Cape Peninsula
University of Technology

Design, Implementation and Deployment of TestBed for LoRa Communication Architecture on Nanosatellite Constellations

by

REZAH ROWE

Thesis submitted in fulfilment of the requirements for the degree

Master of Technology in Electrical Engineering

in the Faculty of Engineering & the Built Environment

at the Cape Peninsula University of Technology

Supervisor: Dr Balyan Vipin

Bellville
October 2023

DECLARATION

I, Rezah Rowe, declare that the contents of this thesis represent my own unaided work, and that the thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

February 2024

Signed **Date**

ABSTRACT

Long distance RF communication in excess of 20km can be challenging for ground-based systems due to limits of Line-of-Sight (LoS). This can potentially be mitigated using satellite based, Long Range (LoRa) networks. With Low Power Wide Area Networks (LPWAN) finding more popular use paired with the broadening efficacy and connectivity of Internet of Things (IoT), the technology is creating opportunities for use in a wider array of applications.

The Africa Space Innovation Centre (ASIC) Labs, specializing in Maritime Domain Awareness (MDA) telemetry from proprietary nanosatellites, sought to determine the feasibility of a LoRa telecommunication-based nanosatellite constellation. The telecommunication would be established using proprietary commercial off-the-shelf (COTS) hardware and Software Defined Radio (SDR) architecture, paired with a terrestrial ground station.

The measurement of feasibility is determined through simulation results of a technology demonstrator or testbed platform. The results are studied in the form of performance measurements, that correlate to reliability of the established connection of the LoRa network infrastructure and the Quality of Service (QoS) of the satellite telemetry.

With results proving favourable, within the constraints of technology and architecture employed by ASIC, the possible use cases for the platform relative to State of The Art (SoTA) are to be determined. Viable prototype(s) could be further tested, extending the available intellectual property and market access of ASIC Labs.

KEYWORDS | *LPWAN, LoRaWAN, LoRa Networks, Sigfox, Symphony Link, LoRa testbed, Software Defined Radio, nanosatellite constellations*

ACKNOWLEDGEMENTS

I would like to thank my Supervisor Dr Vipin, for his continued patience in my prolonged pursuance of this thesis. To Professor Robert van Zyl for the opportunities and faith vested in me. To my family for their support throughout my studies and for all my acquaintances along the way who encouraged me to persist, since before the inception of my studies, till this very day.

The financial assistance of the National Research Foundation towards this research is acknowledged and appreciated. Opinions expressed in this thesis and the conclusions arrived at, are those of the author, and are not necessarily to be attributed to the National Research Foundation.

TABLE OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

NOMENCLATURE

Term	Definition
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
ASIC	Africa Space Innovation Centre
F'SATI	French South African Institute of Technology
SADC	Southern African Development Community
COTS	Commercial Off The Shelf
SDR	Software Defined Radio
QoS	Quality of Service
LPWAN	Lower Power Wide Area Network
SoTA	State of The Art
SMAD	Space Mission Analysis and Design
LEO	Low Earth Orbit
IoT	Internet of Things
BER	Bit Error Rate
CSS	Chirp Spread Spectrum - Encodes information using wideband linear frequency modulated chirp pulses
ISM	Industrial, Scientific and Medical bands of the RF Spectrum
EM Waves	Electromagnetic Waves
OSI	Open Systems Interconnection
PHY	Physical Layer
LoRaMAC	Long Range Media Access Control - Controls the hardware that interacts with the transmission medium (whether wired, wireless or optical)
DPSK	Differential Phase Shift Keying
FSK	Frequency Shift Keying
SRD	Short Range Devices
FOTA	Firmware-over-the-Air
ADR	Adaptive Data Rates
NB-IoT	Narrowband Internet of Things
LTE-M and LTE-Cat M	Abbreviation for LTE Cat-M1 - Category M1
ZigBee	An IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios
GSMA	Groupe Speciale Mobile Association
M2M	Machine to Machine
SMART	Self-Monitoring Analysis and Reporting Technology
ALOHA	Additive Links On-line Hawaii Area
PER	Packet Error Rate
LNS	LoRaWAN Network Server
EUD	End User Device > LTE Term for a device/ "thing" connected to the network
SF	Spreading Factor - In LoRaWAN this refers to chirp rate

Full-Duplex	Point-to-point system composed of two or more connected parties or devices that can communicate with one another in both directions
FCC	Federal Communications Commission
DtS	Direct-to-Satellite
LR-FHSS	Long Range Frequency Hopping Spread Spectrum
SIC	Successive Interference Cancellation
MIMO	Multiple-Input Multiple-Output
D2D	Direct-to-Device
SALSA	Scheduling Algorithm for LoRa to LEO Satellites
MAC	Medium Access Control
TDMA	Time Division Multiple Access
FAIR Policy	A scheduling policy proposed for the uplink transmissions of ED’s in a satellite LoRaWAN network
TDD	Time Division Duplex
P2P	Point-to-Point
SLINK	The S-Band RF Designation by the IEEE for part of the EM Spectrum that covers frequencies from 2 to 4 GHz
SNC	Spacecraft Network Controller
CCSDS	Consultative Committee for Space Data Systems
CR’s	Coding Rates
GNU	Gnu’s not Unix > A free open-source operating system based on Unix.
MATLAB	Matric Laboratory > An industry and university popular program that operates on whole matrices and arrays.
AWGN	Additive White Gaussian Noise
SER	Symbol Error Rate
ECEF	Earth-Centered Earth-Fixed
FSPL	Free Space Path Loss
PDR	Packet Delivery Ratio
GUI	Graphic User Interface

CHAPTER 1: INTRODUCTION

1.1 General background

LoRa was founded in France in 2009 by two friends; Nicolas Sornin and Olivier Seller, with the addition of their third partner, François Sforza in 2010. The aim was to develop long range, low powered, wireless communication technology, initially targeting the metering industry for gas, water and electricity. It was acquired by Semtech in 2012, with its fundamental technologies patented by 2013, in collaboration with the three founders, to improve the LoRa technology and finalize the hardware to be used in the gateways and end devices. Semtech further endeavoured to accelerate the use of IoT solutions by providing products and services that promote LoRa and the LoRaWAN protocol (Slats, 2020).

1.2 Motivation

LoRa networks are traditionally terrestrial based, with implementation finding popularity in rural areas for various purposes. A particularly popular context, is farmers in the monitoring of crops (Nowatzki & Emeritus, 2021). The dilemma with a terrestrial LoRa network however, is that they require Line of Sight (LoS) up to 10km (Jain, n.d.). This can amount to a pre-requisite of expensive initial infrastructure, since many farmlands can spread over vast distances with hills in-between and non-horizontal terrain, impeding network quality and efficiency.

With the growing popularity in terrestrial LoRa networks (Pinelo, et al., 2023), an attractive solution would be a LoRa based nanosatellite constellation with SDR based payloads, paired with terrestrial LoRa networks, overcoming this impairment. Fundamentally improving the versatility of these terrestrial based LoRa networks.

The organisation ASIC Labs of F'SATI, at the Cape Peninsula University of Technology (CPUT) Bellville satellite Campus (ASIC Labs F'SATI, n.d.), that develops and deploys CubeSats into Low Earth Orbit (LEO), participating in its capacity with the maritime domain awareness of South Africa and The Southern African Development Community (SADC) (SAMSA, n.d.), sought to perform a study on the feasibility of LoRa networks used in nanosatellite constellations (Parada , et al., 2023).

A concept design of a technology demonstrator mission was to be performed, consisting of a network of COTS IoT ground-based transceivers and SDR based satellite payload to determine the expected performance measurements of reliability and QoS.

1.3 Problem definition

As covered in 1.2, achieving Intersatellite communication with a small constellation of satellites has its challenges and limitations. This is of particular importance when transmitting signals, which require a large amount of power, where the power budget for satellites are limited and need to be adhered to (Ahmad & Warip, 2016). This is even more prevalent with the much smaller scale of nanosatellites (D. Bulanov, 2018).

Though they have an emerging presence and boast low levels of power consumption, LoRa networks are typically employed in terrestrial monitoring systems (Sun, et al., 2022). As such, implementing a LoRa based system on a satellite without simulations, working solutions, or measured results, is far too costly and could pose too great of a risk to expensive proprietary hardware.

In the interests of ASIC Labs of F'SATI, there is not a suitable simulation platform to serve as a foundation for modelling or testing expected readings or the feasibility of the endeavour. Simulations serve as the preferred spearhead before hardware is considered, purchased or assembled. Academically; a platform for calculating, assimilating and assessing of simulated signals and their influence by phenomena in the harsh environment of space is desirable.

1.4 Aim

The primary aim of this work and body of research, is the design, implementation and deployment of a testbed platform. tailored to test the feasibility of a LoRa based LPWAN telecommunications architecture, for a constellation of nanosatellite constellations. The testbed platform should be capable of performing simulations with pre-set or unknown conditions, providing performance of reliability and QoS measurements between a network of terrestrial transceivers and SDR based IoT payload. Thereafter, it should be adaptable for other possible uses, since LoRa and LPWAN's are applicable to many industries, environments and applications.

1.5 Objectives

The objectives of this research are closely aligned with the overarching aim. The central goal is the design, execution, and deployment of a specialized testbed platform tailored for assessing the feasibility of a LoRa-based Low LPWAN telecommunications architecture.

This platform is to be specifically designed to evaluate the potential of such a system for a constellation of nanosatellites paired with a terrestrial ground station. This will enable the testing of feasibility of the LoRa-based LPWAN telecommunications architecture.

The simulation of a LoRa-based nanosatellite constellation on the testbed platform resonates with the broader aim of deploying a platform capable of simulating various conditions expected in the harsh environment of space. This simulation capability is crucial for assessing the performance of reliability and QoS measurements, as mentioned in the broader aim.

Producing simulated performance measurements from the testbed indicative of constellation QoS and reliability directly relates to the aim of evaluating the performance metrics between a network of terrestrial transceivers and SDR based IoT payload.

Furthermore, determining relevant state-of-the-art use cases for the testbed platform aligns with the broader goal of making the testbed adaptable for various purposes, as highlighted in the primary aim. The adaptability is emphasized by stating that the platform should be applicable to diverse industries, environments, and applications.

Therefore, this provides a structured and chronological approach to achieving the comprehensive aim and deliverables of the research. This collectively contributes to the design, implementation, and deployment of a versatile testbed platform for assessing the feasibility and performance of a LoRa-based LPWAN telecommunications architecture for nanosatellite constellations.

After rigorous testing, iteration, and thorough review, the TestBed can be positioned as a robust platform with broad applications in the satellite industry. It serves as a versatile tool for testing diverse models within the satellite domain and extends its utility to other industries interested in implementing LoRa-based concepts for various purposes. This adaptability positions the TestBed as a valuable resource, fostering innovation and technological advancements across different sectors beyond satellite telecommunications.

Thus, the outlined objectives for this research can be listed as:

1. Design and build a testbed platform for LoRa telecommunication architecture
2. Simulation of LoRa based nanosatellite constellation on testbed platform
3. Produce simulated performance measurements from testbed, indicative of constellation QoS and reliability
4. Determine relevant SoTA use cases for the testbed platform

1.6 Background and Theoretical Framework

The foundation for the academic theory pertinent to this research is derived from the specialized dual Master’s degree undertaken by the researcher.

The first segment of this degree focuses on specific subjects critical to this study. Signal Theory and Signal Processing contribute essential concepts in telecommunications, while Scientific Computing provides computational tools necessary for data analysis. The inclusion of Satellite Applications offers insights into practical satellite use cases, and the exploration of Orbital Mechanics is pivotal for understanding celestial trajectories.

Satellite Subsystems contribute to the comprehension of satellite components and functionalities. Furthermore, the rigorous study of Space Mission Analysis and Design (SMAD) equips the researcher with a comprehensive understanding of planning and executing space missions.

The second facet of the degree involves the creation of a written dissertation, serving as the core document for this research endeavour.

1.7 Research Methodology

1.7.1 Research and understand LoRa

This initial step involves a comprehensive exploration of LoRa technology, aiming to establish a robust foundation for subsequent phases. Understanding the intricacies of LoRa is crucial for making informed decisions during the simulation and testing phases.

1.7.2 Determine protocol best suited for simulation

The selection of an appropriate protocol for simulating a satellite constellation requires careful consideration. This step involves evaluating various protocols to ensure alignment with the unique characteristics and challenges posed by satellite communication. The choice made here directly impacts the accuracy and relevance of subsequent tests.

1.7.3 Assimilate testbed platform

The integration of a testbed platform is a pivotal aspect of the methodology. This step involves discussing the rationale behind selecting a specific programming environment to design and build the testbed platform, considering factors such as flexibility, scalability, and compatibility with the chosen simulation protocol. The effectiveness of the testbed platform is critical for obtaining reliable and meaningful results.

1.7.4 Perform initial tests and simulations

Running preliminary tests and simulations serves as a practical application of the theoretical groundwork laid in the previous steps. This phase allows for the validation of the chosen simulation protocol and the assessment of the testbed platform's functionality. The insights gained during these initial tests inform the subsequent phases of the research.

1.7.5 Observe and record Performance Measurements relating to reliability and QoS

The systematic observation and recording of performance measurements are essential for evaluating the reliability and QoS of the simulated nanosatellite constellation. Discussing the metrics chosen for assessment and their relevance ensures transparency in the evaluation process.

1.7.6 Analyse and deduce on results and observations

This step involves a thorough analysis of the gathered data. This analysis highlights the methodologies employed for data analysis, emphasizing the validity and reliability of the deductions drawn from the results. The analytical process serves as the basis for informed conclusions.

1.7.7 Compare testbed results to SoTA platforms

Comparing the testbed results to SoTA platforms is essential for benchmarking and validating the research outcomes. Discussing the criteria for comparison and addressing any deviations observed, helps establish the significance and contribution of the research to the field of LoRa networks deployed in nanosatellite constellations.

1.7.8 Suggest improvements and changes

Identifying areas for improvement and suggesting changes is a reflective phase that addresses how the findings from the initial tests and comparisons inform recommendations for refining the simulation, enhancing the testbed platform, or adjusting methodologies for future studies.

1.7.9 Identify and suggest alternative use cases for testbed

Exploring alternative purposes and use cases for the testbed broadens the impact of the research. Discussing the criteria used to identify these alternatives and their potential implications fosters a comprehensive understanding of the testbed's versatility and applicability beyond the initial scope of the research.

1.8 Research Delineation

This study focuses primarily on the meticulous design and implementation of a LoRa communication TestBed. The primary objective is to create a specialized architecture tailored for simulations that measure the feasibility and QoS of nanosatellite constellations. Beyond the technical specifications, several critical aspects delineate the scope and objectives of this research.

1.8.1 Simulation Environment as a Prototype

The simulation environment serves as a prototype, emphasizing a controlled setting for evaluating the effectiveness of the LoRa communication testbed. This deliberate choice allows for methodical testing without deploying proprietary satellite hardware in the initial phases.

1.8.2 Exclusion of Tests with Proprietary Satellite Hardware

The decision to exclude tests with proprietary satellite hardware stems from the need for further assessment, improvements, approvals, and iterations by qualified engineers. The prototype, at this stage, is prioritized for functionality and performance evaluation, paving the way for subsequent phases involving real-world satellite hardware.

1.8.3 Functional Model as the Primary Focus

Emphasis is placed on developing a functional model of the LoRa communication testbed as the foremost objective. This choice is rooted in the need to establish a solid foundation before progressing to more intricate phases. A functional model lays the groundwork for subsequent enhancements and iterations.

1.8.4 Technical Specifications

While the study is deeply rooted in technical specifications, it extends beyond solely software and programming. This encompasses considerations related to feasibility assessments, QoS measurements, and the iterative development of a simulation environment. Moreover, it touches upon the necessity for collaboration with qualified engineers and the importance of approvals before integrating proprietary satellite hardware.

1.8.5 Iterative Development Process

The delineation acknowledges the iterative nature of the development process, recognizing that the initial prototype is a stepping stone, anticipating continuous refinement based on assessments, feedback, and collaborative efforts. This iterative approach aligns with the dynamic and evolving nature of satellite communication technologies.

1.9 Thesis Overview

- **Chapter 1 – Introduction**

  This Chapter provides an overview of the work pursued in this dissertation.

- **Chapter 2 – LoRaWAN literature review**

  This Chapter elaborates on the understanding of LoRaWAN, approached via a manner of queries, with its relevance and implementation in satellites, as compared to terrestrial use.

- **Chapter 3 – Satellite Communication**

  This chapter delves into the understanding of satellite communication paired with terrestrial based telecommunication systems as well as intersatellite communication. Forming comparisons of traditional methods with LoRaWAN, including the significance and difficulties in implementing the latter.

- **Chapter 4 – Comparison of thesis**

  This chapter compares recent studies on LoRaWAN as used in satellites as LoRa Gateways, reviewing their implementation and Testing in Real Environments.

- **Chapter 5 – Simulation and Experimentation**

  This Chapter encompasses the testbed/ simulation environment and functionalities.

- **Chapter 6 – Functions and Descriptions**

  This chapter gives an overview of the different respective functions that encompass the simulation.

- **Chapter 7 – *Results of Performance Measurements***

  Visual results from the testbed simulations as charts and graphs of recorded results produced by a single run of the TestBed platform for an SNR of -60 to 60 dB.

- **Chapter 8 - Discussion and Comparison of Performance Measurements**

  Findings from the efficacy of the results relating to the desired performance measurements from the testbed

- **Chapter 9 – Conclusion(s)**

  Summarized findings and conclusion on the study.

CHAPTER 2: BACKGROUND STUDY

2.1 Overview

The feasibility of using LEO satellites to collect messages from IoT devices in rural areas using LoRa modulation proves challenging. Impairments include Doppler shift, multipath fading, and interference.

In the research performed by (Colavolpe, et al., 2019), a redesign is proposed of the LoRa receiver that includes a modified synchronization scheme, a new channel estimation algorithm, and a joint detection and decoding algorithm. The performance of the proposed system through simulations is evaluated, showing that it can achieve a low bit error rate (BER) for a range of signal-to-noise ratios.

Slightly different directions are pursued in (Chenhui & Qingjia, 2020), based on the Semtech STM32F103 and SX1278 modules of LoRa. The LoRa module is used to achieve a self-organizing star-type network design. The monitoring data can be transmitted to the receiving terminal through the BeiDou satellite, which provides data support for analysis and decision-making.

A trade-off study by (Fernandez, et al., 2020) presents different LPWAN technologies for use in CubeSat platforms, focussed on LoRa. The authors compare the most relevant LPWAN technologies, analyse the link budget for different LoRa configurations, and detail the results from ionospheric scintillation tests. The paper further discusses the challenges of IoT devices requiring low-power connectivity to transmit and receive information, and how this affects the communications range and data rate.

These are among but the numerous resources scourged. With the wider spread implementation of IoT, there are various competing LPWAN protocols and methods employed in terrestrial LoRa communication; each with significant differences and similarities. Detailed comparisons are performed to determine the most suitable protocol before assimilation of the LoRa simulation environment. The protocol selected directly affects the feasibility of the implementation.

2.2 LoRaWAN Literature review

A literature review on LPWANs, LoRa, LoRaWAN, SigFox and Symphony Link, what they are and how they relate to investigation and experiments in the research undertaken.

2.2.1 LPWAN

Low-Power Wide Area Networks, more commonly referred to as LPWANs, refers to a method of interconnectivity of battery-powered devices, enabling them to communicate over longer distances than cellular networks at low bandwidths and low bit rates. These LPWAN's can be adapted into existing infrastructure and established using a variety of different protocols, being either licenced or unlicensed. The most notable of which are elaborated on further. (Shea, 2017)

2.2.2 LoRa

LoRa is an abbreviation for *Long Range* but also refers to the proprietary physical, however wireless LoRa modulation technology, which was developed by Semtech. LoRa modulation is used to encode information onto radio waves, using what is known as chirp pulses (The Things Network, n.d.), a derivation of Chirp Spread Spectrum (CSS) modulation, which is employed in wireless long distance LPWAN. LoRa differs from; standard cellular telecommunication, Wi-Fi and Bluetooth in that it uses a low data rate over long distances and unlicensed ISM frequency bands that are globally available and modulates the data into EM Waves using CSS transmission (Bloechl, 2022). The primary function is for uplink-only applications (i.e: Data from sensors and end devices to gateways)

In terms of the Open Systems Interconnection (OSI) model, which is comprised of various layers, LoRa belongs to what is known as the Physical Layer (PHY). (Devopedia, n.d.). Where LoRaWAN aligns with the second and third layers, known as the Datalink and Network layers respectively (The Things Network, n.d.), illustrated in Figure 1: OSI Model Reference Guide .

