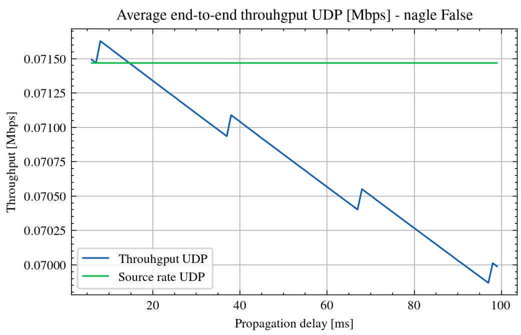
(d) `Latency`

Figure 6.5: GBS results with UDP and Crazy Application.

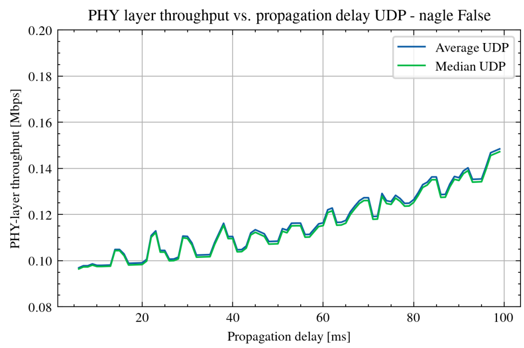
The same can be said for the results in Figure 6.5b: the trend is substantially linear growth, with a lower speed than the case with fixed $SI = 10ms$ studied in Section 5.3.2. Furthermore, the results are mainly due to the inefficiency that we have in the aperiodic traffic interval, which adds a greater waste of resources at the physical level, and in addition introduces randomness in the function. In conclusion, the results in Figures 6.5c and 6.5d are similar with those already obtained in Section 5.3.2: the PDR is always perfect and the latency values are slightly higher than a propagation delay, with the usual serrated trend due to the relationship between SI and RTT.

Smart Scheduler: UDP and Crazy Application

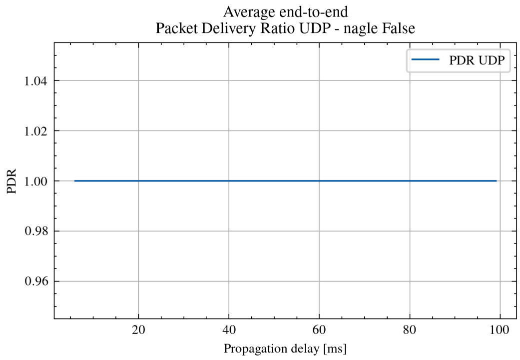
In Figures 6.6a, 6.6c and 6.6d we obtain results very similar to those of the previous case, in particular the latency values settle at just over one t_p. Please notice that the APP-throughput and latency behavior is no longer perfectly linear as in the only periodic case. In particular, it has a jagged behavior and a certain margin of randomness, due to the aperiodic traffic interval in which the GBS approach is used.