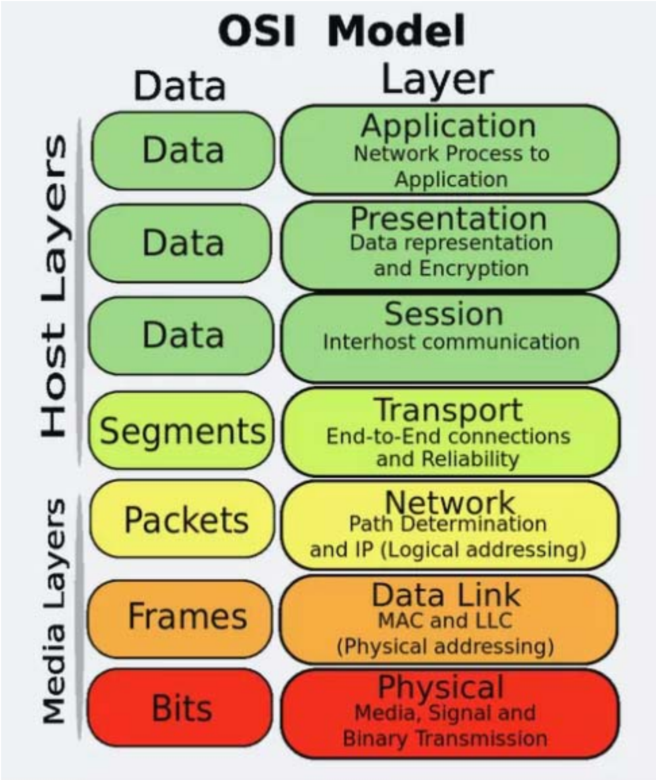

Figure 1: OSI Model Reference Guide (Mitchell, 2019)

2.2.3 LoRaWAN

Though LoRaWAN is an acronym that distinguishes a Long-Range Wireless Area Network, it is simultaneously the name of LoRaWAN, being at the helm of LoRa, is a popular open-source communication protocol used by both private and public networks, providing a greater range than cellular networks (Semtech, n.d.).

LoRa and LoRaWAN are both part of the open licensed-free spectrum. The LoRa physical layer enables the long-range communications link and LoRaWAN defines the communication protocol as well as the system architecture for the network. Importantly, a single radio can be used for both the receiver at the endpoint as well as the base station.

LoRaWAN (initially named LoRaMAC) is built on top of LoRa, as it belongs to the proprietary MAC (Media Access Control) layer protocol that specifies the message formats and security layers required for a networking protocol. It is also the most adopted standard of LPWAN (Slats, 2020).

LoRaWAN became a recognized standard in 2015 when Semtech (an IoT systems and Cloud connectivity service provider) founded an association called the LoRa Alliance. The purpose of this alliance is the standardization of LPWAN globally as well as promoting the adoption and interoperability of the LoRaWAN standard and related technologies (Semtech, n.d.).

The relation between LoRa, LoRa Modulation, LoRaWAN and Semtech are shown in Figure 2: LoRaWAN technology stack.

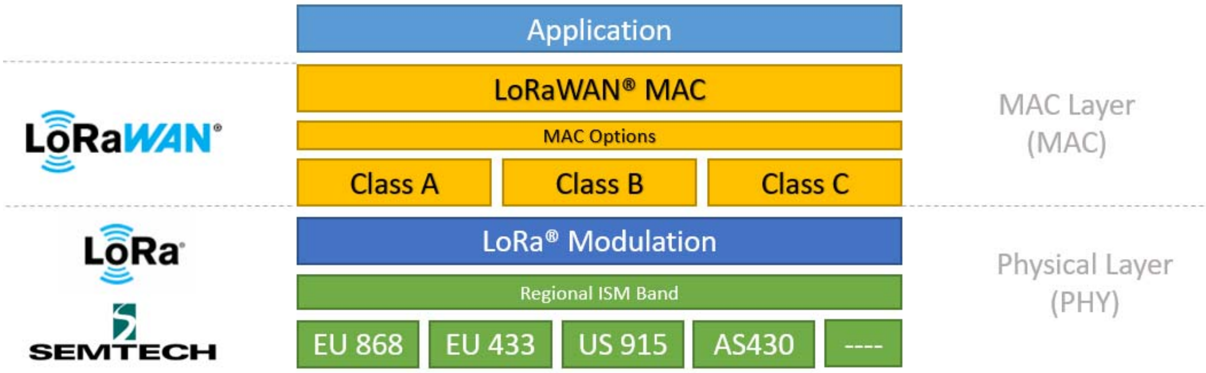

Figure 2: LoRaWAN technology stack (Semtech, n.d.)

2.2.4 SigFox

SigFox is the immediate competitor to LoRaWAN, however, it is widely regarded as a proprietary service (McCleland, 2020). Much like LoRa, Sigfox was founded in 2010, the progenitors of whom were also two French founders who built a global network dedicated to IoT. Based on low power, long range and the transfer of small amounts of data. Sigfox has significantly contributed to the wide-spread adoption of IoT around the world (Sigfox, n.d.). Sigfox reports lower data rates compared to LoRaWAN (between 100 and 600 bps), which depends on the operating region (Sigfox, 2018). Sigfox uses DPSK for Uplinks and FSK for Downlinks, where Uplink signals typically encounter more interference than downlink signals (DZone, n.d.).

2.2.5 Symphony Link

Symphony Link, another compelling alternative to LoRaWAN and SigFox, is a network designed to overcome certain limitations present in LoRaWAN. Symphony Link is a standardized protocol developed by Link Labs. It is built on LoRa CSS, using a proprietary built MAC Layer on top of the compulsory Semtech chip called Symphony Link (McCleland, 2020).

2.2.6 Differences between the protocols

2.2.6.1 SigFox and LoRaWAN

While Sigfox and LoRaWAN were founded around the same period of time in France, they serve similar purpose but also differ in implementation.

SigFox is credited with starting the LPWAN movement, where its business model relies on royalties from network operator resales (Ray, 2015). SigFox differs from LoRa in the methods employed to send data, as well as the quantity, speed and duration of the data sent (Michalski, 2017). Although the LoRa Alliance have ensured an open ecosystem and standardization of the LoRa protocol, it possesses a closed element, being; the Semtech Silicon. (Link Labs, 2018)

These two protocols differ primarily in the quantity, speed and duration of the data sent (DZone, n.d.). The key difference however, is that; LoRaWAN has bi-directional communication and is an open standard, operating over the 169 MHz, 433 MHz and 915 MHz frequencies (for respective global regions) (Valerio, 2020) where Sigfox operates in the unlicensed sub-gigahertz; ISM and SRD bands (between 862 and 928 MHz) of the radio spectrum (Brian, 2018).

Moreover, Sigfox filed for insolvency due to financial difficulties due to challenges in the industry during Covid-19, but further clarified that there was a pause on payments to creditors amidst this turmoil (DZone, n.d.)

2.2.6.2 LoRaWAN and Symphony Link

LoRa is region based to some extent, being widely referred to as “LoRaWAN” across Europe. Symphony Link on the other hand, is based mostly in the United States and Canada. Symphony Link differs slightly from LoRaWAN, in that it was designed to overcome limitations present in LoRaWAN systems (Ray, 2018). However, it boasts less popularity for widespread use, as compared to LoRaWAN.

2.2.7 LoRa Limitations

LoRa technology lacks essential features such as downlink communication, firmware updates over-the-air (FOTA), adaptive data rates (ADR), QoS, multicast support, and comprehensive security measures.

2.2.8 Symphony Link Improvements

Symphony Link addresses these limitations by enabling bidirectional communication, supporting FOTA updates, implementing ADR, offering QoS capabilities, facilitating multicast, and bolstering security through encryption, authentication, and integrity protection.

2.2.9 Advancements Enabled by Symphony Link

The adoption of Symphony Link brings significant advancements; point-to-multipoint network protocol using the Semtech LoRa scheme, longer battery life, better latency and a smaller packet size. It also includes two-way communication, remote device updates, dynamic data rate adjustments, data packet prioritization, multicast transmission, and robust security. These improvements extend the potential applications of wireless technology, notably in global IoT connectivity, remote sensing, and space exploration (RF Wireless World, n.d.).

2.2.10 What about other wireless sensor technologies?

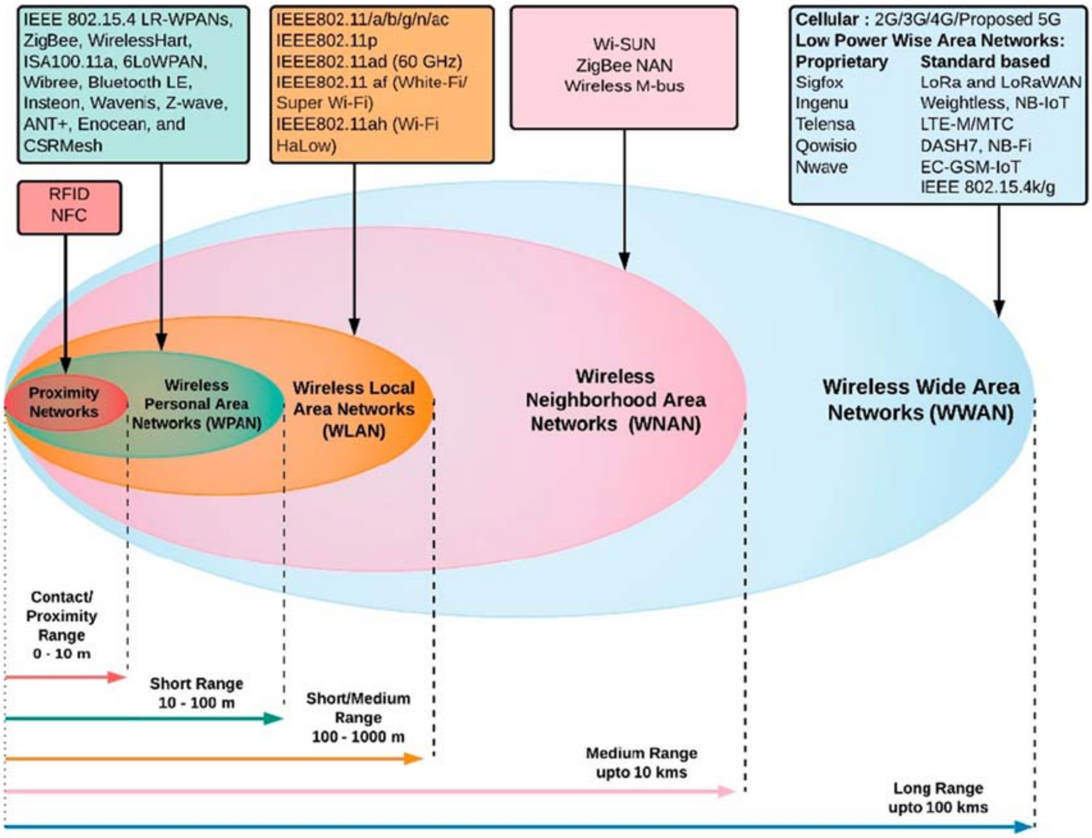

Figure 3: Wireless access geographic coverage (Chaudhari, et al., 2020)

In terms of alternatives, a few other technologies used for LoRa networks are; NB-IoT (which can be seen as the legacy cellular's response to LPWAN and Sigfox), LTE-M an IP based alternative to NB-IoT, LTE-Cat M, Dash7 Alliance Protocol, albeit each with their advantages, pitfalls and tailored applications (McClelland, 2016). There's likely also IP licensing to pay, where the license fees for the cellular spectrum of NB-IoT are very high (Semtech, 2022).

Traditional wireless sensor technologies like Bluetooth, ZigBee and Wi-Fi can also be adapted for long range networks, but this would require significant and costly modification to infrastructure, as they're not designed for a wide area of coverage. A comparison of the area of coverage of wireless technologies can be observed in (Chaudhari, et al., 2020).

2.2.11 Why are LPWAN's appealing?

Compared to traditional long distance wireless networks such as GSMA (Groupe Speciale Mobile Association), LPWAN's exhibit low power usage whilst providing wide area coverage.

Since LPWANs are more tailored towards Machine-to-Machine (M2M) communication and IoT devices, wireless battery powered, end devices are optimized to remain in a low power or "deep sleep" mode whenever they're not in use (Daniłowski, 2021). This extends the life span of end devices to around 10 years using a single coin cell battery, which significantly reduces the need for short term maintenance, where necessary firmware updates can also be remotely updated (The Things Network, n.d.). Also referred to as "Over-the-Air" firmware updates touted by Semtech, but not offered by all LPWAN protocols, Symphony Link claims to be the only LPWWA system that offers this feature (Link Labs, n.d.) & (Link Labs, 2017).

2.2.12 Where are LPWANs used?

Some of the most popular applications include 'SMART' (Self-Monitoring Analysis and Reporting Technology) Agriculture, Buildings, Cities, Electricity Metering, Environment, Home Healthcare, Retail, Water and other various types of metering (Semtech, n.d.). Other known applications are in waste management, parking sensors, lighting and air quality management, where industrial applications can range from radiation and leak detection, item location, shipping and transportation tracking (Rajiv, 2018). Through the aid of IoT prowess, LPWAN is a growing technology that will continue to impact a wider number of fields.

2.2.13 Latest trends in LPWAN

Globally, the deployment of LPWAN networks is on the rise, finding application in diverse sectors including but not limited to; smart cities, industrial IoT, agriculture, and asset tracking, reflecting their versatility.

Forecasts indicate a significant growth trajectory for the LPWAN market in the coming years, largely propelled by the expanding use of IoT devices and applications. Efforts towards standardization are underway to ensure seamless interoperability and compatibility among different LPWAN technologies, promoting their widespread adoption.

LPWAN networks present cost-effective solutions for connecting numerous devices across extensive distances while keeping power consumption at a minimum. These emerging trends underscore a burgeoning interest and uptake of LPWAN technologies across industries and applications, signalling promising prospects for further advancements and innovations in this domain (Water Online, 2024).

2.2.14 Choosing an LPWAN protocol

LPWAN Protocols are not interchangeable and narrowing down a protocol depends on the requirements of the application. The choice of protocol paired with the structure of the simulation environment marks the foundation of determining a feasible network.

Although Sigfox exhibits the most traction in the LPWAN space, thanks in no small part to its marketing campaigns, its downlink capability is severely limited and the signal is quite susceptible to noise (McClelland, 2016). Thus, it is better suited to applications where small messages need to be sent and received.

Compared to Sigfox, LoRaWAN is an open-source communication architecture that; allows for easier access of use, as well as variable data rates, some degree of bi-directionality (asynchronous), less likelihood of message interference, higher gateway capacity, user defined packet size among others (McClelland, 2016). LoRaWAN demonstrates better link budgets than some of its competitors. However, connecting to LoRaWAN outside of Europe might require the personal deployment of a gateway (Ray, 2018).

The LoRaWAN protocol is ALOHA (Additive Links On-line Hawaii Area) based, which means that messages are transmitted without confirmation of receipt at the end nodes. This leads to Packet Error Rates (PER) that exceed 50%. These error rates may be dismissible for meter reading applications, however a PER of 0% is a primary requirement for industrial and enterprise related applications. This makes LoRaWAN more suited to uplink-focussed networks (Ray, 2018).

Although LoRaWAN is more appealing than Sigfox, Symphony Link has notable improvements to LoRaWAN. Some of these include but are not limited to; the use of acknowledgement (Ack) (RF Wireless World, n.d.) or guaranteed message receipt which reduces the error rate, a more secure link, QoS, adaptive data rate, firmware updates over-the-air and flexible frequency band which means no duty cycle limit(1% limit for LoRaWAN) (Ray, 2018) & (Mishra, 2018).

2.3 Verdict/ Choice of protocol

2.3.1 Conclusion between Symphony Link, LoRaWAN and SigFox

Though Symphony Link addresses and overcomes shortfalls of the LoRa protocol, whose backbone it was built on; Symphony Link is more widely adopted in industry than public (Queralta, et al., 2019). Additionally, it is more regionally based and doesn’t share as popular of a presence as its competitors, Sigfox and LoRaWAN.

2.3.2 LoRaWAN vs. Symphony Link and Sigfox

LoRaWAN stands out as a superior protocol for several key reasons. Firstly, it is an open standard, backed by the LoRa Alliance with over 500 global members, promoting interoperability and widespread adoption. In contrast, Symphony Link and Sigfox are proprietary solutions controlled by specific vendors, potentially leading to compatibility issues.

LoRaWAN boasts a higher data rate than Sigfox, supporting up to 50 kbps in Europe and 27 kbps in the United States, compared to Sigfox's meagre 100 bps. LoRaWAN also offers a larger payload size, accommodating up to 243 bytes per packet in Europe and 222 bytes in the US, while Sigfox's limit is just 12 bytes per packet, restricting application complexity.

LoRaWAN excels in power efficiency, with some devices achieving an astounding 10-year battery life. In contrast, Symphony Link's bidirectional communication and firmware updates consume more power, resulting in a roughly 2-year battery life. Additionally, LoRaWAN offers cost advantages, with device and network costs substantially lower than Symphony Link.

Moreover, LoRaWAN provides broader coverage and greater network capacity, reaching up to 15 km in rural areas and supporting up to one million devices per gateway. Symphony Link and Sigfox have shorter coverage ranges and lower network capacities.

2.3.3 Considerations

While LoRaWAN offers numerous advantages, it's important to note some limitations. LoRaWAN lacks bidirectional communication, firmware updates over-the-air, QoS, multicast, and repeater capability, features offered by Symphony Link. Additionally, LoRaWAN does not guarantee message receipt, remove duty cycle limits, or provide dynamic range, which are advantages found in Symphony Link.

Lastly, LoRaWAN operates in unlicensed bands, potentially susceptible to interference from other users or sources of noise, whereas Sigfox operates in licensed bands, offering more reliable and secure communication.

2.4 LoRa information and specifics

2.4.1 LoRa End User Device Gateway, LNS

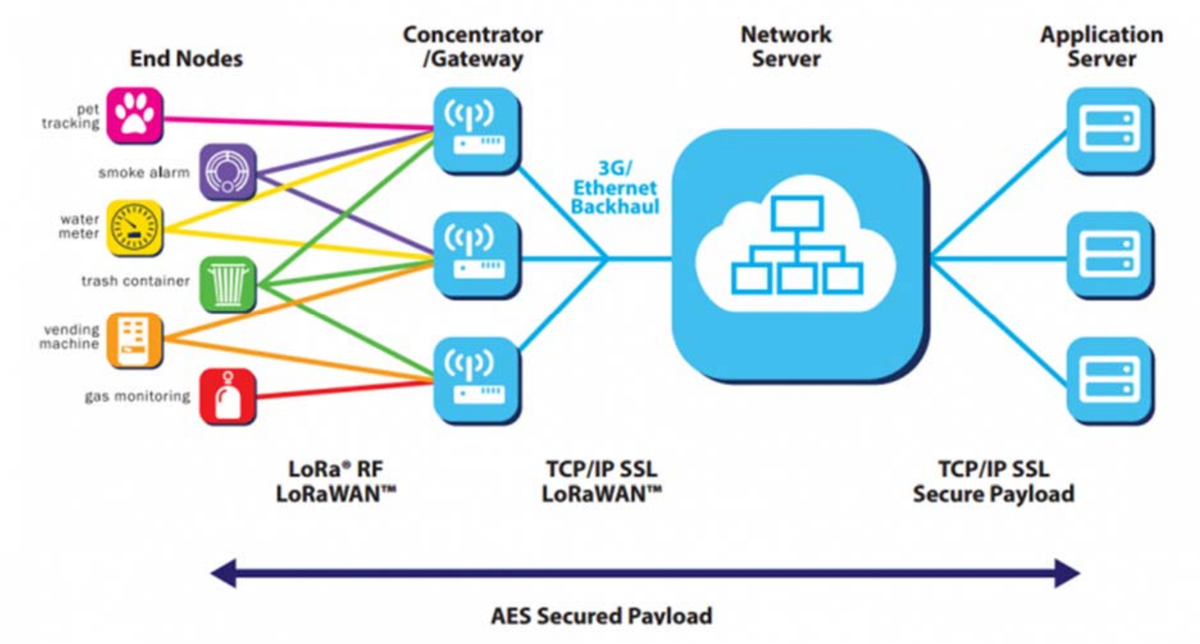

Figure 4: What are LoRaWAN Gateways & Nodes? (Tan, 2021)

As seen in Figure 4 LoRaWAN networks are comprised of; Endpoints or End User Nodes/ Devices, Gateways and a LoRaWAN Network Server (LNS) as the key components of the network.

Endpoints, also known as an End User Device (EUD), is a low-power device, typically; sensors, meters, control actuators or other devices that collect data and sends that data to a nearby gateway. Though there is no fixed association between an endpoint and a specific gateway, the same endpoint can be served by multiple gateways in the area, where each uplink packet sent by the endpoint will be received by all gateways within reach.

Gateways receive LoRa modulated RF messages from any endpoint in hearing distance and forward these data messages to the LNS, which are connected through an IP backbone. The main purpose of gateways are to relay messages from the server to the endpoint. These are essentially collective midpoints for data, with the ability to communicate with many endpoints at the same time, where the number of gateways in a LoRa network determine the capacity of messages that can be sent in a day. Thus, the number of messages a given deployment can support can be increased by the addition of gateways.

Lastly, the gateways transport communications between endpoints and a central/ LNS, which is a crucial component, implemented in a star topology that manages the entire network, including applications. If the LNS receives multiple copies of the same message, it keeps a single copy and discards the others, a process known as message deduplication.

Since multiple gateways can receive the same LoRa RF message from a single end device, the LNS performs data deduplication and deletes all copies, selecting the best gateway for routing downlink messages to the end devices.

The LNS ensures the authenticity of every sensor on the network and the integrity of every message, while also dynamically controlling the network parameters to adapt the system to ever-changing conditions (Jain, n.d.).

2.4.2 Spreading Factor (SF)

LoRa Modulation has six spreading factors (SF), ranging from SF7 to SF12. This is the amount of spreading code applied to the original data signal. The larger the SF the farther the signal can travel without errors at the RF receiver. This means that more distant sensors transmit at a higher SF. A trade-off is made between battery power of an end node and the distance of this node.