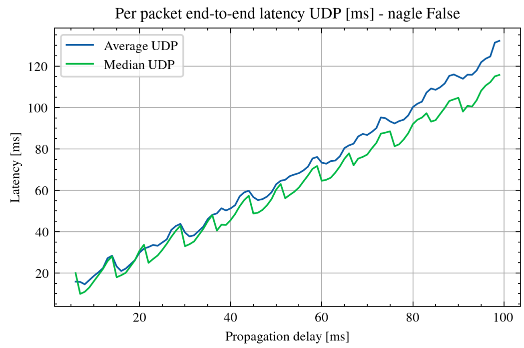
(a) `APP-throughput`



(b) `PHY-throughput`



(c) `PDR`

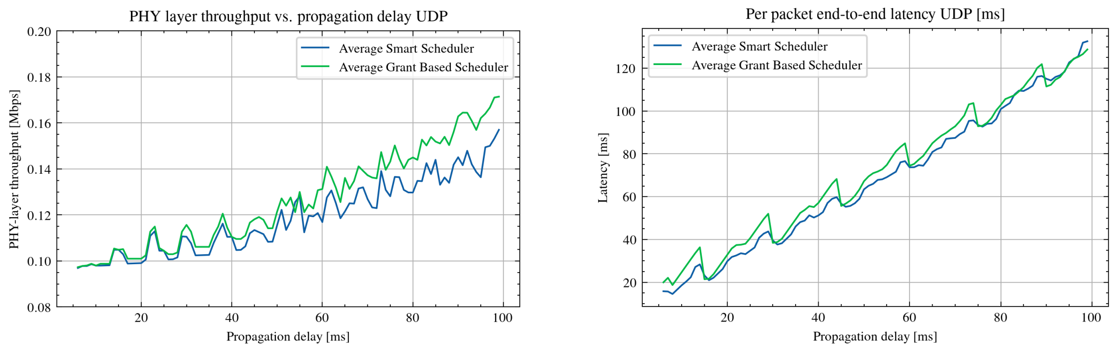


(d) `Latency`

Figure 6.6: Smart Scheduler results with UDP and Crazy Application.

In conclusion, the graph that really allows us to understand the differences in performance between the two approaches is the one in Figure 6.6b. Again, there is some noise and pseudo-periodic behavior due to the aperiodic nature of the traffic in the second interval and the resulting use of GBS. The trend appears to be upward, but by carefully analyzing the graph some improvements can be highlighted. The results are dominated by the huge waste of resources due to the decrease of SI, i.e. increase of data rate, and by the use of GBS in the aperiodic interval. On the other hand, the use of SPS in the intervals with periodic traffic allows to minimize the use of physical resources. This is mainly noticeable for high propagation delay values. In fact, when the t_p has low values, the performances are almost equivalent, with a slight improvement with the use of SPS. While with high propagation delay values, the difference becomes clearer, with a saving of $15Kbps$, which approximately corresponds to 10% of physical resources.

Similar results are obtained if we carry out the same simulations but varying the value of the `distance` variable. In Figure 6.7, it is in fact calculated as a function of the propagation delay used, with a consequent impact on the SNR computation, which will gradually decrease as the propagation delay increases.



(a) `PHY-throughput` (b) `Latency`

Figure 6.7: GBS and Smart Scheduler with variable distance.

Figure 6.7a illustrates the substantial draw in terms of latency performance between the two approaches: both in fact stand at little more than one propagation delay, mainly due to the initial transients of the two protocols which lead to a latency of around $3t_p$ for the transmission of the first packets. Similarly, similar results to the fixed-distance case are also present in Figure 6.7b. Where the small minimal difference in terms of PHY-throughput becomes increasingly evident as the propagation delay increases. In conclusion, the obtained plots in the cases with fixed or variable `distance` illustrates the same results, except for the introduction of greater noise, especially for high propagation delays, in which the SNR value gets worse and worse, with consequent degradation of the communication channel.

As already highlighted, the big waste is dominated by the large use of physical resources that occurs in the aperiodic interval. If we imagine the use of this type of scheduling in scenarios where the length of periodic intervals is much longer than the aperiodic ones, this can lead to an even greater performance improvement. Another possible point of growth could be to replace the GBS approach in aperiodic intervals with a traffic prediction algorithm. This could lead to a further improvement of the scheduler performance, by focusing mainly on how to handle aperiodic traffic. However, this thesis has highlighted how even just optimizing the scheduler work in periodic traffic situations, it is possible to achieve great enhancements in overall performance, in particular in terms of overhead (PHY-throughout) and latency.