Thus, an end device at a further distance from the gateway will need to transmit at a higher SF. This consequently results in a lower data rate as well. Additionally; a larger SF, increases the time on air of the signal, which increases energy consumption, reducing data rate whilst improving communication range (Kim, et al., 2022).

2.4.3 Other notable LoRaWAN information

There is a phenomenon known as “ducting” that helps LoRaWAN signals to travel further distances. Briefly explained, this works by radio propagation via atmospheric ducts that benefit the network, enabling longer range as well as better signal strength (Garg, 2018).

Most LoRaWAN systems can receive eight messages simultaneously. LoRaWAN nodes don’t acknowledge receipt of message, but it can be requested. It should be noted however, that this can lead to inadvertent network collapse since, once a gateway spends time transmitting an acknowledgement, it stops listening for new messages.

As illustrated relative to the other layers of LoRaWAN in Figure 2: LoRaWAN technology stack, LoRaWAN nodes are split into three classes; A, B and C (Bloechl, 2022).

Table 1: LoRaWAN Classes (Semtech, n.d.)

Class A	Class B	Class C
Most energy efficient > Must be supported by all nodes	Uses scheduled receive windows	Always listening to the radio interface > Except when transmitting
Uses pure ALOHA	Determines whether to receive downlink frames or not > Based on beacons sent from gateway	
Can only receive downlink frame after successful uplink transmission	Doesn't require successful uplink transmission	
Intended for battery operated **sensors**	Intended for battery operated **actuators**	Intended for **mains powered actuators**

LoRa uses the concept of FSK but can detect quieter chirps below the noise floor. According to (Saeed, et al., 2020) the modulation and multiple access techniques used in IoT systems cannot be directly used in CubeSats, as a result of the Doppler Effect and propagation-delay constraints.

In terms of directional communication, Semtech unveiled a Full-Duplex gateway that allows for simultaneous, bi-directional communication to and from the gateway and lengthens the downlink window (Camarillo, 2021).

It is important to note that LoRaWAN Networks interfere with each other when there is more than one operating in an area. There could be possible interference when other satellites close by are operating on LoRaWAN networks.

2.5 LoRaWAN Disadvantages

All LoRaWAN gateways are tuned to the same frequencies, thus the LoRaWAN network will see all the traffic. It's thus preferable to have a single network operating in an area. However, specific channels can be set aside for set uses, through cooperation with the LoRa Alliance.

Since gateway transmission for the ISM band in Europe are limited by FCC regulations, this limits the duty cycle on the 868MHz bands to an average of one percent duty cycle (Bloechl, 2022).

CHAPTER 3: SATELLITE COMMUNICATION

This chapter delves into the understanding of satellite communication with terrestrial based systems as well as intersatellite communication. Forming comparisons of traditional methods with LoRaWAN, including the significance and difficulties in implementing the latter.

3.1 Traditional Satellite Communication

Traditional inter-satellite communication uses radio frequencies (RF) to provide links between artificial satellites. This method has limitations such as; interference, bandwidth and power consumption (ITU Council, 2023). Some satellite constellations use intersatellite links to relay data in space without relying on ground stations (Viasat, n.d.), whereas another method uses optical inter-satellite communication via lasers to transmit data at higher speeds and lower power (Yukizane, et al., 2021).

3.2 Direct-to-Satellite (DtS)

Direct-to-Satellite communication is a new approach that enables devices such as smartphones, tablets or IoT sensors to connect directly to satellites without using terrestrial networks. This can provide global coverage, lower latency and higher resilience. Some satellite and telecommunication operators are partnering to offer DtS services for various applications such as emergency communications, rural connectivity and smart agriculture (Suarez & Queiroz, 2022).

3.3 LoRa and satellites

3.3.1 Swarm

Swarm is a company that operates a constellation of 120 satellites that provide global IoT connectivity using custom-designed LoRa transceivers that can communicate with low-power devices on the ground over long distances. Swarm proclaims to offer a low-cost, reliable and easy-to-use service for various IoT applications (Swarm Space, 2022).

3.3.2 Lacuna Space

Lacuna Space is a company that offers LoRaWAN services via its own and partner satellites. These satellites use standard LoRa modules that can connect to any LoRaWAN device on the ground, touting to provide a seamless integration of satellite and terrestrial networks for IoT applications (Lacuna Space, n.d.).

3.3.3 FossaSat-1

Launched in December 2019, FossaSat-1 became Spain's first pico-satellite, acclaimed as the first satellite equipped with LoRa technology. Its successor FossaSat-1B, that took flight in September 2021 is a second generation of the satellite with some improvements. Both of these satellites are open-source and utilize LoRa modules to facilitate IoT communications (Allan, 2018).

These satellites implement LoRaWAN by using a custom firmware that allows the satellite to receive and transmit LoRa packets from the ground. As detailed on their GitHub repository, FossaSat-1 enabled global IoT communications via LoRa modules, offering students access to satellite communication for under 20 Euros (~R400).

This Arduino-powered satellite, (focused on education and research, emphasizing simplicity) introduced a new radio system to space, enhancing cost-effectiveness. It penetrated boundaries, being the first 1P (5cm cubed) satellite with deployable solar cells and a LoRa transmitter, revolutionizing space accessibility (FOSSASystems, 2020).

3.4 The significance of LoRaWAN and the difficulties of implementation

The use of LoRa to connect IoT devices to gateways across wide distances with DtS connectivity is a service that aims to use the LoRaWAN protocol with Long Range Frequency Hopping Spread Spectrum (LR-FHSS) technology to support direct communication between IoT devices and LEO satellites. However, this implementation also faces its own difficulties.

3.4.1 Interference

LoRaWAN operates in unlicensed bands, which means that it can suffer from interference from other users or sources of noise. Interference is important to take into consideration as it can degrade the signal quality and reduce the data rate and reliability of the communication.

3.4.2 Synchronization

LoRaWAN requires precise synchronization between the transmitter and the receiver to decode the signals correctly. However, synchronization can be challenging in satellite communications due to the high speed and variable distance of the satellites, which cause Doppler effects and time delays. These synchronization errors can lead to packet loss and increased power consumption.

3.4.3 Collision

LoRaWAN uses a random-access scheme, which means that multiple devices can transmit at the same time without coordination. Collision however, can occur when two or more devices use the same frequency and time slot to transmit their signals, resulting in corrupted packets and reduced network capacity.

3.5 Overcoming difficulties and Improving Performance

Some methods being worked on to overcome difficulties and improve the performance and feasibility of LoRaWAN for DtS connectivity.

3.5.1 Successive Interference Cancellation (SIC)

Successive Interference Cancellation is an algorithm that can improve the collision robustness and network capacity of LoRaWAN for satellite communications, decoding multiple interfering signals by exploiting their different power levels and spreading factors.

This scheme passes information iteratively between equalizer, demodulator, and decoder to enhance multiple-input multiple-output (MIMO) reception in downlink LTE. This is especially prevalent in scenarios where the signal-to-noise ratio (SNR) is low, outperforming conventional hard-decision schemes with perfect per-layer rate control (Xhonneux, et al., 2022).

3.5.2 Long Range Frequency Hopping Spread Spectrum (LR-FHSS)

LR-FHSS is a fast FHSS modulation scheme used for uplink only, to spread the signal over a wide bandwidth. This affords LR-FHSS the ability to increase the network capacity and interference resilience of LoRaWAN for satellite communications while maintaining the same radio link budget as LoRa. It is a new physical layer designed for extremely long distances and large-scale communication scenarios, importantly satellite IoT (Boquet, et al., 2020).

3.5.3 D2D-aided transmission

D2D-aided transmission is a technique that uses device-to-device (D2D) communication, aiding LR-FHSS LoRaWAN protocol in direct-to-satellite to relay signals from IoT devices to satellites. D2D-aided transmission can enhance the coverage and reliability of LoRaWAN for satellite communications.

Consideration of a practical ground-to-satellite fading model is shown to improve the network capacity, for which a closed-form outage probability expression is derived. In (Maleki, et al., 2022), the analytical expressions are validated through computer simulations, capturing the effects of noise, fading, unslotted ALOHA-based time scheduling, the receiver's capture effect, IoT device distributions, and distance from node to satellite.

CHAPTER 4: COMPARISON OF THESES

This chapter compares recent studies on LoRaWAN as used in satellites as Gateways, reviewing their implementation and Testing in Real Environments

4.1 LoRaWAN LR-FHSS for Direct-to-Satellite

The study in (Mikhaylov & Alves, 2022) introduces the analysis and simulation of DtS with the use of LR-FHSS based data transmission to LEO satellites. Designed for wide-ranging frequency signal reception and transmission, this scheme transmits packets across multiple channels, minimizing interference and improving reliability. LEO satellites equipped with non-terrestrial LoRaWAN gateways, receive packets from ground nodes and forward them to the satellite's payload, further increasing network capacity and collision robustness. This enables direct connectivity between machine devices and LEO satellites as well as simultaneous communication with multiple ground nodes.

This modulation scheme for DtS machine-type communication for remote areas with the accompanying long-range and low-power capabilities supports large-scale networks and offers high interference robustness, thanks to increased spectral efficiency, header replication, and reduced coding.

The study develops analytical and simulation models for LR-FHSS packet delivery from ground nodes to LEO satellites, providing numerical results to assess the feasibility of large-scale networks. Use of a simulator is employed to model node distributions, traffic patterns, and time-frequency resource allocation, where these models generate insights into packet losses for two data rates defined in the EU region.

4.2 Sparse Satellite Constellation Design for DtS IoT Services

This study proposes a different approach to designing sparse satellite constellations for DtS IoT services. (Capez, et al., 2022) introduces a mathematical model to optimize the number and placement of satellites in the constellation, while taking into account various constraints such as the maximum gap time between passing-by satellites and the minimum elevation angle for communication with IoT devices.

The proposed approach is evaluated using simulations and is compared with existing methods, showing significant improvements in terms of reducing the number of in-orbit satellites while maintaining efficient communication with resource-constrained IoT devices. Overall, this study illustrates a promising solution for designing cost-effective and scalable satellite constellations for DtS IoT services.

4.3 Scheduling Algorithm for LoRa to LEO Satellites

A Scheduling Algorithm for LoRa to LEO Satellites (also known as SALSA) is designed to schedule uplink transmissions from end devices to a LoRaWAN gateway installed on a LEO satellite. Making use of ALOHA which is used to transmit data over a public network channel. It functions within the OSI model's Medium Access Control (MAC) sublayer.

It cancels collisions from random ALOHA MAC sublayer and packet drops due to intermittent link availability. SALSA uses a Time Division Multiple Access (TDMA) approach to divide the satellite's orbit into time-slots, as well as a first come, first serve policy to schedule transmissions within each time-slot.

To ensure that all end devices have an equal chance to transmit, SALSA also proposes what is referred to as the FAIR policy, which allocates transmission slots based on the total time needed for scheduling at least one uplink transmission from all the End Devices within every single slide (Afhamisis & Palatella, 2016).

4.4 Narrow-Band Inter-satellite Network for IoT

This paper evaluates potential multiple access techniques for an adaptive network architecture and potentially applicable multiple access techniques. Though not explicitly elaborated upon however, it mentions that a Time Division Duplex (TDD) scheme with session-oriented point-to-point (P2P) protocols in the data link layer is considered more suitable for limited resources.

A TDD scheme is a method of communication where the transmission and reception of data occurs on the same frequency but at different time intervals, which allows for the efficient utilization of the available bandwidth. In the context of the adaptive network architecture, implementing a TDD scheme helps optimize the use of limited resources.

Session-oriented protocols imply that there is a logical session established between the communicating devices, ensuring reliable and efficient data transfer. The study primarily focuses on the design methodology and implementation of the adaptive network architecture with limited resources for nanosatellites. This involves the development of a software-defined S-Band radio (SLINK). Notably, the SLINK incorporates the physical and data link layer functionalities.

The communication layer between the SLINK radio and SNC (Spacecraft Network Controller) is defined according to the CCSDS (Consultative Committee for Space Data Systems) recommendation and the network layer is implemented on the spacecraft side (Yoon, et al., 2019).

4.5 Assessing LoRa for Satellite-to-Earth Communications

This paper analyses the link budget for different configurations of the LoRa system, taking into account the impact of ionospheric scintillation on communication range and data rate, a phenomenon of which, can cause signal fading and degradation in satellite communications.

Appropriate tests are conducted to assess the performance of LoRa in satellite communications, comparing it with other LPWAN technologies. Some of the potential proposed solutions to address the issues of the phenomenon, provide valuable insights into the key challenges and limitations that need to be addressed in order to fully realize its potential.

Tests performed, involved the use of an SDR based test setup to evaluate the performance of different LoRa device configurations under different levels of ionospheric scintillation. The tests measured the received power and packet delivery ratio as a function of the intensity scintillation index, and the results showed the robustness of the LoRa modulation in these new environments.

The impact of having redundancy in the LoRa system was also tested by comparing the performance of two different Coding Rates (CRs). The results showed that the throughput decreases as ionospheric scintillation severity increases, and that having more redundancy is only positive in a scenario where the communications are in the limit of the link budget, where the scintillation has low or medium severity (Fernandez, et al., 2020).

4.6 Low-Earth-Orbit satellite communications using LoRa-like signals

This scholarly paper, which was part of a PhD thesis, is a convincing look into the realm of LoRa CSS modulation; serving as the focal point of satellite communications in LEO using LoRa-like signals.

The first contribution proposes several synchronization algorithms that can accurately decode LoRa-like signals received with random arrival times and significant Doppler effects, especially the Doppler time-variation. The Doppler effect in this context is the change in frequency of a signal due to the relative motion of the transmitter and the receiver, where the Doppler time-variation is the change in the Doppler effect over time; caused by the high speed of LEO satellites.

The synchronization algorithms are based on different techniques, such as cross-correlation, matched filtering, frequency estimation, and phase estimation. Where the performance of the algorithms is evaluated in terms of bit error rate (BER), packet error rate (PER), and computational complexity.

The second contribution proposes a novel approach to decode interfering LoRa-like signals in uplink and downlink contexts, based on the successive interference cancellation (SIC) algorithm. Where SIC is a technique that allows the recovery of multiple signals that are transmitted simultaneously on the same channel by iteratively decoding and subtracting the strongest signal from the received signal until all signals are decoded or no more signals can be decoded.

This study introduces innovative methods for decoding interfering LoRa-like signals in both uplink and downlink scenarios, drawing upon the SIC algorithm. It also meticulously explores the influence of synchronization and Doppler effects on LoRa-like signals while presenting viable solutions to mitigate these effects.

Through a combination of simulations and practical experiments, rigorous assessment of the proposed methodologies is explored. The results demonstrate the potential to enhance the reliability and capacity of LoRa-like satellite communications.

Overall, this research underscores the suitability of LoRa-like signals for LEO satellite communications, advocating for global IoT connectivity and advancements in space exploration (Temim & A., 2022).

CHAPTER 5: SIMULATION AND EXPERIMENTATION

This Chapter encompasses the testbed/ simulation environment and functionalities

5.1 Programming languages

The programming languages commonly used with IoT are C, C++ and Python or a combination of these languages in some instances. Some languages may be more suitable than others, which will be determined. The ns-3 simulator also has a module to generate Python bindings from C/ C++ code, which provisions for a great amount of interoperability (nsnam, n.d.).

5.2 Simulating ground based IoT network with SDR based IoT satellite payload

Some useful resources have implemented fully functional GNU SDR's of LoRa transceivers, serving as a good foundation for the satellite payload portion of the LoRa constellation (Tapparel, et al., 2020) & (Open Source Libs, n.d.). For the ground based IoT transceivers, inspiration can be drawn from the vast amounts of LoRa projects and open-source GitHub repositories along with suitable platforms.

5.3 Key functionalities of the TestBed

The TestBed platform should preferably meet the capabilities set out in the objectives, producing accurate simulations with results and QoS measurements of a LoRa network using ground based IoT transceivers with nanosatellite constellations. Simulation results should preferably indicate information and telemetry including but not limited to; PER, data transfer rate, successful data transfer percentage, connection up-time, distance between satellite and ground-based transceivers.

5.4 What can be identified as unknown conditions?

The environment of space is harsh and numerous conditions can cause satellite failure. However, the research can be refined to LoRa network conditions and the absence of connection with ground nodes during single or multiple passes of a satellite due to:

- ❖ Weather conditions preventing connection with satellite
- ❖ Faulty end nodes or transceiver(s)
- ❖ Ground nodes occupied and not receiving transmissions
  - ➢ Possibly for maintenance or testing
- ❖ PER's in excess of 90%

5.5 Acceptable tolerance of result accuracy?

With the nature and expense of satellites, the highest plausible degree of accuracy is always the aim, however for this prototype, an operational model is desired. To prove as a functional proof of concept, the platform will require further implementation and testing with real satellite hardware, the implementation of which, is outside the scope of this research. A truly accurate simulation would require further extensive variables and testing to be taken into account.

5.6 Compare results to actual in field measurements?

As tried and tested as results may appear, in-field implementation will differ to some degree due to various factors that are outside the control of the researcher. Further to this, perfecting any endeavour into the field of space requires vast amounts of consideration for the numerous conditions and factors encountered which falls outside of the scope of this study. However, the prototype platform will build the foundation to warrant hardware implementation and experimentation.

5.7 Types of Software

Aside from the open source LoRa modules, simulators and environments, the testbed platform to be assimilated is the main software involved in this body of work. Accompanying development software like MATLAB has been consulted where the programming language C is applicable and used. The advantage of MATLAB is it's extensive and continued updates, packages and support. MATLAB is well trusted and implemented by many industries, with various visualisations and reporting. It's the perfect environment for continued iterations and improvements to the TestBed model.

5.8 Design Requirements

Although simulations will never be 100% true to life; the design needs to take environmental effects into consideration if accurate simulation is to be performed. Some notable effects to be modelled include; Bit, Symbol and Packet Error Rates as well as SNR, where comparable to real world values are strived for.

Both satellite and ground station parameters along with simple signal propagation delay encompass the main components of the simulation. Where the simulation output/ results should be numerical, aided with visualization.

5.9 Stages of analysis

Understanding the stages of analysis involves a systematic approach that aligns with the primary objectives and their corresponding sub-goals.

5.9.1 Stage One

Initiating the analysis journey, the first stage entails a thorough assimilation of the LoRa simulation environment, ensuring its compatibility and functionality alignment with the intended purpose. This phase is characterized by meticulous testing procedures aimed at evaluating the environment's capability to execute simulations accurately and with the desired level of precision.

5.9.2 Stage Two

Transitioning into the second stage, the focus shifts towards the generation and documentation of precise performance metrics within the network. This phase serves as a pivotal milestone within the framework of this study, as it lays the foundation for insightful observations and conclusions.

5.9.3 Stage Three

Concluding the analysis journey, the final stage is dedicated to envisioning future enhancements and exploring innovative applications for the simulation environment. This phase goes beyond the immediate scope of the study, delving into the SoTA use cases and potential avenues for further development and refinement.

CHAPTER 6: FUNCTIONS AND DESCRIPTIONS

This chapter gives an overview of the different respective parameters and functions that encompass the simulation. The MATLAB code for all of these functions can be found in the Appendices; from Appendix A to Appendix I, with a sample of the simulation report in Appendix J.

There are many repositories that can be used as a baseline to work from, including offerings from the likes of The Things Network. The chosen repository to work from is based on the MATLAB file exchange repository LoRaMatlab by (Al-Hourani, 2021), where variables for this simulation are passed between various functions for this simulation, as detailed below.

6.1 BERLoRa.m

This function forms the vast majority of the simulation, where all the other functions are centred around it. Its main purpose is the simulation of LoRa communication system performance, with up and down-chirp signals, mixing them with Rayleigh and Additive White Gaussian Noise (AWGN) noise for coherent and non-coherent detection.

The main parameters that it evaluates are the BER, Symbol Error Rate (SER), and PER. Over various a predefined ranged of SNR values.

The Signal and Noise (AWGN and Rayleigh fading channels) are simulated by generating additive white Gaussian noise (`*n*`) and channel coefficients (`*h*`). For the signal reception and processing, the received signals (`*Rx1*` and `*Rx2*`) are generated by adding this simulated noise to the transmitted signals.

The simulated theoretical BER, SER, and PER are derived for both non-coherent and coherent detection scenarios by comparing the simulation results and the original transmitted symbols. For non-coherent detection (`*ynCoh1*` and `*ynCoh2*`), Fast Fourier Transforms (FFT) is used to estimate the received symbols. Whereas for coherent detection (`*yCoh1*` and `*yCoh2*`), this demodulation occurs by convolving the received signals with phase-shifted versions of the transmitted chirp signals.

The simulated theoretical results for the BER, SER, and PER are them plotted and compared for different error rate metrics and SNR values. Additionally, certain variables are stored in a transmitted_data_struct() and passed to other functions for the rest of the simulation.

6.2 QoS

The addition of QoS performance metrics has been embedded in the *BERLoRa.m* function. The addition to this function by the researcher are some valuable QoS metrics; visualized with graphs and recorded into different report formats (.txt, .doc and .csv). The parameters calculated and visualized in this script are; Packet Delay Ratio, Average Packet Delay, Average Throughput, Energy Efficiency, Spectral Efficiency, Link Reliability and Jitter, though some functions require further perusal.

6.3 loramod.m

This function forms part of the repository referenced which generates LoRa Chirp modulation signals with spreading factor and bandwidth settings defined in the *BERLoRa.m* function.

It takes an input vector which represents symbols to be modulated, with various parameters defined in the function *[y] = loramod(x, SF, BW, fs, varargin).* These input arguments are; *x* (input symbols), *SF* (spreading factor), *BW* (bandwidth), *fs* (sampling frequency), as well as an optional parameter. Where the number of symbols are calculated based on the spreading factor. Additionally, it includes conditional statements to check the number of input arguments that ensure these arguments are provided, otherwise it raises an error.

LoRa generally uses different SF's to control data rate and range, where the SF chosen and used for the purposes of this research is static. Subsequent code blocks check the validity of input arguments, such as ensuring the input arguments are positive integers within the range of valid symbols. If the optional argument is not provided it defaults to 1; otherwise, it takes the value passed as the optional argument.

The main part of this function involves a loop over the elements of the input symbols. For each element, *gamma* and *lambda* are computed, two symbols (*t1* and *t2*) of the modulated signal are then calculated using LoRa modulation formulas.

Finally, the resulting modulated signals are concatenated into the output variable *y* to generate a modulated signal for LoRa transmission and referenced in the *BERLoRa.m* script.

6.4 satellite_simulation.m

Aptly named, this code calculates and visualizes the trajectory of a satellite orbiting the Earth with appropriate parameters, as elaborated further.

To begin with, some parameters are initiated, namely; *orbit_altitude_km,* which defines the altitude of the satellite above the Earth's surface in kilometres, *orbit_inclination_deg* being the inclination angle of the satellite's orbit in degrees and *orbit_period_min* being the orbital period (for a singular orbit around the Earth) of the satellite in minutes.

Further satellite orbit parameters are declared and calculated with; *earth_radius_km*, *orbit_radius_km*, *orbit_velocity_kmps*, which are; the radius of the Earth in kilometres, the total radius of the satellite's orbit (adding the Earth's radius to the satellite's altitude) and the orbital velocity of the satellite (determined based on the orbit's radius and period) in kilometres per second.

A vector representing time in minutes is created with *time_minutes*, from 0 to the specified *orbit_period_min*, with a step of 1 minute. Separate arrays are initialized to store position data in both *satellite_longitude_deg* as well the satellite's longitude in degrees at each time step in *satellite_latitude_deg*.

It is to be noted that the *earth* is assumed to be spherical for simplicity, with a specified semimajor axis and close to zero eccentricity. However, the *satellite* is initialized with the satellite's semimajor axis, being its orbit radius (in meters) and inclination angle.

The satellite position at each time step is iterated over each time step in *time_minutes*, with the inside of the loop being *satellite.TimeSinceLaunch* set to the present time in seconds, converting minutes to seconds.

The internal *position* function of the iterated loop is used to compute the satellite's position in the Earth-Centered Earth-Fixed (ECEF) coordinate system, returning coordinates in meters. This ECEF position is further converted to geodetic coordinates of latitude and longitude in radians, stored in *satellite_latitude_deg* and *satellite_longitude_deg* after converting to degrees.

After the above calculations, the satellite trajectory is then plotted in two dimensions for visualization, where the x-axis represents longitude and the y-axis represents latitude in degrees.

Majority of the variables called and used in this script have their values defined/ declared in the main.m script to be easily modified, as well as called/ referenced in other scripts for the simulation.

6.5 propagation_delay.m

This function takes three input arguments that calculate and account for the propagation delay between a ground station and a satellite based on their geographical positions and the speed of light (approximately 299,792 km/s).

The *ground_station_position* and *satellite_position* variables are two element vectors representing the latitude and longitude of the ground station and satellite respectively. Where the directional gain of the ground station's antenna is set to 20 dBi (isotropic decibels)

The distance between the ground station and satellite is calculated in kilometres using the *distance* function, which computes the distance between two points on the Earth's surface via the Haversine formula. The propagation delay is then calculated in seconds; dividing the distance by the speed of light. This gives the time it takes for a signal to travel from the ground station to the satellite or vice versa.

Example values for; *ground_station_position*, *satellite_position*, are provided that represent the geographical coordinates of a ground station in Bellville, Cape Town of the Western Cape and a satellite in Durban, along with the speed of light. These values are then finally used to calculate the *propagation_delay*, displaying the results in seconds. Majority of these values are declared and stored in the main.m script, where they can be easily modified and called by other scripts.

Key elements of this function, is that it simulates signal propagation through a communication channel by considering path loss, atmospheric fading, and AWGN based on a specified SNR.

Two particular input arguments: *tx_signal* (transmitted signal) and *SNR_dB* (the signal-to-noise ratio in decibels) are used to first calculate the path loss in dB using a simplified model; based on the ratio of the signal power before and after adding AWGN.
Lastly, a simplified atmospheric fading effect is generated by adding random values.

6.6 FSPL.m

This function calculates the Free Space Path Loss (FSPL) in decibels for the communication link between the satellite and ground station, quantifying the loss of signal power as the LoRa signal propagates through free space. It’s a simple form of FSPL, using the Ground Station location and Satellite Orbit Altitude along with the link frequency for the range of SNR values declared in the BERLoRa.m script. The final result is trimmed to its actual size, excluding any unused slots.

6.7 Doppler_Shift.m

This function calculates the Doppler Shift based on the relation to the observer in relation to the source of the wave, being the ground station and satellite communication link. The script assumes constant velocities for simplicity in metres per second. It extrapolates values both from the ground station and satellite parameters, using the Haversine formula to calculate the great-circle distance between satellite and ground station.

6.8 generate_txt_report.m/ generate_word_report.m

These scripts generate a report detailing readings and values for the most relevant parameters/ performance metrics of the simulation, detailing both inputs and outputs. These values are extracted across all the functions and calculations performed for the simulation.

The output destination of the report can be adjusted beginning of the script, where currently, results will be saved to a folder on the user desktop. For now, it defaults to a folder named "Simulation Results" on the desktop.

6.9 generate_reports_csv.m

This function performs the same function as the *generate_txt_report.m* and *generate_word_report.m*, except that the results are written to an excel spreadsheet, formatted in a much easier manner to view, with the results separated in various columns. The spreadsheet allows for further observations and data analysis to be performed by generating graphs in any csv viewer such as excel.

6.10 main.m

This script is the main function that calls previously defined functions, passing values and variables back and forth, triggering them.to finally output the simulation results. This is the only function that needs to be run and all the other scripts will be run to trigger their respective purposes.

To begin with, it creates a structure to store values for all the scripts related to this simulation such as; *satellite_simulation()* and *ground_station_simulation();* used for simulating the behaviour of the satellites and ground station, respectively.

Calling the *Doppler_Result*, *FSPL_dB*, *signal_propagation* and text reports naturally simulates Doppler Shift, FSPL, signal propagation and fading, considering; path loss, atmospheric effects and channel modelling.

Essentially, this code is the final part of code used to simulate and analyse the performance of a DtS communication system employing LoRaWAN with a terrestrial ground station. It covers aspects such as BER, SER, PER, satellite and ground station characteristics, signal transmission, propagation, QoS performance metrics and data analysis. The specific parameters and simulation details can be customized as needed for a particular scenario.

6.11 Other

Tested although not perfected, various other simulation results are displayed when the *main.m* function is run. These include; Satellite Orbit, Satellite Ground Track, Satellite Visibility Coverage, Satellite Orbit Shape.

CHAPTER 7: RESULTS OF PERFORMANCE MEASUREMENTS

Visual results from the testbed simulations as charts and graphs of recorded results produced by a single run of the TestBed platform for an SNR of -60 to 60 dB. The results are discussed in Chapter 8