7 Conclusions

This thesis has allowed to highlight the importance of scheduling processes within cellular communications in NTN scenarios. In particular, emphasizing the challenges to be faced and overcome in order to obtain an efficient communication system even in these extreme environments. Precisely, parameters such as propagation delay, latency and PHY-throughput have been taken into account. The work started from the existing *ns3-ntn* module developed by the Department of Information Engineering of the University of Padua. It has been expanded, corrected and adapted for the specific objectives of this thesis. In the first phase the focus was to study the performance of transport protocols such as UDP and TCP, highlighting their weakness, particularly in relation to propagation delay. Once this was done, a targeted study on the performance of the Grant Based Scheduler under periodic traffic conditions was carried out. Then, a dedicated Semi-Persistent Scheduler has been developed and implemented, showing its advantages over the module's native scheduling technique. Finally, the two approaches were compared with a specific application capable of changing its traffic type from periodic to aperiodic and vice versa. The adaptability of the *Smart Scheduler* approach showed a fair balance in terms of latency and large advantages in terms of overhead, managing to decrease the PHY-throughput.
Possible future work could focus on the realization of traffic prediction mechanisms specific for aperiodic intervals and on the extension of this study in situations with multiple moving users, who vary their distance from the satellite, inevitably changing also the related propagation delay.

References

[1] IBM. *The future of 5G: What to expect from this transformational technology*. URL: https://www.ibm.com/blog/5g-future/.

[2] Dhanashree Shukla and Sudhir D. Sawarkar. "A study of wireless network evolution from 4G to 5G: standalone vs non-standalone". In: *2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*. 2022, pp. 1–6. DOI: 10.1109/SMARTGENCON56628.2022.10084020.

[3] Redgate. *What's the real story behind the explosive growth of data?* URL: https://www.red-gate.com/blog/database-development/whats-the-real-story-behind-the-explosive-growth-of-data.

[4] Mckiney&Company. *What is 5G?* URL: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-5g.

[5] *Big Data Statistics 2023: How Much Data is in The World?* Accessed: 2024-08-15. URL: https://firstsiteguide.com/big-data-stats/#:~:text=There%20were%2079%20zettabytes%20of,big%20data%20will%20need%20analysis..

[6] RPC Senate. *THE IMPORTANCE OF 5G*. URL: https://www.rpc.senate.gov/policy-papers/the-importance-of-5g.

[7] Nadezhda N. Pokrovskaia and Alexandr K. Garin. "Strategies for Bridging the Digital Divide by Enhancing Communication Competence". In: *2022 Communication Strategies in Digital Society Seminar (ComSDS)*. 2022, pp. 166–170. DOI: 10.1109/ComSDS55328.2022.9769146.

[8] Andrew Lappalainen and Catherine Rosenberg. "Bridging the digital divide: success depends on content provider and application developer involvement [point of view]". In: *Proceedings of the IEEE* 109.1 (2021), pp. 2–10. DOI: 10.1109/JPROC.2020.3028611.

[9] Mehmet Izzet Saglam. "Conditional Handover for Non-Terrestrial Networks". In: *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*. 2023, pp. 1–5. DOI: 10.1109/WINCOM59760.2023.10322919.

[10] Hao Lin, Mustafa A. Kishk, and Mohamed-Slim Alouini. "Virtual Backhaul Connectivity for Enhanced Coverage in Fiber-Less Areas". In: *IEEE Wireless Communications* 31.4 (2024), pp. 324–330. DOI: 10.1109/MWC.012.2300371.

[11] Mario Minardi et al. "SDN-based Testbed for Emerging Use Cases in Beyond 5G NTN-Terrestrial Networks". In: *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. 2023, pp. 1–6. DOI: 10.1109/NOMS56928.2023.10154319.

[12] Alessandro Vanelli-Coralli et al. "Non-Terrestrial Networks Overview". In: *5G Non-Terrestrial Networks: Technologies, Standards, and System Design*. 2024, pp. 53–90. DOI: 10.1002/9781119891185.ch3.

[13] Marco Mezzavilla et al. "End-to-End Simulation of 5G mmWave Networks". In: *IEEE Communications Surveys and Tutorials* 20.3 (2018), pp. 2237–2263. DOI: 10.1109/COMST.2018.2828880.

[14] Mattia Sandri. *3GPP NTN Channel Model Implementation into ns- 3*. 2022. URL: https://gitlab.com/mattiasandri/ns-3-ntn/-/tree/ntn-dev.

[15] Frederick W. Vook et al. "5G New Radio: Overview and Performance". In: *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. 2018, pp. 1247–1251. DOI: 10.1109/ACSSC.2018.8645228.