7.1 Bit Error Rate vs SNR

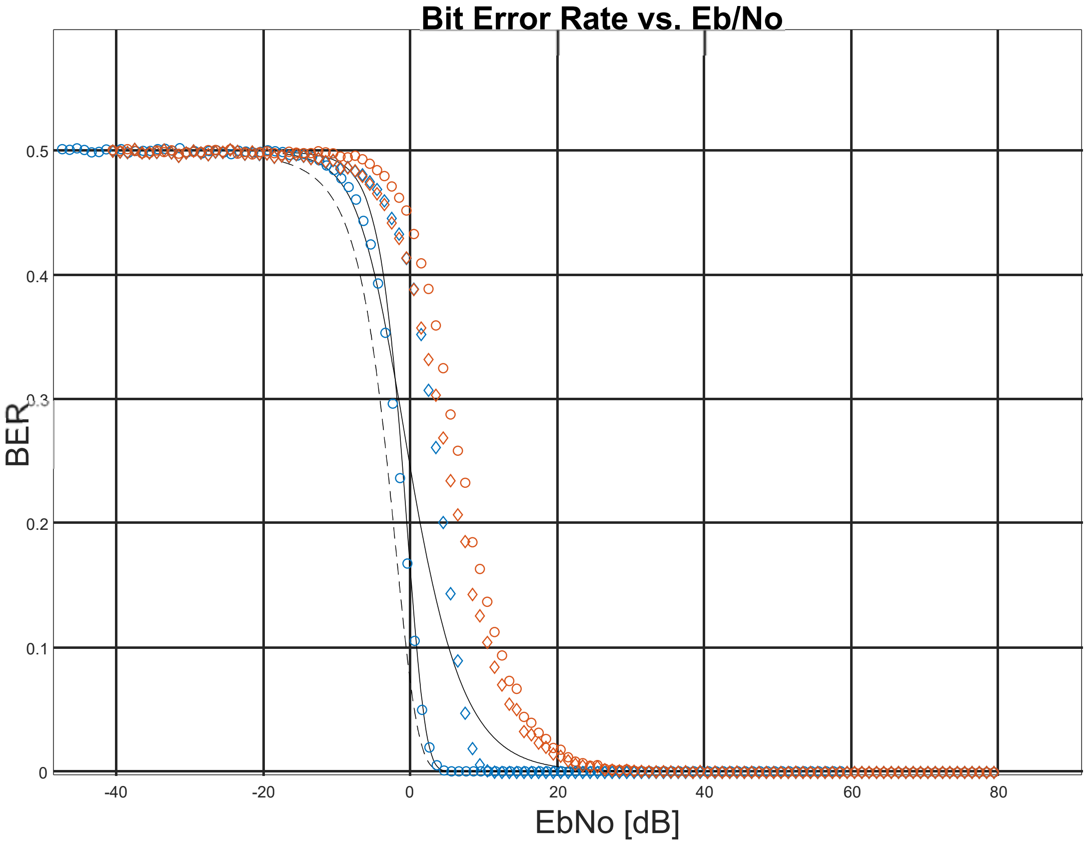

Figure 5: Bit Error Rate vs SNR

7.2 Packet Error Rate vs SNR

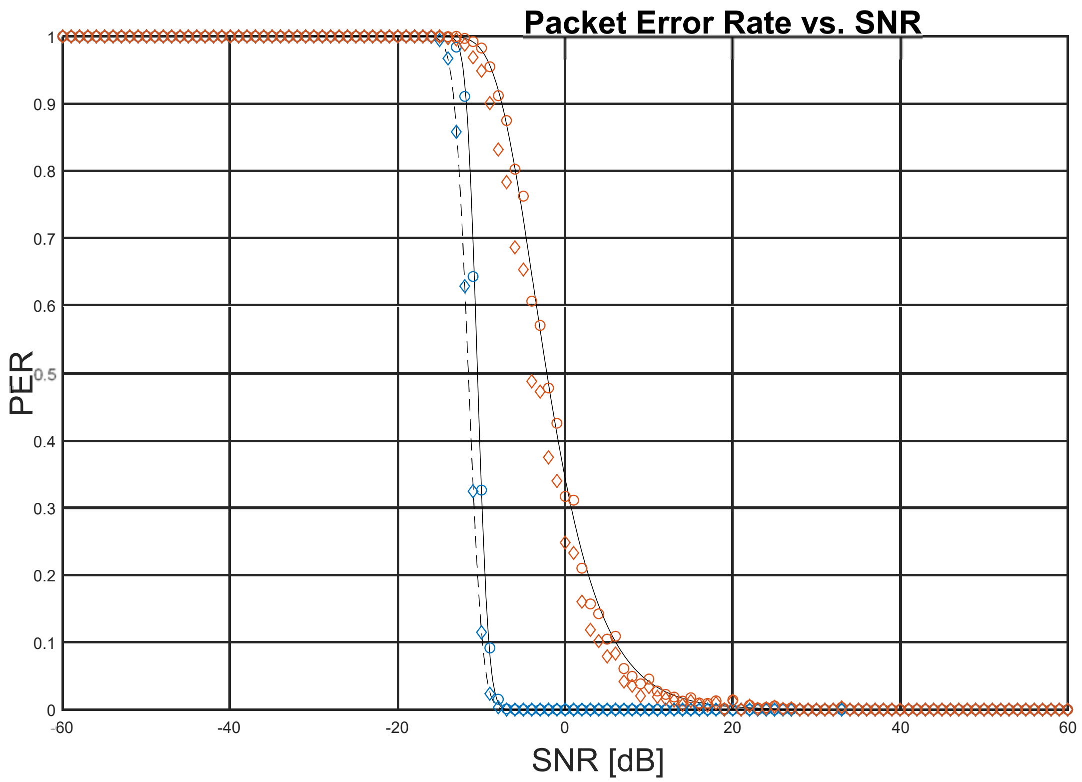

Figure 6: Packet Error Rate vs SNR

7.3 Symbol Error Rate vs SNR

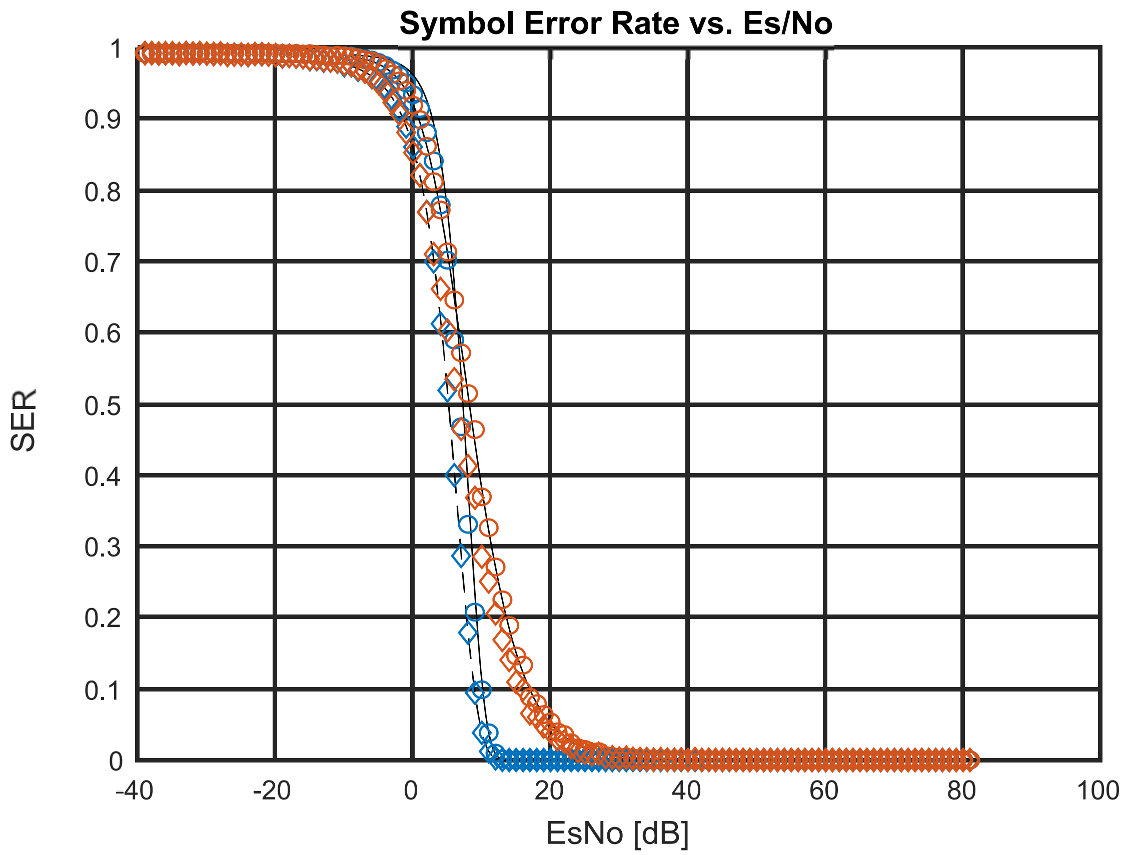

Figure 7: Symbol Error Rate vs SNR

7.4 Packet Delivery Ratio vs SNR

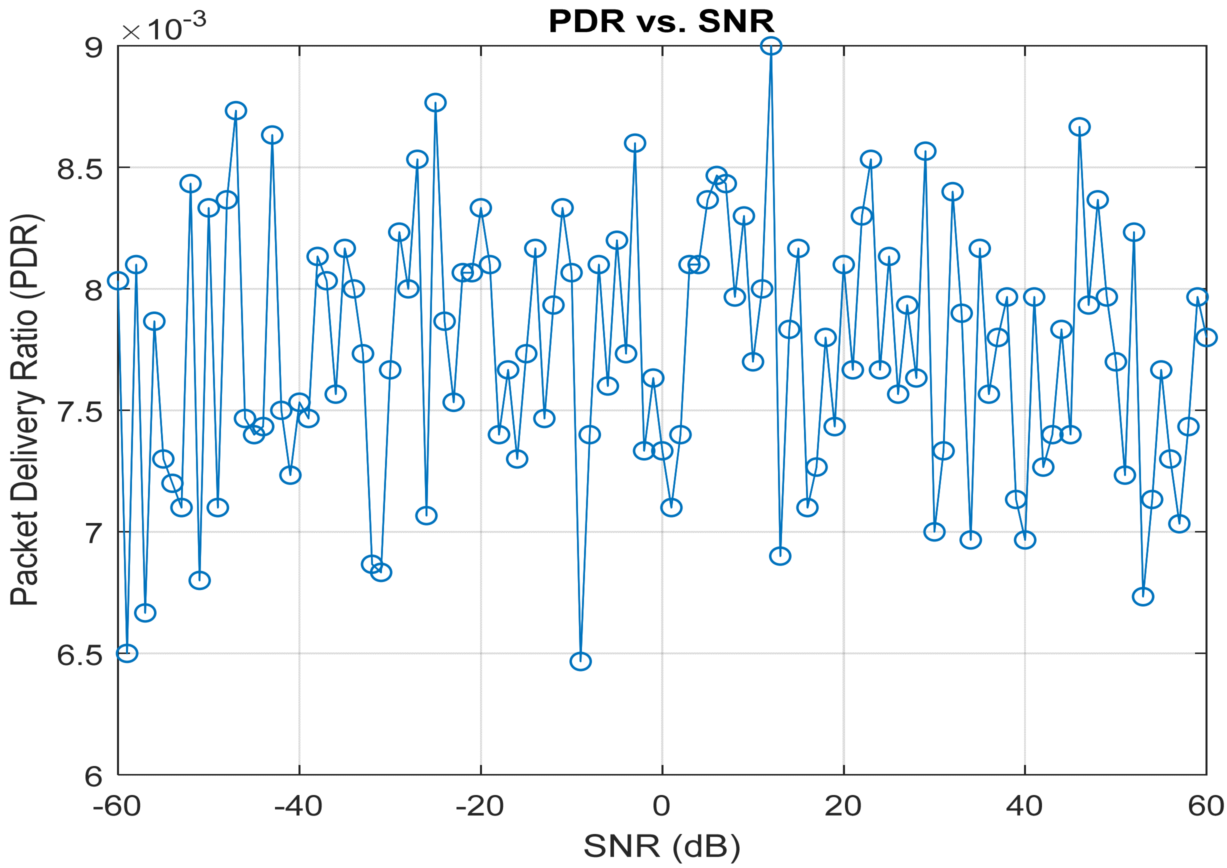

Figure 8: Packet Delivery Ratio vs SNR

7.5 Average Throughput vs SNR

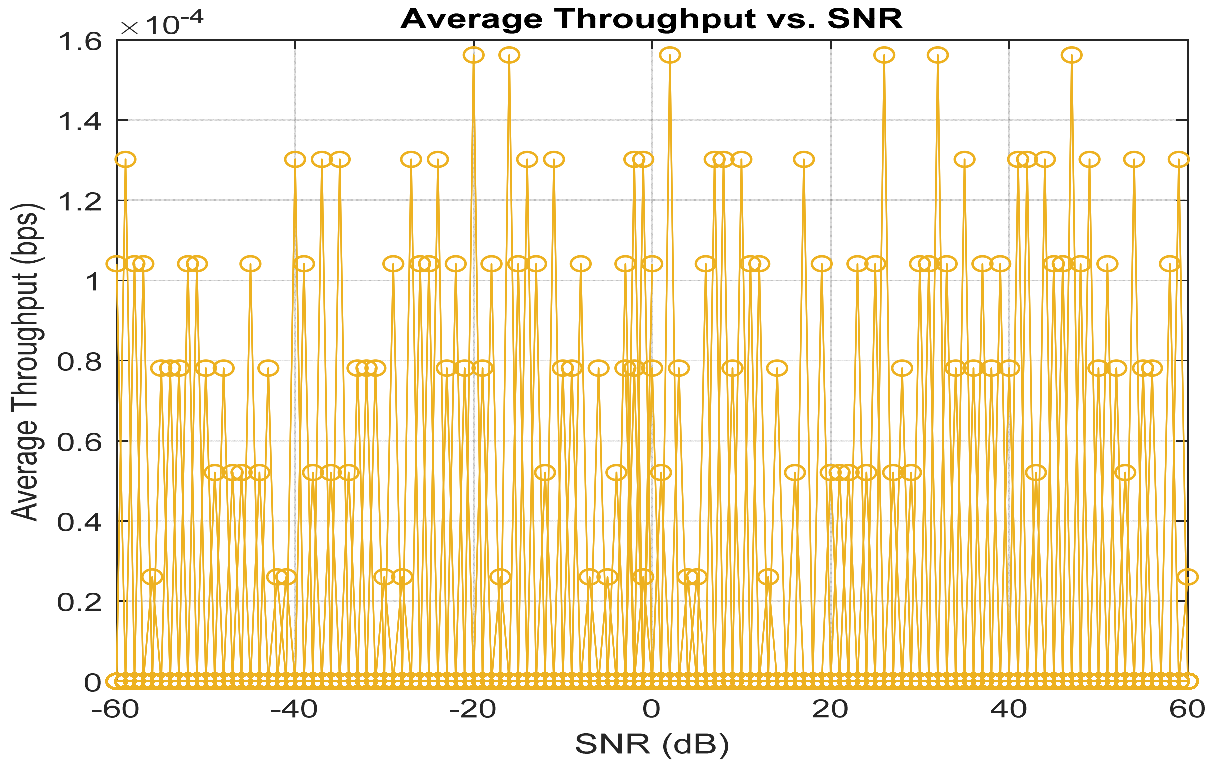

Figure 9: Average Throughput vs SNR

7.6 Energy Efficiency vs SNR

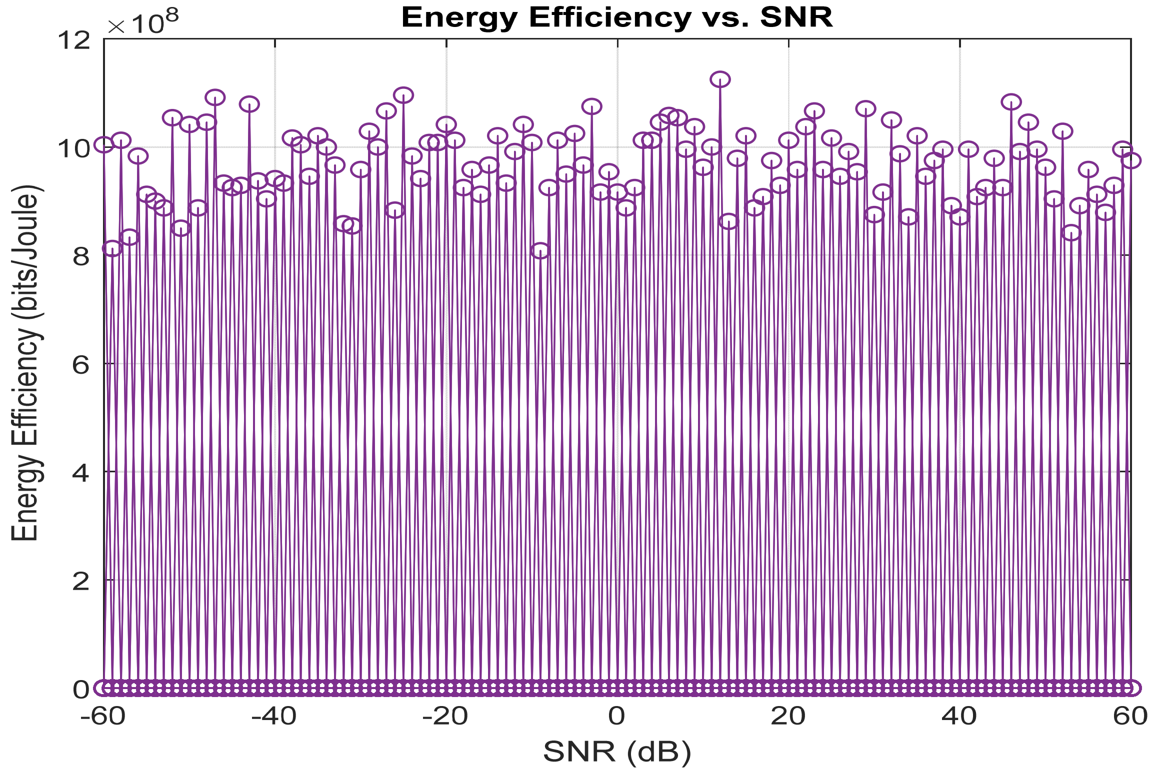

Figure 10: Energy Efficiency vs SNR

7.7 Spectral Efficiency vs SNR

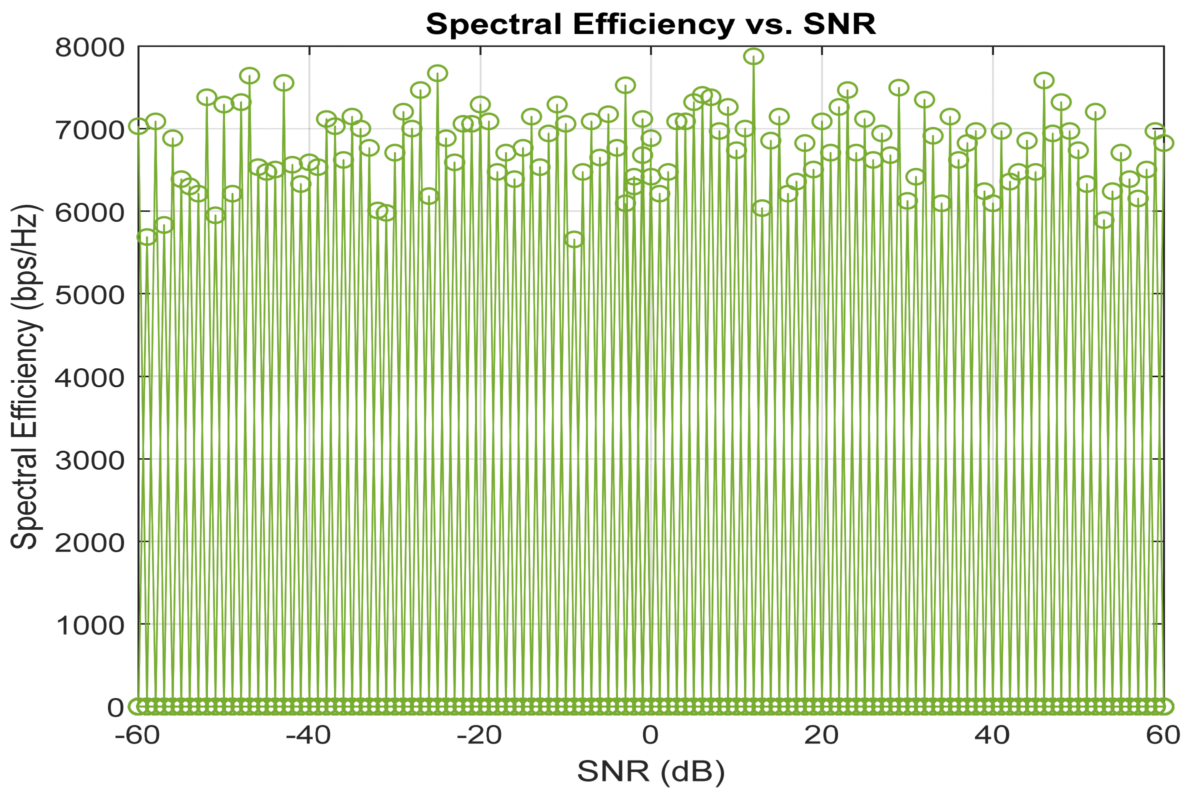

Figure 11: Spectral Efficiency vs SNR

7.8 Link Reliability vs SNR

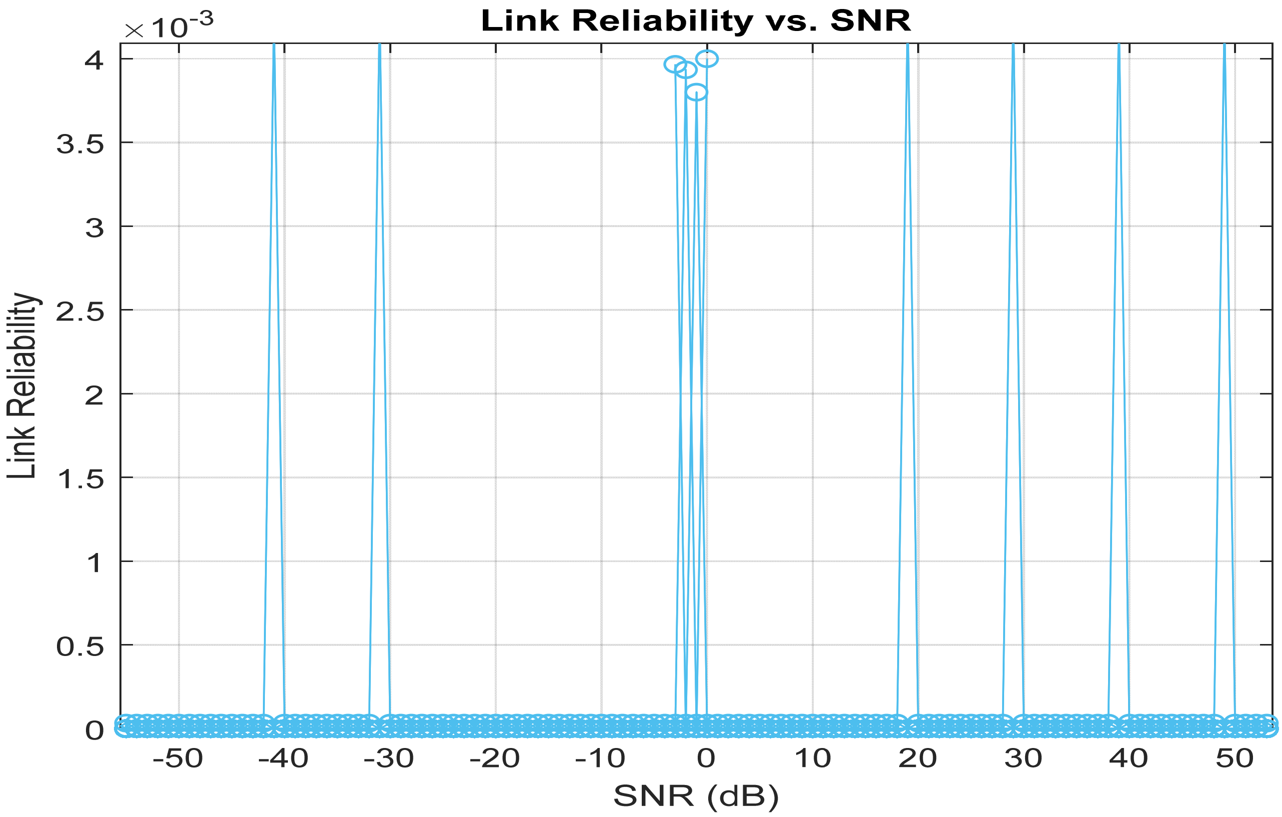

Figure 12: Link Reliability vs SNR

CHAPTER 8: DISCUSSION AND COMPARISON OF PERFORMANCE MEASUREMENTS

Findings from the results relating to the desired performance measurements from the testbed

8.1 Model Baseline

As stated in Chapter 6, this TestBed was based off the MATLAB file exchange repository; LoRaMatlab by (Al-Hourani, 2021), which was purely centred around the accurate simulation and comparison of theoretical to numerical Bit, Symbol and Packet Error rates. Most of the model is compiled into a singular function, with an aided function that handles the modulation of LoRa signals via generation of chirps, spreading factor and bandwidth settings.

8.2 Baseline additions implemented

Since this baseline model was built in MATLAB, it affords modularity, allowing other functions to be integrated. The expansion of which forms the basis of the TestBed. Variables and values are passed to and from the baseline function, with the addition of other calculations, visual and numerical outputs written directly therein. The modular functions added include, but are not limited to; the results displayed in graph format in Chapter 7, as well as the different telemetry reports described in Chapter 6 and partially displayed in APPENDIX J: Simulation Report Sample.

Naturally, the simulation was adapted to include ground station and satellite parameters over specific trajectories, most of which can be adjusted from the main.m function mentioned in Chapter 6. This was done for the most simplicity and least repetition of calculations.

8.3 Research Outputs

The primary output from this project is a functional testbed to simulate LoRa communication architecture on DtS nanosatellite constellations, including BER, SER and PER measurements, evaluating QoS metrics for a range of SNR values. These results were achieved with the TestBed, albeit notably with the requirement for further improvements.

The second research output, is accurate simulation results to motivate the whether benefit of this architecture to ASIC labs is viable for subsequent improvements and further implementation on proprietary satellite hardware. In its current state, simple observations can be made and it serves as a foundation for functional simulations to be made, however, further iteration for the accuracy of results and measurements are required.

8.4 Summary of Results

The simulation results are purely theoretical and have thus, not been validated. The aim however, was a functional TestBed. The knowledge base, is purely based on theory, thus all results displayed and discussed, still require closer observation by experienced engineers. The foundational groundwork however, affords a suitable starting point.

The range of input parameters can be adjusted as desired, most of which are called to and from the added and written main.m function as previously mentioned in Chapter 6. An example of these parameters can be determined from the report in Appendix J.

8.4.1 Packet Delivery Ratio vs SNR

As the measure of successful packets delivered between the ground station and satellite for an SNR range of -60dB to 60dB, the simulation displays varied levels of efficiency throughout. With a minimum Packet Delivery Ratio (PDR) just below 6.6×10^{-3} at around -10dB and a peak of 9×10^{-3} at around 10dB, this proves a reliable connection.

8.4.2 Average Throughput vs SNR

Following the PDR vs SNR, the average rate of successful data transmission over a specific channel, as expected, values thereof are somewhat poorer at lower displayed values of SNR, but still stable for the SNR range. For LoRa, this proves significant, since an SNR above 10 dB is often considered as reliable communication and moderate throughput for terrestrial LoRa communication.

8.4.3 Energy and Spectral Efficiency vs SNR

Where Energy Efficiency signifies the effectiveness of the system to utilize transmit power and Spectral Efficiency; the effective use of the frequency spectrum to transmitted data respectively over the aforementioned SNR range, high data rates are achieved within the available bandwidth at relatively moderate energy levels. The values appear high, but naturally, the simulation is optimized for best case scenario with all the available energy necessary.

8.4.4 Link Reliability vs SNR

Referring to how dependable the communication system is, based on the measure of how consistently and accurately data can be transmitted and received without errors proved tough to simulate. The displayed results however, indicate successes mostly in intervals of positive dB,

8.5 Summary of Outcome

8.5.1 Performance of the TestBed

A functional TestBed with accuracy of QoS performance metrics, calculations and subsequently, results were the core focus of the simulation, centred around the modularity and proof of performance of the model. Theoretically, these were achieved, with satisfying results. As mentioned however, the accuracy of the results and their numerical comparisons still require validation.

8.5.2 What is the state-of-the-art platform by comparison?

At present, there are a few popular platforms for LoRa communication architecture for satellites or constellations. LoRa networks have been primarily occupied with establishing terrestrial based presence. However, the new narrative of LoRaWAN satellite systems is in the early phases of commercialisation (Mohney, 2021).

-

8.5.3 Other open-source platforms

Semtech and the LoRa Alliance have provided resources that encourage user engagement through learning and experimentation (Semtech, n.d.). The freedom that these open-sourced LPWAN protocols provides, nurtures further innovation. As such, there are a few small open source LoRa projects and modules with varied simulation platforms (Open Source Libs, n.d.) that can be accessed to help assimilate a tailored LoRa testbed platform.

Among a number of reputable platforms to work from, a popular open-source software known as ns-3, described as “a discrete network simulator” is a simulation environment made primarily for research and educational use. This software encourages simulation models sufficiently realistic so as to be used as a real-time network simulator. This software will serve as a good comparison/ standard for the foundation and development of the intended LoRa testbed (nsnam, n.d.).

A popular alternative, OMNeT++, which is an object-oriented modular network simulation framework, provides the foundation and tools for writing network-based simulations. It is an ideal simulation platform for the development of satellite networks and constellations (OMNeT++, n.d.).

Another well-known open-source simulation framework specifically designed for simulating LoRa/ LoRaWAN networks, is appropriately named LoRaSim. It provides the evaluation and performance of LoRa/ LoRaWAN networks in terms of packet delivery ratio, CPU utilization, memory usage, execution time, and the number of collisions (Voigt & Bor, 2016).

8.5.4 Possible alternative use cases for this testbed?

After a successful proof of concept, further iterations succeeding the prototype could encourage the deployment of LoRa satellite configurations to bolster the range and effectiveness of terrestrial based LoRa systems. This would be especially useful for farmers and users in rural areas

8.5.5 Suggestions for further/ future improvements

After review and improvements by qualified practitioners to include other important environment variables, the finalized platform could form the groundwork for simulations used by other interested parties in satellite based LoRa communications to configure their own IoT constellations. Alternatively, the platform could be offered as a service to other members in the low budget, small satellite space industry.

The TestBed currently simulates for a single satellite and can be improved to simulate an entire constellation of satellites. With the scalability and functionality of MATLAB, more data and visualizations can be produced from the simulation, as well as other calculations.

What was initially intended as part of this body of work, was a Graphic User Interface (GUI). This would be a great addition, since currently, most of the simulation values are stored in the *main.m* function, with a few others declared in the base of the other functions. The addition of a GUI would allow the user to input all the parameters for the simulation, choose a destination for the reports and serve as a much friendly interface for the simulation as a complete package.

CHAPTER 9: CONCLUSION

Summarized findings

Building an accurate TestBed to simulate numerous values proved rather challenging, especially when attempting to simulate true numerical comparisons. There are a great many options to do so however, and MATLAB is a powerful adversary for continuous building and improvement of personalized environment simulations.

Thus. with enough iteration; successful and accurate simulation of a LoRa constellation tailored to test the feasibility of a LPWAN based telecommunications architecture with pre-set or unknown conditions, can provide performance of reliability and QoS measurements between a network of terrestrial transceivers and SDR based IoT payload.

Overall, the results obtained prove the feasibility of achieving readings that are viable as references for true numerical values if the simulation can account for realistic measurements.

REFERENCES

1. Afhamisis, M. & Palatella, M. R., 2016. SALSA: A Scheduling Algorithm for LoRa to LEO Satellites. *IEEE Wireless Communications,* 10, Volume 10, pp. pp. 60-67.
2. Ahmad, A. & Warip, M. N. B. M., 2016. Challenges to Inter-Satellite Communication System: A Review. *International Journal of Computer Applications.*
3. Al-Hourani, A. A.-T. F. &. A.-F. A., 2021. IoT Network Design using Open-Source LoRa Coverage Emulator. *IEEE Access,* 03 April.pp. 209, 684-209, 698.
4. Allan, A., 2018. *FossaSat-1, an Open Source Satellite for the Internet of Things.* [Online] Available at: https://www.hackster.io/news/fossasat-1-an-open-source-satellite-for-the-internet-of-things-7f31cab00ef5
[Accessed 13 05 2022].
5. ASIC Labs F'SATI, n.d. *About F'SATI.* [Online] Available at: https://blogs.cput.ac.za/fsati/
6. Bloechl, M., 2022. *What is LoRaWAN? Ultimate Guide for 2022.* [Online] Available at: https://www.link-labs.com/blog/what-is-lorawan#:~:text=LoRa%20works%20by%20moving%20an,tone%20is%20increasing%20over%20time.
[Accessed 12 08 2022].
7. Boquet, G. et al., 2020. *LR-FHSS: Overview and Performance Analysis.* [Online] Available at: https://arxiv.org/pdf/2010.00491.pdf
[Accessed 23 02 2023].
8. Brian, R., 2018. *SigFox Vs. LoRa: A Comparison Between Technologies & Business Models.* [Online]
Available at: https://www.link-labs.com/blog/sigfox-vs-lora
[Accessed 15 7 2020].
9. Camarillo, C., 2021. *Semtech Unveils LoRa® Corecell Reference Design for Full Duplex Gateway Applications Enabling LoRaWAN® Gateways to Receive and Transmit Data Simultaneously.* [Online]
Available at: https://www.semtech.com/company/press/semtech-unveils-lora-corecell-reference-design-for-full-duplex-gateway-applications-enabling-lorawan-gateways-to-receive-and-transmit-data-simultaneously
[Accessed 19 07 2022].
10. Capez, G. M., Henn, S., Garello, R. & Fraire, J. A., 2022. Sparse Satellite Constellation Design for Global and Regional Direct-to-Satellite IoT Services. 11 10, 58(5), pp. pp. 3946-3962.

11. Chaudhari, B. S., Zennaro, M. & Borkar, S., 2020. LPWAN Technologies: Emerging Application Characteristics, Requirements, and Design Considerations. *Future Internet,* 12(3), p. 46.
12. Chenhui, W. & Qingjia, M., 2020. Design of rapid monitoring system of geological disaster based on LoRa. *MATEC Web of Conferences,* Volume 306, p. 3006.
13. Colavolpe, G. et al., 2019. Reception of LoRa Signals From LEO Satellites. *IEEE Transactions on Aerospace and Electronic Systems,* 55(6), pp. 3587-3602.
14. D. Bulanov, K. O. Y. W. a. Q.-B. C., 2018. Inter-Satellite Communication for LEO CubeSat Network: QoS Parameters and Feasibility of Massive MIMO. *Journal of Communications Technology and Electronics(,* 63(10).
15. Daniłowski, P., 2021. *Battery life of LoRa devices.* [Online] Available at: https://yosensi.io/posts/battery-life-of-lora-devices/ [Accessed 08 10 2021].
16. Devopedia, n.d. *LoRa.* [Online] Available at: https://devopedia.org/lora#:~:text=LoRa%20doesn't%20need%20costly,built%20on%20top%20of%20LoRa%20.
[Accessed 14 7 2021].
17. DZone, n.d. [Online] Available at: https://dzone.com/articles/explaining-sigfox#:~:text=The%20SigFox%20network%20consists%20of,Cloud%20(Internet)
[Accessed 14 04 2022].
18. DZone, n.d. *DZone.* [Online] Available at: https://www.datacenterdynamics.com/en/news/iot-startup-sigfox-files-for-bankruptcy/
[Accessed 13 06 2021].
19. Fernandez, L., Ruiz-De-Azua, J. A., Calveras , A. & Camps, A., 2020. Assessing LoRa for Satellite-to-Earth Communications Considering the Impact of Ionospheric Scintillation. *IEEE Access,* 07 09.Volume 8.
20. Fernandez, L., Ruiz-De-Azua, J. A., Calveras, A. & Camps, A., 2020. Assessing LoRa for Satellite-to-Earth Communications Considering the Impact of Ionospheric Scintillation. *IEEE Access,* Volume 8, pp. 165570-165582.
21. FOSSASystems, 2020. *www.github.com/FOSSASystems/FOSSASAT-1B.* [Online] Available at: https://github.com/FOSSASystems/FOSSASAT-1B [Accessed 15 07 2022].
22. Garg, P., 2018. *Did we just prove the world isn't flat? or maybe it is ?.* [Online] Available at: https://www.thethingsnetwork.org/article/atmospheric-duct [Accessed 21 07 2021].

23. ITU Council, 2023. *Report on the implementation of the strategic plan and the activities of the union, july 2022 – april 2023.* [Online] Available at: https://www.itu.int/highlights-report-activities/2022-2023/wp-content/uploads/sites/3/2023/07/Annual-report-2022-2023.pdf
[Accessed 15 08 2023].
24. Jain, A., n.d. *Difference between LOS and NLOS in Wireless LoRa Technology.* [Online] Available at: https://www.engineersgarage.com/difference-between-los-and-nlos-in-wireless-lora-technology/
[Accessed 15 05 2022].
25. Jain, A., n.d. *Understanding LoRa architecture: nodes, gateways, and servers.* [Online] Available at: https://www.engineersgarage.com/understanding-lora-architecture-nodes-gateways-and-servers/
[Accessed 22 05 2022].
26. Kim, S., Lee, H. & Jeon, S., 2022. An Adaptive Spreading Factor Selection Scheme for a Single Channel LoRa Modem. *Sensors (Basel),* 10 February, 20(4), p. 1008.
27. Lacuna Space, n.d. *https://lacuna.space/about/.* [Online] Available at: https://www.lacuna.space/
[Accessed 08 03 2023].
28. Link Labs, 2017. *Firmware-over-the-Air (FOTA) with LoRa.* [Online] Available at: https://www.link-labs.com/blog/firmware-over-the-air-fota-with-lora
[Accessed 14 07 2022].
29. Link Labs, 2018. *SigFox Vs. LoRa: A Comparison Between Technologies & Business Models.* [Online]
Available at: https://www.link-labs.com/blog/sigfox-vs-lora#:~:text=LoRa%20is%20a%20spread%2Dspectrum,and%20thus%20gets%20more%20interference).
[Accessed 15 06 2021].
30. Link Labs, n.d. *Symphony Link™.* [Online] Available at: https://www.link-labs.com/symphony
[Accessed 17 07 2022].
31. Maleki, A., Nguyen, H. H., Bedeer, E. & Barton, R., 2022. D2D-aided LoRaWAN LR-FHSS in Direct-to-Satellite IoT Networks. *arXiv,* 08 12, Volume 2212.04331v1, p. 14.
32. McCleland, C., 2020. *IoT Connectivity - Comparing NB-IoT, LTE-M, LoRa, SigFox, and other LPWAN Technologies.* [Online] Available at: https://www.iotforall.com/iot-connectivity-comparison-lora-sigfox-rpma-lpwan-technologies#:~:text=become%20an%20issue.-,Symphony%20Link,Semtech's%20chips%20called%20Symphony%20Link.
[Accessed 28 07 2020].

33. McClelland, C., 2016. *Comparison of LPWAN Technologies - Which is Best for Me?.* [Online]
Available at: https://www.leverege.com/blogpost/comparison-of-lpwan-technologies [Accessed 9 8 2020].
34. Michalski, T., 2017. *Explaining SigFox.* [Online]
Available at: https://dzone.com/articles/explaining-sigfox#:~:text=The%20SigFox%20network%20consists%20of,Cloud%20(Internet) [Accessed 30 06 2020].
35. Mikhaylov , M. A. U. & Alves, H., 2022. Analysis and Simulation of LoRaWAN LR-FHSS for Direct-to-Satellite Scenario. *IEEE Wireless Communications Letters,* 03, 11(3), pp. 548-552.
36. Mishra, H., 2018. *Symphony Link – Best Alternative of LoRaWAN.* [Online]
Available at: https://iotbyhvm.ooo/symphony-link-best-alternative-of-lorawan/ [Accessed 25 6 2021].
37. Mitchell, B., 2019. *OSI Model Reference Guide.* [Online]
Available at: https://www.lifewire.com/osi-model-reference-guide-816289 [Accessed 14 06 2021].
38. Mohney, D., 2021. *LoRaWAN, the next big thing in satellite IoT?.* [Online]
Available at: https://www.spaceitbridge.com/lorawan-the-next-big-thing-in-satellite-iot.htm [Accessed 14 10 2021].
39. Nowatzki, J. & Emeritus, P., 2021. *Basics of LoRa Technology for Crop and Livestock Management.* [Online]
Available at: https://www.ndsu.edu/agriculture/ag-hub/publications/basics-lora-technology-crop-and-livestock-management#:~:text=This%20technology%20allows%20farmers%20to,stock%20tanks%20in%20remote%20pastures. [Accessed 24 07 2022].
40. nsnam, n.d. *Projects.* [Online]
Available at: https://www.nsnam.org/about/projects/ [Accessed 5 11 2021].
41. nsnam, n.d. *What is ns-3.* [Online]
Available at: https://www.nsnam.org/about/what-is-ns-3/ [Accessed 5 11 2021].
42. OMNeT++, n.d. *OMNeT++ Discrete Event Simulator.* [Online]
Available at: https://omnetpp.org/. [Accessed 06 08 2023].

43. Open Source Libs, n.d. *88 Open Source Lora Software Projects.* [Online] Available at: https://opensourcelibs.com/libs/lora [Accessed 5 11 2021].
44. Open Source Libs, n.d. *Summary.* [Online] Available at: https://opensourcelibs.com/lib/gr-lora_sdr [Accessed 3 11 2021].
45. Parada , R., Baeza, V. M., Barraca-Ibort, D. N. & Monzo, C., 2023. LoRa-Based Low-Cost Nanosatellite for Emerging Communication Networks in Complex Scenarios. *Space-Air-Ground-Integrated Communications in the New Era: Applications, Challenges, and Countermeasures,* 25 August.
46. Pinelo, J., Rocha, A. D. & Arvana, M., 2023. Unveiling LoRa's Oceanic Reach: Assessing the Coverage of the Azores LoRaWAN Network from an Island. *Sensors,* 23 09.
47. Queralta, J. P. et al., 2019. Comparative Study of LPWAN Technologies on Unlicensed Bands for M2M Communication in the IoT: beyond LoRa and LoRaWAN. *Procedia Computer Science,* Volume 155, pp. 343-350.
48. Rajiv, 2018. *Applications and future of LoRa WAN technology.* [Online] Available at: https://www.rfpage.com/applications-future-lora-wan-technology/ [Accessed 10 8 2020].
49. Ray, B., 2015. *What is SigFox.* [Online] Available at: https://www.link-labs.com/blog/what-is-sigfox [Accessed 29 06 2020].
50. Ray, B., 2018. *NB-IoT vs. LoRa vs. Sigfox.* [Online] Available at: https://www.link-labs.com/blog/nb-iot-vs-lora-vs-sigfox [Accessed 03 07 2020].
51. Ray, B., 2018. *SigFox Vs. LoRa: A Comparison Between Technologies & Business Models.* [Online] Available at: https://www.link-labs.com/blog/sigfox-vs-lora [Accessed 29 7 2020].
52. Ray, B., 2018. *What Is LoRaWAN? [Technical Breakdown].* [Online] Available at: https://www.link-labs.com/blog/what-is-lorawan [Accessed 28 06 2020].
53. RF Wireless World, n.d. *Home of RF and Wireless Vendors and Resources.* [Online] Available at: https://www.rfwireless-world.com/Terminology/Symphony-Link-vs-LoRaWAN.html#:~:text=Symphony%20Link%20is%20wireless%20system,architecture%20for%20the%20wireless%20system.&text=It%20is%20easy%20to%20use,no%20limit%20on%20duty%20cycle. [Accessed 16 05 2021].

54. RF Wireless World, n.d. *Symphony Link vs LoRaWAN-Difference between Symphony Link and LoRaWAN.* [Online] Available at: https://www.rfwireless-world.com/Terminology/Symphony-Link-vs-LoRaWAN.html
[Accessed 15 6 2021].
55. Saeed, N. et al., 2020. CubeSat communications: Recent advances and future challenges - arxiv.org. *arXiv,* 23 04, Volume 2, p. 24.
56. SAMSA, n.d. *Marine Domain Awareness.* [Online] Available at: https://www.samsa.org.za/Pages/Marine-Domain-Awareness.aspx
57. Semtech, 2022. *10 Things About LoRaWAN & NB-IoT.* [Online] Available at: 10 Things About LoRaWAN & NB-IoT
[Accessed 21 07 2022].
58. Semtech, n.d. *Learning Center.* [Online] Available at: https://lora-developers.semtech.com/learn/
[Accessed 12 10 2021].
59. Semtech, n.d. *LoRa (PHY).* [Online] Available at: https://www.semtech.com/lora/what-is-lora
[Accessed 13 09 2020].
60. Semtech, n.d. *LoRa Ecosystem.* [Online] Available at: https://www.semtech.com/lora/ecosystem/lora-alliance#:~:text=Established%20in%202015%2C%20the%20LoRa,the%20LoRaWAN%C2%AE%20open%20standard.
[Accessed 18 6 2020].
61. Semtech, n.d. *LoRa Technology Is Connecting Our Smart Planet.* [Online] Available at: https://www.semtech.com/lora/lora-applications
[Accessed 6 9 2020].
62. Semtech, n.d. *What are LoRa® and LoRaWAN®?.* [Online] Available at: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan
[Accessed 4 7 2020].
63. Shea, S., 2017. *LPWAN (low-power wide area network).* [Online] Available at: https://www.techtarget.com/iotagenda/definition/LPWAN-low-power-wide-area-network
[Accessed 16 10 2021].
64. Sigfox, 2018. *Sigfox,* Labrège: Sigfox.
65. Sigfox, n.d. *Our Story.* [Online] Available at: https://www.sigfox.com/en/sigfox-story
[Accessed 19 6 2020].

66. Slats, L., 2020. *A Brief History of LoRa®: Three Inventors Share Their Personal Story at The Things Conference.* [Online]
Available at: https://blog.semtech.com/a-brief-history-of-lora-three-inventors-share-their-personal-story-at-the-things-conference
[Accessed 2020].
67. Suarez, I. & Queiroz, C., 2022. *The Coming Era of Satellite Direct-to-Handset Connectivity.* [Online]
Available at: https://interactive.satellitetoday.com/via/december-2022/the-coming-era-of-satellite-direct-to-handset-connectivity/
[Accessed 21 04 2023].
68. Sun, Z. et al., 2022. Recent Advances in LoRa: A Comprehensive Survey. *ACM Transactions on Sensor Networks,* 18(4).
69. Swarm Space, 2022. *Swarm M138 Modem.* [Online]
Available at: https://swarm.space/
[Accessed 14 02 2023].
70. Tan, J., 2021. *A Gentle Introduction to LoRaWAN Gateways & Nodes.* [Online]
Available at: https://www.seeedstudio.com/blog/2021/04/27/a-gentle-introduction-to-lorawan-gateways-nodes/
[Accessed 21 03 2022].
71. Tapparel, J. et al., 2020. An Open-Source LoRa Physical Layer Prototype on GNU Radio. *arXiv.org,* 20 2.
72. Temim, B. & A., M., 2022. *Low-Earth-Orbit satellite communications using LoRa-like signals.* [Online]
Available at: https://www.theses.fr/2022BORD0089.pdf
[Accessed 21 05 2023].
73. The Things Network, n.d. *LoRaWAN Architecture.* [Online]
Available at: https://www.thethingsnetwork.org/docs/lorawan/architecture/
[Accessed 27 06 2021].
74. The Things Network, n.d. *What are LoRa and LoRaWAN?.* [Online]
Available at: https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/
[Accessed 21 4 2020].
75. Valerio, P., 2020. *LoRaWAN as wide-area, low-power, private IoT wireless network.* [Online]
Available at: https://iot.eetimes.com/lorawan-as-wide-area-low-power-private-iot-wireless-network/
[Accessed 2 5 2020].
76. Viasat, n.d. *Viasat real-time space relay.* [Online]
Available at: https://www.viasat.com/space-innovation/space-systems/intersatellite-

communications/
[Accessed 24 09 2022].

77. Voigt , T. & Bor, M., 2016. *LoRaSim.* [Online] Available at: https://www.lancaster.ac.uk/scc/sites/lora/lorasim.html [Accessed 19 08 2023].
78. Water Online, 2024. *LoRa Alliance Issues 2023 Annual Report Highlighting LoRaWAN Maturity, Robust Adoption, And Diversity Of End-To-End Solutions.* [Online] Available at: https://www.wateronline.com/doc/lora-alliance-issues-annual-report-highlighting-lorawan-maturity-diversity-solutions-0001
[Accessed 16 02 2024].
79. Xhonneux, M. et al., 2022. *A Two-User Successive Interference Cancellation LoRa Receiver with Soft-Decoding.* [Online] Available at: https://dial.uclouvain.be/pr/boreal/object/boreal%3A256596/datastream/PDF_01/view
[Accessed 12 09 2022].
80. Yoon, Z., Frese, W. & Briess, K., 2019. Design and Implementation of a Narrow-Band Intersatellite Network with Limited Onboard Resources for IoT. *Sensors 2019,* 27 09, 19(19), p. 4212.
81. Yukizane, M., Yokota, Y. & Kurii, T., 2021. Optical Inter-satellite Communication Technology for High-Speed, Large-Capacity Data Communications. *Special Issue on Social Infrastructure that Guarantees Safety, Security, Fairness, and Efficiency,* October.16(1).