[16] *Iridium Communications*. URL: https://en.wikipedia.org/wiki/Iridium_Communications (visited on 08/15/2024).

[17] Microwave Product Digest. *Demystifying Non-Terrestrial Networks*. URL: https://www.mpdigest.com/2024/05/20/demystifying-non-terrestrial-networks/.

[18] StarLink. *Satellite Technology*. URL: https://www.starlink.com/technology.

[19] David Lumb. *SpaceX Puts Its First Satellites in Orbit to Connect T-Mobile's Customers*. 2024. URL: https://www.cnet.com/tech/mobile/spacex-launches-first-satellites-to-connect-t-mobiles-customers/ (visited on 08/15/2024).

[20] Husnain Shahid et al. "Emerging Advancements in 6G NTN Radio Access Technologies: An Overview". In: *2024 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. 2024, pp. 593–598. DOI: 10.1109/EuCNC/6GSummit60053.2024.10597037.

[21] Marco Giordani and Michele Zorzi. "Non-Terrestrial Networks in the 6G Era: Challenges and Opportunities". In: *IEEE Network* 35.2 (2021), pp. 244–251. DOI: 10.1109/MNET.011.2000493.

[22] Juan Pedro Tomás. "6G-NTN shows importance of NTNs for 6G in new white paper". In: *RCR Wireless News* (2024). Available online: https://www.rcrwireless.com/20240408/6g/6g-ntn-shows-importance-ntns-6g-new-white-paper.

[23] Ju-Hyung Lee et al. "Integrating LEO Satellite and UAV Relaying via Reinforcement Learning for Non-Terrestrial Networks". In: *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. 2020, pp. 1–6. DOI: 10.1109/GLOBECOM42002.2020.9348105.

[24] Rohde&Schwarz. *Reti non terrestri (NTN)*. URL: https://www.rohde-schwarz.com/it/soluzioni/test-and-measurement/wireless-communication/standard-cellulari/5g-test-and-measurement/non-terrestrial-networks-ntn/-reti-non-terrestri-ntn_256719.html#media-gallery-6.

[25] Zaid Abdullah et al. "Integrated Access and Backhaul via Satellites". In: *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2023, pp. 1–6. DOI: 10.1109/PIMRC56721.2023.10294025.

[26] "IEEE Approved Draft Standard for a Framework for Structuring Low Altitude Airspace for Unmanned Aerial Vehicle (UAV) Operations". In: *IEEE P1939.1/D6.0, October 2020* (2021), pp. 1–92.

[27] Iskandart, S. Gratsia, and M. E. Ernawan. "LTE uplink cellular capacity analysis in a High Altitude Platforms (HAPS) communication". In: *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*. 2017, pp. 1–5. DOI: 10.1109/TSSA.2017.8272889.

[28] Gao Yuxuan, Li Yue, and Shi Penghui. "Research Status of Typical Satellite Communication Systems". In: *2021 19th International Conference on Optical Communications and Networks (ICOCN)*. 2021, pp. 1–3. DOI: 10.1109/ICOCN53177.2021.9563909.

[29] ESA. *Types of orbits*. URL: https://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits.

[30] Marko Höyhtyä et al. "Multi-Layered Satellite Communications Systems for Ultra-High Availability and Resilience". In: *Electronics* 13.7 (2024). ISSN: 2079-9292. DOI: 10.3390/electronics13071269. URL: https://www.mdpi.com/2079-9292/13/7/1269.

[31] Dengke Wang et al. "The Potential of Multilayered Hierarchical Nonterrestrial Networks for 6G: A Comparative Analysis Among Networking Architectures". In: *IEEE Vehicular Technology Magazine* 16 (2020), pp. 99–107. URL: https://api.semanticscholar.org/CorpusID:236882760.

[32] Pansoo Kim et al. "A Study on the development of a modem verification environment for 3GPP NR NTN". In: *2022 International Conference on Electronics, Information, and Communication (ICEIC)*. 2022, pp. 1–2. DOI: 10.1109/ICEIC54506.2022.9748763.

[33] Yun Li et al. "Review of IoT NTN Core Network Enhancement Technologies". In: *2024 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. 2024, pp. 1–6. DOI: 10.1109/BMSB62888.2024.10608221.

[34] Rohde&Schwarz. *Non-Terrestrial networks (NTN)*. URL: https://www.rohde-schwarz.com/us/solutions/test-and-measurement/wireless-communication/cellular-standards/5g-test-and-measurement/non-terrestrial-networks-ntn/non-terrestrial-networks-ntn_256719.html.

[35] V. Arun and M. Azhagiri. "Design of Long-Term Evolution Based Mobile Edge Computing Systems to Improve 5G Systems". In: *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*. 2023, pp. 160–165. DOI: 10.1109/ICECAA58104.2023.10212420.

[36] Jaime Galán-Jiménez and Alfonso Gazo-Cervero. "Overview and Challenges of Overlay Networks: A Survey". In: *International Journal of Computer Science and Engineering Survey* 2 (Feb. 2011), pp. 19–37. DOI: 10.5121/ijcses.2011.2102.

[37] Benjamin Finley and Alexandr Vesselkov. "Cellular IoT Traffic Characterization and Evolution". In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. 2019, pp. 622–627. DOI: 10.1109/WF-IoT.2019.8767323.

[38] Alessandro Traspadini et al. "On the Energy Consumption of UAV Edge Computing in Non-Terrestrial Networks". In: *2023 57th Asilomar Conference on Signals, Systems, and Computers*. 2023, pp. 1684–1690. DOI: 10.1109/IEEECONF59524.2023.10476934.

[39] 3rd Generation Partnership Project. *3GPP – Home*. Accessed: 2024-06-12. 2024. URL: https://www.3gpp.org/.

[40] 3GPP. *New Radio (NR); Radio Frequency (RF) requirements analysis*. Technical Specification (TS) 38.811. Version 16.6.0 (2023-12). 3rd Generation Partnership Project (3GPP), 2023. URL: https://www.3gpp.org/ftp/Specs/archive/38_series/38.811.

[41] Zhuoer Zhang et al. "A Joint Frequency Offset Estimation Algorithm for 5 G NTN". In: *2024 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. 2024, pp. 1–6. DOI: 10.1109/BMSB62888.2024.10608289.

[42] S.Popa, Draghiciu N, and Reiz R. "Fading Types in Wireless Communications Systems". In: *Journal of Electrical and Electronics Engineering* 1 (May 2008).

[43] *Inter-carrier Interference*. URL: https://www.sciencedirect.com/topics/computer-science/intercarrier-interference.

[44] Donglin Wang et al. "Investigating the Impact of Variables on Handover Performance in 5G Ultra-Dense Networks". In: *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. 2023, pp. 567–572. DOI: 10.1109/EuCNC/6GSummit58263.2023.10188324.

[45] Wei Zhang, Shidong Zhou, and Xiujun Zhang. "An Asynchronous Grant-Free Multiple Access Scheme with Rateless Codes for URLLC". In: *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. 2020, pp. 1–5. DOI: 10.1109/VTC2020-Fall49728.2020.9348873.

[46] NR Explained Team. *Timing Advance in 5G NR*. Accessed: Month Year. 2024. URL: https://www.nrexplained.com/ta.

[47] Chaitali J. Pawase and KyungHi Chang. "Demodulation Reference Signal (DM-RS) based channel estimation for non-terrestrial networks to support high mobility". In: *ICT Express* 10.1 (2024), pp. 46–52. ISSN: 2405-9595. DOI: `10.1016/j.icte.2023.05.004`. URL: `https://www.sciencedirect.com/science/article/pii/S2405959523000589`.

[48] Ural Mutlu and Yasin Kabalcı. "Performance Analyses of Hybrid-ARQ in Fifth Generation New Radio". In: *2021 3rd Global Power, Energy and Communication Conference (GPECOM)*. 2021, pp. 269–274. DOI: `10.1109/GPECOM52585.2021.9587674`.