APPENDIX A: BERLoRa.m and QoS

```
function [transmitted_data, SF, CtrSNR, SNR] =
BERLoRa(satellite_parameters, satellite_data,
ground_station_parameters)
% Clear the workspace and command window
clear
clc

%% Packet Size
Ns = 3e4 ; % Total number of symbols
% Symbols per packet
Nm = 1e1 ;
N = Ns/Nm ;

%% SF and BW initializing
SF = 7 ;          % Spreading Factor (SF)
M = 2^SF ;        % Number of modulation levels (M) based on SF
Ts = M ;          % Symbol duration
BW = 125e3 ;      % Bandwidth (BW)

%% Define parameters related to time
t = 0:1/Ts:0.999 ;            % Time vector for one symbol duration
tR = kron(ones(1,Ns),t) ;     % Replicate the time vector for the
entire packet

%% Create random symbols for transmission
x = randi(M,1,Ns) - 1 ;

%% Create Chirps for LoRa modulation
upChirp = loramod(x,SF,BW,BW) ;      % Create an up-chirp signal
dnChirp = loramod(0,SF,BW,BW,-1) ;  % Create a down-chirp signal
(used as reference signal)

% Generate the LoRa signal by mixing up and down chirps
signal = upChirp.*repmat(dnChirp,length(upChirp)./length(dnChirp),1)
;  %Carrier signal

%% Initiate Signal-to-Noise Ratio (SNR) parameters
SNR = -30 : 30; % Range of SNR values
EbNo = SNR + 10.*log10(M/SF) ;  % Energy per bit to Noise power
spectral density ratio
EsNo = EbNo + 10.*log10(SF) ;   % Energy per symbol to Noise power
spectral density ratio

%% Simulation Loop for different SNR values
for CtrSNR = 1 : length(SNR)
    % Calculate the noise standard deviation (sigma) based on SNR
    sigma = 1./10.^(SNR(CtrSNR)./20) ;

    % Simulate Rayleigh fading channel: Generate random complex
fading coefficients
    h = abs(reshape(1/sqrt(2).*(randn(1,Ns/Nm) +
j.*randn(1,Ns/Nm)).*ones(M.*Nm,1),M.*Ns,1)) ;
    n = sigma/sqrt(2).*(randn(M.*Ns,1) + j.*randn(M.*Ns,1)) ;
```

```
    % Simulate Additive White Gaussian Noise (AWGN) and Rayleigh
fading channels
    Rx1 = signal + n ;
    Rx2 = h.*signal + n ;

    %% Non-Coherent Detection
    % Non-coherent detection for AWGN channel
    [~,idx] = max(fft(reshape(Rx1,M,Ns))) ;
    ynCoh1 = idx - 1 ;
    % Non-coherent detection for Rayleigh channel
    [~,idx] = max(fft(reshape(Rx2,M,Ns))) ;
    ynCoh2 = idx - 1 ;

    %% Coherent Detection
    for CtrM = 1 : M
        % AWGN
        rtemp1 = conv(Rx1,exp(-j.*2.*pi.*(M - CtrM + 1).*t)) ;
        r1(CtrM,:) = real(rtemp1(Ts+1:Ts:end)) ;
        % Rayleigh
        rtemp2 = conv(Rx2,exp(-j.*2.*pi.*(M - CtrM + 1).*t)) ;
        r2(CtrM,:) = real(rtemp2(Ts+1:Ts:end)) ;
    end

    % Coherent detection results for AWGN and Rayleigh channels
    [~,idx] = max(r1) ;
    yCoh1 = idx - 1 ;

    [~,idx] = max(r2) ;
    yCoh2 = idx - 1 ;

    %% BER Calculation and Simulation
    BER_nCOH_AWGN_SIMULATION(CtrSNR) = sum(sum(abs(de2bi(ynCoh1,M) -
de2bi(x,M))))/(SF.*Ns) ;
    BER_nCOH_RAY_SIMULATION(CtrSNR)  = sum(sum(abs(de2bi(ynCoh2,M) -
de2bi(x,M))))/(SF.*Ns) ;
    BER_COH_AWGN_SIMULATION(CtrSNR)  = sum(sum(abs(de2bi(yCoh1,M) -
de2bi(x,M))))/(SF.*Ns) ;
    BER_COH_RAY_SIMULATION(CtrSNR)   = sum(sum(abs(de2bi(yCoh2,M) -
de2bi(x,M))))/(SF.*Ns) ;

    %% SER Calculation and Simulation
    SER_nCOH_AWGN_SIMULATION(CtrSNR) = sum(abs(ynCoh1 - x)>0)/Ns ;
    SER_nCOH_RAY_SIMULATION(CtrSNR)  = sum(abs(ynCoh2 - x)>0)/Ns ;
    SER_COH_AWGN_SIMULATION(CtrSNR)  = sum(abs(yCoh1 - x)>0)/Ns ;
    SER_COH_RAY_SIMULATION(CtrSNR)   = sum(abs(yCoh2 - x)>0)/Ns ;

    %% PER Calculation and Simulation
    PER_nCOH_AWGN_SIMULATION(CtrSNR) = sum(sum(abs(reshape(x,N,Nm) -
reshape(ynCoh1,N,Nm)),2)>0)/N ;
    PER_nCOH_RAY_SIMULATION(CtrSNR)  = sum(sum(abs(reshape(x,N,Nm) -
reshape(ynCoh2,N,Nm)),2)>0)/N ;
    PER_COH_AWGN_SIMULATION(CtrSNR)  = sum(sum(abs(reshape(x,N,Nm) -
reshape(yCoh1,N,Nm)),2)>0)/N ;
    PER_COH_RAY_SIMULATION(CtrSNR)   = sum(sum(abs(reshape(x,N,Nm) -
reshape(yCoh2,N,Nm)),2)>0)/N ;

    %% Loading Display
```

```
    CtrSNR
    disp(['Processing SNR: ' num2str(SNR(CtrSNR)) ' dB']);
end

%% Analytic
SNR_theory = linspace(-30,30,1e3) ;
EbNo_theory = SNR_theory + 10.*log10(M/SF) ;
EsNo_theory = EbNo_theory + 10.*log10(SF) ;

% % Initialize variables to store theoretical BER results
% BER_nCOH_AWGN_Theory = zeros(size(SNR_theory));
% BER_nCOH_RAY_Theory = zeros(size(SNR_theory));
% BER_COH_AWGN_Theory = zeros(size(SNR_theory));
% % BER_COH_RAY_Theory = zeros(size(SNR_theory));
%
% % Initialize variables to store theoretical SER results
% SER_nCOH_AWGN_Theory = zeros(size(SNR_theory));
% SER_nCOH_RAY_Theory = zeros(size(SNR_theory));
% SER_COH_AWGN_Theory = zeros(size(SNR_theory));
% % SER_COH_RAY_Theory = zeros(size(SNR_theory));
%
% % Initialize variables to store theoretical PER results
% PER_nCOH_AWGN_Theory = zeros(size(SNR_theory));
% PER_nCOH_RAY_Theory = zeros(size(SNR_theory));
% PER_COH_AWGN_Theory = zeros(size(SNR_theory));
% % PER_COH_RAY_Theory = zeros(size(SNR_theory));

% Theoretical BER Calculations
BER_nCOH_AWGN_Theory = 0.5.*qfunc(sqrt(10.^(SNR_theory./10).*2.*M) -
sqrt(1.386.*SF+1.154)) ; % Using a closed-form approximation for
LoRa modulation
BER_nCOH_RAY_Theory  =
berfading(EbNo_theory,'fsk',M,1,'noncoherent') ; % Using the BER
fading function for non-coherent detection
BER_COH_AWGN_Theory  = berawgn(EbNo_theory,'fsk',M,'coherent') ; %
Using the BER AWGN function for coherent detection
% BER_COH_RAY_Theory   = berfading(EbNo_theory,'fsk',M,1,'coherent')
;

% Theoretical SER Calculations
SER_nCOH_AWGN_Theory = (M - 1)./(2.^(SF - 1)).*BER_nCOH_AWGN_Theory
; % Using SER formula with non-coherent AWGN detection
SER_nCOH_RAY_Theory  = (M - 1)./(2.^(SF - 1)).*BER_nCOH_RAY_Theory ;
% Using SER formula with non-coherent Rayleigh detection
SER_COH_AWGN_Theory  = (M - 1)./(2.^(SF - 1)).*BER_COH_AWGN_Theory ;
% Using SER formula with coherent AWGN detection
% SER_COH_RAY_Theory   = (M - 1)./(2.^(SF - 1)).*BER_COH_RAY_Theory
;

% Theoretical PER Calculations
PER_nCOH_AWGN_Theory = 1 - (1 - SER_nCOH_AWGN_Theory).^Nm ; % Using
PER formula with non-coherent AWGN detection
PER_nCOH_RAY_Theory  = 1 - (1 - SER_nCOH_RAY_Theory).^Nm ; % Using
PER formula with non-coherent Rayleigh detection
PER_COH_AWGN_Theory  = 1 - (1 - SER_COH_AWGN_Theory).^Nm ; % Using
PER formula with coherent AWGN detection
% PER_COH_RAY_Theory   = 1 - (1 - SER_COH_RAY_Theory).^Nm ;
```

```
% Create a structure to hold/ store simulation results
transmitted_data = struct();
transmitted_data.SF = SF;
transmitted_data.SNR = SNR;
transmitted_data.CtrSNR = CtrSNR;
transmitted_data.signal = signal;
transmitted_data.BW = BW;
transmitted_data.BER = struct('nCOH_AWGN', BER_nCOH_AWGN_SIMULATION,
...
                              'nCOH_RAY', BER_nCOH_RAY_SIMULATION,
...
                              'COH_AWGN', BER_COH_AWGN_SIMULATION,
...
                              'COH_RAY', BER_COH_RAY_SIMULATION);
transmitted_data.SER = struct('nCOH_AWGN', SER_nCOH_AWGN_SIMULATION,
...
                              'nCOH_RAY', SER_nCOH_RAY_SIMULATION,
...
                              'COH_AWGN', SER_COH_AWGN_SIMULATION,
...
                              'COH_RAY', SER_COH_RAY_SIMULATION);
transmitted_data.PER = struct('nCOH_AWGN', PER_nCOH_AWGN_SIMULATION,
...
                              'nCOH_RAY', PER_nCOH_RAY_SIMULATION,
...
                              'COH_AWGN', PER_COH_AWGN_SIMULATION,
...
                              'COH_RAY', PER_COH_RAY_SIMULATION);

%% Plots
% BER vs Energy per bit to Noise power spectral density ratio
figure(1)
clf
ax = gca ;
hold on
plot(EbNo_theory,BER_nCOH_AWGN_Theory,'k-')
plot(EbNo_theory,BER_nCOH_RAY_Theory,'k-')
plot(EbNo_theory,BER_COH_AWGN_Theory,'k--')
% plot(EbNo_theory,BER_COH_RAY_Theory,'k--')
plot(EbNo,BER_nCOH_AWGN_SIMULATION,'o','color',ax.ColorOrder(1,:))
plot(EbNo,BER_nCOH_RAY_SIMULATION,'o','color',ax.ColorOrder(2,:))
plot(EbNo,BER_COH_AWGN_SIMULATION,'d','color',ax.ColorOrder(1,:))
plot(EbNo,BER_COH_RAY_SIMULATION,'d','color',ax.ColorOrder(2,:))
xlabel(' EbNo [dB] ')
ylabel(' BER ')
title('Bit Error Rate vs. Eb/No')
box on
grid on

% SER Energy per symbol to Noise power spectral density ratio
figure(2)
clf
ax = gca ;
hold on
plot(EsNo_theory,SER_nCOH_AWGN_Theory,'k-')
plot(EsNo_theory,SER_nCOH_RAY_Theory,'k-')
```

```
plot(EsNo_theory,SER_COH_AWGN_Theory,'k--')
% plot(EsNo_theory,SER_COH_RAY_Theory,'k--')
plot(EsNo,SER_nCOH_AWGN_SIMULATION,'o','color',ax.ColorOrder(1,:))
plot(EsNo,SER_nCOH_RAY_SIMULATION,'o','color',ax.ColorOrder(2,:))
plot(EsNo,SER_COH_AWGN_SIMULATION,'d','color',ax.ColorOrder(1,:))
plot(EsNo,SER_COH_RAY_SIMULATION,'d','color',ax.ColorOrder(2,:))
xlabel(' EsNo [dB] ')
ylabel(' SER ')
title('Symbol Error Rate vs. Es/No')
box on
grid on

% PER vs SNR
figure(3)
clf
ax = gca ;
hold on
plot(SNR_theory,PER_nCOH_AWGN_Theory,'k-')
plot(SNR_theory,PER_nCOH_RAY_Theory,'k-')
plot(SNR_theory,PER_COH_AWGN_Theory,'k--')
% plot(SNR_theory,PER_COH_RAY_Theory,'k--')
plot(SNR,PER_nCOH_AWGN_SIMULATION,'o','color',ax.ColorOrder(1,:))
plot(SNR,PER_nCOH_RAY_SIMULATION,'o','color',ax.ColorOrder(2,:))
plot(SNR,PER_COH_AWGN_SIMULATION,'d','color',ax.ColorOrder(1,:))
plot(SNR,PER_COH_RAY_SIMULATION,'d','color',ax.ColorOrder(2,:))
xlabel(' SNR [dB] ')
ylabel(' PER ')
title('Packet Error Rate vs. SNR')
box on
grid on

%% QoS metrics
    % Initialize the received packet matrix
    packet_received = zeros(Ns, SF);
    transmission_time = zeros(1, Ns);  % Initialize transmission
times
    successful_indices = [];  % Initialize successful indices

    % Initialize variables to store PDR_SIMULATION for each SNR
    PDR_SIMULATION = zeros(1, length(SNR));

    for CtrSNR = 1:length(SNR)
       % Initialize reception_time as a vector of NaN values
       reception_time = NaN(1, length(successful_indices));

        % Calculate reception time for each successful packet
        for i = 1:length(successful_indices)
            idx = successful_indices(i);  % Index of the successful
packet
            % Ensure that idx is within valid bounds
            disp(['Processing index: ', num2str(idx)]);
            if idx <= length(transmission_time) && idx <=
length(SNR)
            transmission_duration = transmission_time(idx);
            % Assuming a linear relationship with SNR
            reception_time(i) = transmission_duration / (1 +
SNR(idx));
```

```
            else
%             disp(['Index out of bounds: ', num2str(idx)]);
            reception_time(i) = NaN;  % Or use any other suitable
value
            end
        end

        % Initialize Calculate QoS metrics for the current SNR
        avg_packet_delay = zeros(size(SNR));
        jitter = zeros(size(SNR));
        avg_throughput = zeros(size(SNR));
        energy_efficiency = zeros(size(SNR));
        spectral_efficiency = zeros(size(SNR));
        link_reliability = zeros(size(SNR));

        % Initialize variables to count transmitted and received
packets
        total_transmitted_packets = 0;
        total_received_packets = 0;

       % Calculate average packet delay for successful receptions
       successful_indices = find(packet_received(:, SF));

        if ~isempty(successful_indices) && CtrSNR <=
size(transmission_time, 2)
            % Filter out valid indices
            valid_indices = successful_indices(successful_indices <=
length(reception_time));
            if ~isempty(valid_indices)
                % If there are valid indices left, calculate delays,
average packet delay, and jitter
                packet_delays = reception_time(valid_indices) -
transmission_time(valid_indices);
                avg_packet_delay = mean(packet_delays);
                jitter = std(packet_delays); % Calculate the
standard deviation as jitter
% %             else
% %                 avg_packet_delay(CtrSNR) = NaN;  % No valid
packets received successfully
% %                 jitter(CtrSNR) = NaN;  % No valid packets
received successfully
            end
%         else
%             avg_packet_delay(CtrSNR) = NaN;  % No packets received
successfully or CtrSNR is out of bounds
%             jitter(CtrSNR) = NaN;  % No packets received
successfully or CtrSNR is out of bounds
        end



    %% Calculate Transmission time & Packet reception
    % Placeholder variables
    data_rate_bps = 50000;  % 50 kbps (bits per second)
    packet_size_bits = 1000;  % 1000 bits
    noise = zeros(1, length(SNR) * SF);
```

```
    for packet_idx = 1:Ns
        % Generate random binary packet (0s and 1s)
        transmitted_packet = randi([0, 1], 1, SF);
        % Record the transmission time for this packet
        transmission_time(packet_idx) = packet_size_bits /
data_rate_bps;

        % Calculate noise for each SNR value
        for i = 1:length(SNR)
            snr_lin = 10^(SNR(i) / 10);
            noise_var = sqrt(1 / (2 * snr_lin));

            % Replicate transmitted_packet for each SNR value
            modulated_symbols = (2 * transmitted_packet - 1);

            % Add noise for this SNR value
            noise((i - 1) * SF + 1:i * SF) = noise_var * randn(1,
SF);

            % Combine modulated_symbols and noise for this SNR
            received_symbols = modulated_symbols + noise((i - 1) *
SF + 1:i * SF);
        end

        % Demodulate the received symbols (simple thresholding)
        received_packet = received_symbols > 0;
        % Store the received packet in the matrix
        packet_received(packet_idx, :) = received_packet;

        % Check if the packet was successfully received
        if all(received_packet)
            % Store the index of this successful packet
            successful_indices(end+1) = packet_idx;
            transmission_time = transmission_time(packet_idx);
            total_received_packets = total_received_packets + 1;
        end
         % Count this packet as transmitted
         total_transmitted_packets = total_transmitted_packets + 1;
    end
        % Calculate PDR
        PDR_SIMULATION(CtrSNR) = total_received_packets /
total_transmitted_packets;
        % Calculate the total number of bytes received for this SNR
value
        total_bytes_received = sum(de2bi(yCoh2(:, CtrSNR), M),
'all');
        % Calculate the simulation time for this SNR value (assuming
one symbol duration per packet)
        simulation_time = N / Nm * Ts;

        % Calculate the average throughput (in bits per second)
        avg_throughput(CtrSNR) = total_bytes_received /
simulation_time;
        % Calculate the energy efficiency (in bits per Joule)
        energy_efficiency(CtrSNR) = PDR_SIMULATION(CtrSNR) /
(sigma^2 / BW);
```

```
        % Calculate the spectral efficiency (in bits per second per
Hertz)
        spectral_efficiency(CtrSNR) = PDR_SIMULATION(CtrSNR) * SF *
BW;
        % Define the threshold for successful reception (e.g., 10%
of SF)
        threshold = 0.1 * SF;
        % Calculate the link reliability (percentage of packets with
RSSI above the threshold)
        link_reliability(CtrSNR) = sum(sum(abs(r2(:, CtrSNR)) >
threshold)) / Ns;


%% Plot QoS Metrics
    % Plot 2: Packet Delivery Ratio (PDR) vs. SNR
    figure(4);
    clf
    ax = gca;
    hold on;
    plot(SNR, PDR_SIMULATION, 'o-', 'DisplayName',
'PDR','color',ax.ColorOrder(1,:));
    plot(SNR, PDR_SIMULATION, 'k-', 'DisplayName',
'PDR','color',ax.ColorOrder(2,:));
    xlabel('SNR (dB)');
    ylabel('Packet Delivery Ratio (PDR)');
    title('PDR vs. SNR');
    box on;
    grid on;

    % Plot 3: Average Packet Delay vs. SNR
    figure(5);
%     clf
    ax = gca;
    hold on;
    plot(SNR, avg_packet_delay, '-o', 'DisplayName', 'Average Packet
Delay','color',ax.ColorOrder(2,:));
    plot(SNR, avg_packet_delay, 'k-', 'DisplayName', 'Average Packet
Delay','color',ax.ColorOrder(3,:));
    xlabel('SNR (dB)');
    ylabel('Latency/ Average Packet Delay (s)');
    title('Latency/ Average Packet Delay vs. SNR');
    box on;
    grid on;

    % Plot 4: Average Throughput vs. SNR
    figure(6);
%     clf
    ax = gca;
    hold on;
    plot(SNR, avg_throughput, '-o', 'DisplayName', 'Average
Throughput','color',ax.ColorOrder(3,:));
    plot(SNR, avg_throughput, 'k-', 'DisplayName', 'Average
Throughput','color',ax.ColorOrder(4,:));
    xlabel('SNR (dB)');
    ylabel('Average Throughput (bps)');
    title('Average Throughput vs. SNR');
    box on;
```

```
    grid on;

    % Plot 5: Energy Efficiency vs. SNR
    figure(7);
%     clf
    ax = gca;
    hold on;
    plot(SNR, energy_efficiency, '-o', 'DisplayName', 'Energy
Efficiency','color',ax.ColorOrder(4,:));
    plot(SNR, energy_efficiency, 'k-', 'DisplayName', 'Energy
Efficiency','color',ax.ColorOrder(5,:));
    xlabel('SNR (dB)');
    ylabel('Energy Efficiency (bits/Joule)');
    title('Energy Efficiency vs. SNR');
    box on;
    grid on;

    % Plot 6: Spectral Efficiency vs. SNR
    figure(8);
%     clf
    ax = gca;
    hold on;
    plot(SNR, spectral_efficiency, '-o', 'DisplayName', 'Spectral
Efficiency','color',ax.ColorOrder(5,:));
    plot(SNR, spectral_efficiency, 'k-', 'DisplayName', 'Spectral
Efficiency','color',ax.ColorOrder(6,:));
    xlabel('SNR (dB)');
    ylabel('Spectral Efficiency (bps/Hz)');
    title('Spectral Efficiency vs. SNR');
    box on;
    grid on;

    % Plot 7: Link Reliability vs. SNR
    figure(9);
%     clf
    ax = gca;
    hold on;
    plot(SNR, link_reliability, '-o', 'DisplayName', 'Link
Reliability','color',ax.ColorOrder(6,:));
    plot(SNR, link_reliability, 'k-', 'DisplayName', 'Link
Reliability','color',ax.ColorOrder(7,:));
    xlabel('SNR (dB)');
    ylabel('Link Reliability');
    title('Link Reliability vs. SNR');
    box on;
    grid on;

%     % Plot 9: Jitter vs. SNR
%     figure(10);
%     clf
%     ax = gca;
%     hold on;
%     plot(SNR, jitter, '-o', 'DisplayName',
'Jitter','color',ax.ColorOrder(7,:));
%     plot(SNR, jitter, 'k-', 'DisplayName',
'Jitter','color',ax.ColorOrder(7,:));
%     xlabel('SNR (dB)');
```

```
%     ylabel('Jitter');
%     title('Jitter vs. SNR');
%     box on;
%     grid on;

%% Store QoS Results in output
    % Store QoS metrics in the result structure
    transmitted_data.PDR = PDR_SIMULATION;
    transmitted_data.avg_packet_delay = avg_packet_delay;
    transmitted_data.avg_throughput = avg_throughput;
    transmitted_data.energy_efficiency = energy_efficiency;
    transmitted_data.spectral_efficiency = spectral_efficiency;
    transmitted_data.link_reliability = link_reliability;
    transmitted_data.jitter = jitter;

    end     % End of CtrSNR loop

    % Display or save the QoS metrics as needed
    disp('QoS metrics calculated.');
    disp('Simulation completed.');
end     % End of main BERLoRa function
```

APPENDIX B: loramod.m

```
function [y] = loramod(x, SF, BW, fs, varargin)
% LoRa Modulation Function
% Inputs:
%   x: Input symbols (0 to M-1)
%   SF: Spreading factor
%   BW: Bandwidth
%   fs: Sampling frequency
%   varargin: Optional modulation inversion parameter
% Output:
%   y: Modulated LoRa signal

% Input validation
if nargin < 4
    error('Error: Not enough input arguments. Usage: loramod(x, SF,
BW, fs, varargin)');
end

if nargin > 5
    error('Error: Too many input arguments. Usage: loramod(x, SF,
BW, fs, varargin)');
end

M = 2^SF;

% Check that M is a positive integer power of 2
if (~isreal(M) || ~isscalar(M) || M <= 0 || (ceil(M) ~= M) ||
~isnumeric(M))
    error('Error: Invalid value for SF. SF must be a positive
integer power of 2.');
end

% Check that x is within range
if (any(x < 0) || any(x >= M))
    error('Error: Input symbols (x) must be non-negative integers
less than M.');
end

% Default inversion setting
Inv = 1;

if nargin == 5
    Inv = varargin{1};
end

% Calculate basic parameters
Ts = 2^SF / BW;
beta = BW / (2 * Ts);
n_symbol = fs * M / BW;
t_symbol = (0:n_symbol - 1) * 1 / fs;

% Initialize the output signal
y = [];

% Modulate each symbol in x
for ctr = 1 : length(x)
```

```
    gamma = (x(ctr) - M/2) * BW / M;
    lambda = 1 - x(ctr) / M;
    t1 = t_symbol(1:end * lambda);
    t2 = t_symbol(end * lambda + 1:end);

    % Calculate modulation for the symbol
    modulation1 = exp(-j * 2 * pi * (t1' * gamma + beta * t1'.^2) *
Inv);
    modulation2 = exp(-j * 2 * pi * (t2' * (-BW + gamma) + beta *
t2'.^2) * Inv);

    % Concatenate the modulated symbols to the output signal
    y = [y; modulation1; modulation2];
end

% Reshape the output signal to a row vector
y = reshape(y, 1, numel(y))';
end
```

APPENDIX C: satellite_simulation.m

```
function satellite_data = satellite_simulation(satellite_parameters, 
ground_station_parameters)
    % Simulate satellite trajectory based on the provided parameters

    % Extract satellite parameters
    orbit_altitude_km = satellite_parameters.OrbitAltitude_km;
    orbit_inclination_deg = satellite_parameters.Inclination_deg;
    eccentricity = satellite_parameters.eccentricity;   % Satellite 
Eccentricity
    % orbit_period_min = satellite_parameters.OrbitPeriod_min;

    % Extract ground station parameters
    ground_station_latitude_deg = 
ground_station_parameters.GroundStationLocation(1)
    ground_station_longitude_deg = 
ground_station_parameters.GroundStationLocation(2)

    % Define constants
    earth_radius_km = 6371;  % Earth radius in kilometers
    G = 6.67430e-11;  % Gravitational constant (m^3/kg/s^2)
    earth_mass_kg = 5.972e24;  % Earth mass in kilograms

    % Convert altitude to meters
    orbit_altitude_m = orbit_altitude_km * 1e3;

    % Calculate semi-major axis of the orbit (making it elliptical)
    semi_major_axis_m = earth_radius_km * 1e3 + orbit_altitude_m;

    % Calculate orbital velocity using vis-viva equation
    orbit_velocity_mps = sqrt((G * earth_mass_kg) / 
semi_major_axis_m);

    % Calculate orbital period in seconds
    orbit_period_seconds = 2 * pi * sqrt(semi_major_axis_m^3 / (G * 
earth_mass_kg));

    % Orbital period in minutes
    orbit_period_min = (orbit_period_seconds/60);

    % Create time vector
    time_seconds = 0:60:orbit_period_seconds;

    % Initialize arrays to store position data
    satellite_longitude_deg = zeros(size(time_seconds));
    satellite_latitude_deg = zeros(size(time_seconds));

    % Calculate satellite position at each time step
    for t = 1:length(time_seconds)
        % Calculate the satellite's mean anomaly
        mean_anomaly_rad = (2 * pi * t) / (orbit_period_seconds);

        % Calculate the satellite's true anomaly using Kepler's 
equation (elliptical orbit)
        eccentric_anomaly_rad = 2 * atan(sqrt((1 - eccentricity) / 
(1 + eccentricity)) * tan(mean_anomaly_rad / 2));
```

```
        true_anomaly_rad = 2 * atan(sqrt((1 + eccentricity) / (1 -
eccentricity)) * tan(eccentric_anomaly_rad / 2));

        % Calculate the satellite's longitude
        satellite_longitude_rad = true_anomaly_rad +
orbit_inclination_deg * pi / 180;  % Convert inclination to radians

        % Calculate the satellite's latitude for an elliptical orbit
        satellite_latitude_rad = asin(sin(orbit_inclination_deg * pi
/ 180) * sin(true_anomaly_rad));

        % Convert longitude and latitude to degrees
        satellite_longitude_deg(t) =
rad2deg(satellite_longitude_rad);
        satellite_latitude_deg(t) = rad2deg(satellite_latitude_rad);
    end

    % Create the satellite data structure
    satellite_data = struct();
    satellite_data.OrbitAltitude_km = orbit_altitude_km;
    satellite_data.Inclination_deg = orbit_inclination_deg;
    satellite_data.OrbitPeriod_min = orbit_period_min;      %
Orbital period in minutes
    satellite_data.OrbitVelocity_mps = orbit_velocity_mps;
    satellite_data.Longitude_deg = satellite_longitude_deg;
    satellite_data.Latitude_deg = satellite_latitude_deg;

    %% Visualization plots
    % Plot Satellite Trajectory
    figure(11);
    plot(satellite_longitude_deg, satellite_latitude_deg);
    xlabel('Longitude (degrees)');
    ylabel('Latitude (degrees)');
    title('Satellite Orbit');

    % Plot Satellite Ground Track
    figure(12);
    plot(satellite_longitude_deg, satellite_latitude_deg);
    xlabel('Longitude (degrees)');
    ylabel('Latitude (degrees)');
    title('Satellite Orbit');
    grid on;

    % Create a World Map
    worldmap('World');
    % Plot the satellite ground track
    geoshow(satellite_latitude_deg, satellite_longitude_deg,
'DisplayType', 'line', 'Color', 'b', 'LineWidth', 2);
    title('Satellite Ground Track');
    grid on;

    % Visibility Coverage Plot
    % Calculate difference in longitude between satellite and ground
station
    delta_longitude = satellite_longitude_deg -
ground_station_longitude_deg;
```

```
    % Calculate azimuth using Haversine formula
    azimuth = atan2d(sin(deg2rad(delta_longitude)),
cos(deg2rad(ground_station_latitude_deg)) *
tan(deg2rad(satellite_latitude_deg)) -
sin(deg2rad(ground_station_latitude_deg)) *
cos(deg2rad(delta_longitude)));

    % Calculate range using the law of cosines
    range = sqrt((earth_radius_km + orbit_altitude_km).^2 +
earth_radius_km.^2 - 2 * (earth_radius_km + orbit_altitude_km) *
earth_radius_km .* cos(deg2rad(delta_longitude)));

    % Create a polar plot
    figure(13);
    polarplot(deg2rad(azimuth), range, 'b', 'LineWidth', 2);
    title('Satellite Visibility Coverage');
    grid on;

    % Create a Polar Plot
    figure(14);
    % Convert altitude to meters
    satellite_altitude_m = orbit_altitude_km * 1e3;
    polarplot(deg2rad(satellite_longitude_deg),
satellite_altitude_m, 'b', 'LineWidth', 2);

    % Customize the polar plot
    title('Satellite Orbit Shape');
    grid on;

end
```