[49] Flavien Ronteix–Jacquet. "Reducing latency and jitter in 5G radio access networks". Theses. Ecole nationale supérieure Mines-Télécom Atlantique, Dec. 2022. URL: `https://theses.hal.science/tel-04012662`.

[50] Faris Syahmi Samidi et al. "5G New Radio: Dynamic Time Division Duplex Radio Resource Management Approaches". In: *IEEE Access* PP (Aug. 2021), pp. 1–1. DOI: `10.1109/ACCESS.2021.3104277`.

[51] *Frequency Division Duplexing*. URL: `https://www.sciencedirect.com/topics/engineering/frequency-division-duplexing`.

[52] Yinan Qi et al. "On the Phase Tracking Reference Signal (PT-RS) Design for 5G New Radio (NR)". In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. 2018, pp. 1–5. DOI: `10.1109/VTCFall.2018.8690852`.

[53] Xiaoran Liu et al. "Peak-to-Average Power Ratio Analysis for OFDM-Based Mixed-Numerology Transmissions". In: *IEEE Transactions on Vehicular Technology* 69.2 (2020), pp. 1802–1812. DOI: `10.1109/TVT.2019.2960801`.

[54] 3GPP. *New Radio (NR); System Requirements for Non-Standalone (NSA) and Standalone (SA) Operation*. Technical Specification (TS) 38.821. Version 16.6.0 (2023-12). 3rd Generation Partnership Project (3GPP), 2023. URL: `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2438`.

[55] Andrey I. Veyler. "5G NR random Access Procedure Performance Evaluation". In: *2022 IEEE 23rd International Conference of Young Professionals in Electron Devices and Materials (EDM)*. 2022, pp. 271–274. DOI: `10.1109/EDM55285.2022.9855128`.

[56] Xiaotao Wang, Jianxin Jia, and Kai Kang. "An Adaptive Discontinuous Reception Mechanism for 5G Unlicensed Band". In: *2021 IEEE/CIC International Conference on Communications in China (ICCC)*. 2021, pp. 1143–1148. DOI: 10.1109/ICCC52777.2021.9580399.

[57] Volker Braun, Karol Schober, and Esa Tiirola. "5G NR Physical Downlink Control Channel: Design, Performance and Enhancements". In: *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. 2019, pp. 1–6. DOI: 10.1109/WCNC.2019.8885990.

[58] Keysight Technologies. *LTE Technology Overview*. Accessed: 2024-08-18. URL: https://www.keysight.com/us/en/lib/resources/technical-specifications/lte-technology-overview-1803101.html.

[59] Irian Leyva-Pupo et al. "Dynamic Scheduling and Optimal Reconfiguration of UPF Placement in 5G Networks". In: *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 103–111. ISBN: 9781450381178. DOI: 10.1145/3416010.3423221. URL: https://doi.org/10.1145/3416010.3423221.

[60] 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*. 2020, pp. 1–6. DOI: 10.1109/GLOBECOM42002.2020.9322125.

[61] TELCOMA. *5G NR Grant free dynamic Scheduling - Transmission without Grant*. 2024. URL: https://telcomaglobal.com/p/5g-nr-grant-free-dynamic-scheduling (visited on 08/18/2024).

[62] Yungang Pan et al. "Multi-Traffic Resource Optimization for Real-Time Applications with 5G Configured Grant Scheduling". In: *ACM Trans. Embed. Comput. Syst.* 23.4 (2024). ISSN: 1539-9087. DOI: 10.1145/3664621. URL: https://doi.org/10.1145/3664621.

[63] Qifeng Sun and Hongqiang Li. "Research and application of a UDP-based reliable data transfer protocol in wireless data transmission". In: *2011 International Conference on Computer Science and Service System (CSSS)*. 2011, pp. 1514–1516. DOI: 10.1109/CSSS.2011.5972154.

[64] M. Miller and Shu Lin. "The Analysis of Some Selective-Repeat ARQ Schemes with Finite Receiver Buffer". In: *IEEE Transactions on Communications* 29.9 (1981), pp. 1307–1315. DOI: 10.1109/TCOM.1981.1095141.

[65] Gautham Prasad et al. "Enhanced Hybrid Automatic Repeat Request Scheduling for Non-Terrestrial IoT Networks". In: *IEEE Open Journal of the Communications Society* 4 (2023), pp. 2802–2814. URL: https://api.semanticscholar.org/CorpusID:258676144.

[66] Michael N. Dazhi et al. "Terminal-Aware Multi-Connectivity Scheduler for Uplink Multi-Layer Non-Terrestrial Networks". In: *2022 IEEE Globecom Workshops (GC Wkshps)*. 2022, pp. 1133–1139. DOI: 10.1109/GCWkshps56602.2022.10008521.

[67] Vikalp Mandawaria et al. "Uplink zone-based scheduling for LEO satellite based Non-Terrestrial Networks". In: *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. 2022, pp. 1313–1318. DOI: 10.1109/WCNC51071.2022.9771894.

[68] Zubair Arnjad et al. "Latency Reduction for Narrowband LTE with Semi-Persistent Scheduling". In: *2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*. 2018, pp. 196–198. DOI: 10.1109/IDAACS-SWS.2018.8525713.

[69] Fenghui Zhang, Liqing Shan, and Xinsheng Zhao. "Mathematical Representation for Reliability of Sensing-Based Semi-Persistent Scheduling in LTE-V2X". In: *IEEE Transactions on Vehicular Technology* 71.9 (2022), pp. 10115–10119. DOI: 10.1109/TVT.2022.3176005.

[70] Malik Muhammad Saad et al. "Enhanced Semi-persistent scheduling (e-SPS) for Aperiodic Traffic in NR-V2X". In: *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. 2022, pp. 171–175. DOI: 10.1109/ICAIIC54071.2022.9722686.

[71] StarLink. *ns-3 Network Simulator*. URL: https://www.nsnam.org.

[72] Mattia Sandri. *An ns-3 Module for Non-Terrestrial Network (NTN) Simulation: Implementation, Design and Performance Evaluation*. Master's thesis. https://hdl.handle.net/20.500.12608/40304. 2022.

[73] Marco Giordani et al. *SATNEX 3.5 Midterm Report*. Tech. rep. ESA, 2024.

[74] 3GPP. *ETSI TS 138 321 V15.3.0 (2018-09): 5G; NR; Medium Access Control (MAC) protocol specification (3GPP TS 38.321 version 15.3.0 Release 15)*. Tech. rep. Sept. 2018. URL: https://www.etsi.org/deliver/etsi_ts/138300_138399/138321/15.03.00_60/ts_138321v150300p.pdf.

[75] 3GPP. *ETSI TS 138 811 V15.3.0 (2019-12): 5G; Non-Terrestrial Network (NTN); Rural scenario propagation model (3GPP TS 38.811 version 15.3.0 Release 15)*. Tech. rep. Dec. 2019. URL: https://www.etsi.org/deliver/etsi_ts/138800_139099/138811/15.03.00_60/ts_138811v150300p.pdf.

[76] Hoang D. Le et al. "On the Throughput Performance of TCP Cubic in Millimeter-Wave Cellular Networks". In: *IEEE Access* 7 (2019), pp. 178618–178630. DOI: 10.1109/ACCESS.2019.2959134.

[77] Yi Zheng et al. "A red-based discard strategy for unacknowledged mode RLC in LTE". In: *2013 IEEE Global Communications Conference (GLOBECOM)*. 2013, pp. 4753–4758. DOI: 10.1109/GLOCOMW.2013.6855702.

[78] Jani Puttonen and Fedor Chernogorov. "The Effect of Discontinuous Reception and RRC Release Timer Parameterization on Mobility". In: *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*. 2014, pp. 1–5. DOI: 10.1109/VTCSpring.2014.7023119.

[79] Neal Cardwell, Stefan Savage, and Thomas E. Anderson. "Modeling TCP latency". In: *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)* 3 (2000), 1742–1751 vol.3. URL: https://api.semanticscholar.org/CorpusID:6581992.