APPENDIX D: propagation_delay.m

```
function propagation_delay_sec =
propagation_delay(ground_station_parameters, satellite_data,
satellite_parameters)
    % Extract latitude and longitude of the ground station
    ground_station_latitude_deg =
ground_station_parameters.GroundStationLocation(1);
    ground_station_longitude_deg =
ground_station_parameters.GroundStationLocation(2);

    % Extract satellite latitude and longitude (use the values at a
specific time step)
    satellite_latitude_deg = satellite_data.Latitude_deg;
    satellite_longitude_deg = satellite_data.Longitude_deg;

    % Extract other parameters
    % Speed of light in km/s (approximately 299,792 km/s)
    speed_of_light_kmps = satellite_parameters.speed_of_light_kmps;

    % Calculate the distance between ground station and satellite
    distance_km = distance(ground_station_latitude_deg,
ground_station_longitude_deg, satellite_latitude_deg,
satellite_longitude_deg, earthRadius('km'));

    % Calculate propagation delay in seconds
    propagation_delay_sec = distance_km / speed_of_light_kmps;

    % Display the propagation delay in seconds (Tests)
    % disp(['Propagation Delay: ' num2str(propagation_delay_sec) '
seconds\n']);
    % fprintf('\nPropagation Delay: %.6f\n', propagation_delay_sec,
'seconds\n');
    % fprintf('Latitude %.4f, \nLongitude %.4f\n',
    % ground_station_parameters.GroundStationLocation); % Ground
Station location
    % fprintf('\nsatellite_position(deg):\nLatitude:
%.4f\nLongitude: %.4f\n\n', satellite_latitude_deg,
satellite_longitude_deg);
end
```

APPENDIX E: FSPL.m

```
function FSPL_dB = FSPL(transmitted_data, satellite_data, 
satellite_parameters, ground_station_parameters)
% Initialize arrays to store FSPL
BW = transmitted_data.BW;
SNR = transmitted_data.SNR;
FSPL = zeros(size(SNR));

SF = 7;
current_size = 0;

% Ground station location
lat_ground_station = 
deg2rad(ground_station_parameters.GroundStationLocation(1));  % 
Latitude in radians
lon_ground_station = 
deg2rad(ground_station_parameters.GroundStationLocation(2));  % 
Longitude in radians

% Satellite orbit altitude in meters
orbit_altitude_m = satellite_parameters.OrbitAltitude_km * 1000;

% Link frequency in Hz
phase = angle(transmitted_data.signal);
f_carrier = (diff(phase) / (2 * pi * ((2^SF)/BW))*1000); % Carrier 
frequency in Hz

    for t_index = 1:length(SNR)
        %t = SNR(t_index);

        % Debugging prints
        % disp(['t_index = ', num2str(t_index)]);
        % disp(['size(SNR) = ', num2str(size(SNR))]);

        % Calculate the distance between satellite and ground 
station using haversine formula
        % Assuming a spherical Earth
        R = 6371e3;  % Radius of the Earth in meters
        dlon = deg2rad(satellite_data.Longitude_deg(t_index)) - 
lon_ground_station;
        dlat = deg2rad(satellite_data.Latitude_deg(t_index)) - 
lat_ground_station;
        a = sin(dlat/2)^2 + cos(lat_ground_station) * 
cos(satellite_parameters.Inclination_deg * (pi / 180)) * 
sin(dlon/2)^2;
        c = 2 * atan2(sqrt(a), sqrt(1-a));
        d = R * c + orbit_altitude_m;  % Total distance including 
altitude

        % Calculate FSPL in dB and store it in the next available 
slot
        % FSPL(t_index) = 20 * log10(4 * pi * d * f_carrier / c);
        % FSPL_dB = FSPL(t_index);
        FSPL = 20 * log10(4 * pi * d * f_carrier / c);
        current_size = current_size + 1;
```

```
        % FSPL_result = FSPL(transmitted_data, satellite_parameters, 
ground_station_parameters);
    end


 % Trim FSPL_dB to the actual size
 FSPL_dB = FSPL(1:current_size);
 % fprintf('FSPL (dB) = %f dB\n', FSPL_dB);  % Debug output
end
```

APPENDIX F: Doppler_Shift.m

```
function Doppler_Shift = Doppler_Shift(transmitted_data,
satellite_parameters, satellite_data, ground_station_parameters)
   % Define variables
    SNR = transmitted_data.SNR;
    BW = transmitted_data.BW;
    SF = 7;

    phase = angle(transmitted_data.signal);
    f_carrier = (diff(phase) / (2 * pi * ((2^SF)/BW))*1000); %
Carrier frequency in Hz
    c = satellite_parameters.speed_of_light_kmps * 1000; % Speed of
light in m/s

    % Define time span and time step
    start_time = 0;            % Start time in seconds
    end_time = 10;             % End time in seconds
    time_step = 0.01;          % Time step in seconds
    time_seconds = start_time:time_step:end_time;

    % Preallocate Doppler shift and distance arrays
    doppler_shift_Hz = zeros(size(time_seconds));
    distance_km = zeros(size(time_seconds));

    % Assuming constant velocity
    v_satellite = 100;  % Constant velocity of the satellite in m/s
    v_ground_station = 50;  % Constant velocity of the ground
station in m/s

    % Latitude and longitude coordinates of ground station
    latitude_ground_station =
ground_station_parameters.GroundStationLocation(1);
    longitude_ground_station =
ground_station_parameters.GroundStationLocation(2);

    % Full Calculation of Haversine formula
    function dist = haversine(position_satellite,
position_ground_station)
        R = 6371e3;  % Radius of the Earth in meters

        % Convert latitude and longitude from degrees to radians
        lat1 = deg2rad(position_satellite(1));
        lon1 = deg2rad(position_satellite(2));
        lat2 = deg2rad(position_ground_station(1));
        lon2 = deg2rad(position_ground_station(2));

        % Haversine formula
        dlat = lat2 - lat1;
        dlon = lon2 - lon1;
        a = sin(dlat/2)^2 + cos(lat1) * cos(lat2) * sin(dlon/2)^2;
        C = 2 * atan2(sqrt(a), sqrt(1-a));
        dist = R * C;  % Great-circle distance in meters
    end

    % Iterate over time steps and align with SNR values
```

```
    for t_index = 1:length(SNR)
        % Calculate the distance between satellite and ground
station
        position_satellite = [satellite_data.Latitude_deg(t_index),
satellite_data.Longitude_deg(t_index)];
        position_ground_station = [latitude_ground_station,
longitude_ground_station];
        distance = haversine(position_satellite,
position_ground_station);
        distance_km(t_index) = distance / 1000;  % Convert to
kilometers

        % Calculate the relative velocity between satellite and
ground station
        v_relative = v_satellite - v_ground_station;

        % Calculate Doppler shift for the current time step and SNR
        doppler_shift_Hz = f_carrier * v_relative / c;

        % Debug outputs
        fprintf('SNR = %.2f dB\n', SNR(t_index));
        fprintf('Doppler Shift = %.4f Hz\n',
doppler_shift_Hz(t_index));
        fprintf('Distance Sat & GS = %.2f km\n',
distance_km(t_index));
    end

    % Store the Doppler shift and distance arrays in the output
struct
    Doppler_Shift.doppler_shift_Hz = doppler_shift_Hz;
    Doppler_Shift.distance_km = distance_km;

end
```

APPENDIX G: generate_txt_report.m

```
  % Rest of the functions (ground_station_simulation, BERLoRa,
propagation_delay, propagation_delay, sim_analysis_results)
    function generate_txt_report(transmitted_data, satellite_data,
satellite_parameters, ground_station_parameters, Doppler_Result,
propagation_delay_sec, FSPL_dB, CtrSNR, SF, SNR, BER, SER, PER)
    % Create a timestamp for the report
    timestamp = datestr(now, 'yyyy-mm-dd_HH-MM-SS');

    % Define the report file name
    report_filename = [' C:\Users\Default\Desktop
\SimulationReports\SimulationReport_', timestamp, '.txt'];

    % Open the report file for writing
    report_file = fopen(report_filename, 'w');

    % Output simulation parameters to the report file
    fprintf(report_file, 'Simulation Report - %s\n', timestamp);
    fprintf(report_file,
'________________________________________\n\n');

    % Output ground station parameters
    fprintf(report_file, '|     Ground Station Parameters     |\n');
    fprintf(report_file, '|_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _|\n');
    fprintf(report_file, 'Antenna Height (m): %.2f\n',
ground_station_parameters.Height_m);                        %
Ground station height in meters
    fprintf(report_file, 'Antenna Gain (dBi): %.2f\n',
ground_station_parameters.AntennaGain_dBi);                 %
Antenna Gain in dBi
    fprintf(report_file, 'Transmitter Power (dBm): %.2f\n',
ground_station_parameters.TransmitterPower_dBm);        %
Transmitter power in dBm
    fprintf(report_file, 'Receiver Sensitivity (dBm): %.2f\n',
ground_station_parameters.ReceiverSensitivity_dBm);  % Receiver
sensitivity in dBm
    fprintf(report_file, '           ^          ');
    fprintf(report_file, '\nGround Station Position(deg):\n');
    fprintf(report_file, 'Latitude %.4f, \nLongitude %.4f\n',
ground_station_parameters.GroundStationLocation); % Ground Station
Antenna gain

    % Output satellite parameters
    fprintf(report_file, '\n|       Satellite Parameters
|\n');
    fprintf(report_file, '|_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _|\n');
    fprintf(report_file, 'Link Frequency (GHz): %.2f\n',
satellite_parameters.LinkFrequency_GHz);   % Communication link
frequency in GHz
    fprintf(report_file, 'Transmit Power (dBm): %.2f\n',
satellite_parameters.PowerTransmitted_dBm);% Transmit power in dBm
    fprintf(report_file, 'Antenna Gain (dBi): %.2f\n',
satellite_parameters.AntennaGain_dBi);      % Satellite antenna
gain in dBi
```

```
    fprintf(report_file, 'Orbit Altitude (km): %.2f\n',
satellite_parameters.OrbitAltitude_km);     % Satellite orbit
altitude in kilometers
    fprintf(report_file, 'Orbit Inclination (deg): %.2f\n',
satellite_parameters.Inclination_deg);  % Satellite orbit
inclination in degrees
    fprintf(report_file, 'Orbit Eccentricity: %.2f\n',
satellite_parameters.eccentricity);          % Satellite orbit
eccentricity
    fprintf(report_file, 'Orbit Period (min): %.2f\n',
satellite_data.OrbitPeriod_min);             % Satellite orbit
period in minutes

    % Satellite starting position
    fprintf(report_file, '           ^          ');
    fprintf(report_file, '\nInitial Satellite Position:\n');
    fprintf(report_file, 'Latitude (deg): %.4f\n',
satellite_data.Latitude_deg(1));     % Initial satellite latitude in
degrees
    fprintf(report_file, 'Longitude (deg): %.4f\n',
satellite_data.Longitude_deg(1));   % Initial satellite longitude in
degrees

    % Convert results to percentages
    BER_percentage = 100 * BER;
    SER_percentage = 100 * SER;
    PER_percentage = 100 * SER;

    % Output simulation results
    fprintf(report_file, '\n| Simulation Parameters - Performance
Metrics |\n');
    fprintf(report_file, '|_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _|\n');
    fprintf(report_file, 'Spreading Factor: %d\n', SF);
     fprintf(report_file, 'SNR Range (dB): %.2f : %.2f\n', SNR(1),
SNR(CtrSNR));
    fprintf(report_file, 'Initial BER: %.6f > (%.4f%%)\n', BER(1),
BER_percentage(1));
    fprintf(report_file, 'Initial SER: %.6f > (%.4f%%)\n', SER(1),
SER_percentage(1));
    fprintf(report_file, 'Initial PER: %.6f > (%.4f%%)\n', PER(1),
PER_percentage(1));
    fprintf(report_file, '\n- - - - - - - - - - - - - - - - - - - -
-');
    fprintf(report_file, '\n          Simulation Results       \n');
    fprintf(report_file, '- - - - - - - - - - - - - - - - - - - - -
');

    % Output satellite positions
    fprintf(report_file, '\n\n       Telemetry Per Time Step\n\n');
    for CtrSNR = 1 : length(SNR)
        % Print longitude and latitude at each time step

        fprintf(report_file, '                  __           ');
        fprintf(report_file, '\n             Time Step %d\n',
CtrSNR);    % Time step
```

```
        fprintf(report_file,
'________________________________________\n\n');
        fprintf(report_file, 'Satellite Latitude (deg): %.4f\n',
satellite_data.Latitude_deg(CtrSNR));    % Output sat latitude in
degrees
        fprintf(report_file, 'Satellite Longitude (deg): %.4f\n',
satellite_data.Longitude_deg(CtrSNR));  % Output sat longitude in
degrees
        fprintf(report_file, 'Distance of Sat <> GS: %.4f km\n',
Doppler_Result.distance_km(CtrSNR));   % Distance between Sat and GS
        fprintf(report_file, 'BER: %.6f > (%.4f%%)\n', BER(CtrSNR),
BER_percentage(CtrSNR));
        fprintf(report_file, 'SER: %.6f > (%.4f%%)\n', SER(CtrSNR),
SER_percentage(CtrSNR));
        fprintf(report_file, 'PER: %.6f > (%.4f%%)\n', PER(CtrSNR),
PER_percentage(CtrSNR));
        fprintf(report_file, 'SNR (dB): %.2f\n', SNR(CtrSNR));   %
SNR in dB
        fprintf(report_file, 'FSPL (dB) = %f dB\n',
FSPL_dB(CtrSNR));
        fprintf(report_file, 'Doppler Shift %.4f Hz\n',
Doppler_Result.doppler_shift_Hz(CtrSNR));   % Distance between Sat
and GS
        fprintf(report_file, 'Propagation Delay (seconds): %.6f\n',
propagation_delay_sec(CtrSNR));     % Output propagation delay
        % % Print QoS metrics
        fprintf(report_file, '\n| QoS Metrics |\n');
        fprintf(report_file,'Packet Delivery Ratio: %.8f \n',
transmitted_data.PDR(CtrSNR));
        fprintf(report_file,'Average Packet Delay: %.8f seconds\n',
transmitted_data.avg_packet_delay);
        fprintf(report_file,'Average Throughput: %.8f bps\n',
transmitted_data.avg_throughput(CtrSNR));
        fprintf(report_file,'Energy Efficiency: %.8f bits/Joule\n',
transmitted_data.energy_efficiency(CtrSNR));
        fprintf(report_file,'Spectral Efficiency: %.8f bps/Hz\n',
transmitted_data.spectral_efficiency(CtrSNR));
        fprintf(report_file,'Link Reliability: %.8f \n',
transmitted_data.link_reliability(CtrSNR));
        fprintf(report_file,'Jitter: %.8f \n',
transmitted_data.jitter);

%         fprintf(report_file, 'CtrSNR: %.2f\n', CtrSNR(i));     %
Output CtrSNR

        fprintf(report_file, '\n'); % Space between increments
    end

    % Close the report file
    fclose(report_file);

    % Display a message indicating where the report is saved
    fprintf('Simulation report saved to: %s\n', report_filename);

    end
```

APPENDIX H: generate_csv_report.m

```
function     generate_csv_report(transmitted_data, satellite_data,
satellite_parameters, ground_station_parameters, Doppler_Result,
propagation_delay_sec, FSPL_dB, CtrSNR, SF, SNR, BER, SER, PER)
    % Create a timestamp for the report
    timestamp = datestr(now, 'yyyy-mm-dd_HH-MM-SS');

    % Define the report file name
    report_filename =
['C:\Users\Default\Desktop\SimulationReports\SimulationReport_',
timestamp, '.csv'];

    % Open the report file for writing
    report_file = fopen(report_filename, 'w');

    % Output simulation parameters to the report file
    fprintf(report_file, 'Simulation Report,%s\n\n', timestamp);

    % Output ground station parameters
    fprintf(report_file, 'Ground Station Parameters,,\n');
    fprintf(report_file, 'Parameter,Value\n');
    fprintf(report_file, 'Antenna Height (m),%.2f\n',
ground_station_parameters.Height_m);
    fprintf(report_file, 'Antenna Gain (dBi),%.2f\n',
ground_station_parameters.AntennaGain_dBi);
    fprintf(report_file, 'Transmitter Power (dBm),%.2f\n',
ground_station_parameters.TransmitterPower_dBm);
    fprintf(report_file, 'Receiver Sensitivity (dBm),%.2f\n',
ground_station_parameters.ReceiverSensitivity_dBm);
    fprintf(report_file, 'Ground Station Position (deg),Latitude
%.4f, Longitude %.4f\n',
ground_station_parameters.GroundStationLocation);

    % Output satellite parameters
    fprintf(report_file, '\nSatellite Parameters,,\n');
    fprintf(report_file, 'Parameter,Value\n');
    fprintf(report_file, 'Link Frequency (GHz),%.2f\n',
satellite_parameters.LinkFrequency_GHz);
    fprintf(report_file, 'Transmit Power (dBm),%.2f\n',
satellite_parameters.PowerTransmitted_dBm);
    fprintf(report_file, 'Antenna Gain (dBi),%.2f\n',
satellite_parameters.AntennaGain_dBi);
    fprintf(report_file, 'Orbit Altitude (km),%.2f\n',
satellite_parameters.OrbitAltitude_km);
    fprintf(report_file, 'Orbit Inclination (deg),%.2f\n',
satellite_parameters.Inclination_deg);
    fprintf(report_file, 'Orbit Eccentricity,%.2f\n',
satellite_parameters.eccentricity);
    fprintf(report_file, 'Orbit Period (min),%.2f\n',
satellite_data.OrbitPeriod_min);

    % Output satellite starting position
    fprintf(report_file, '\nInitial Satellite Position,,\n');
    fprintf(report_file, 'Parameter,Value\n');
    fprintf(report_file, 'Latitude (deg),%.4f\n',
satellite_data.Latitude_deg(1));
```

```
    fprintf(report_file, 'Longitude (deg),%.4f\n',
satellite_data.Longitude_deg(1));

    % Convert results to percentages
    BER_percentage = 100 * BER;
    SER_percentage = 100 * SER;
    PER_percentage = 100 * SER;

    % Output simulation results
    fprintf(report_file, '\nSimulation Results,,\n');
    fprintf(report_file, 'Parameter,Value\n');
    fprintf(report_file, 'Spreading Factor,%d\n', SF);
    fprintf(report_file, 'SNR Range (dB),%.2f :,%.2f\n', SNR(1),
SNR(CtrSNR));
    fprintf(report_file, 'Initial BER: ,%.6f  >, (%.4f%%)\n',
BER(CtrSNR), BER_percentage(CtrSNR));
    fprintf(report_file, 'Initial SER: ,%.6f  >, (%.4f%%)\n',
SER(CtrSNR), SER_percentage(CtrSNR));
    fprintf(report_file, 'Initial PER: ,%.6f  >, (%.4f%%)\n',
PER(CtrSNR), PER_percentage(CtrSNR));


    % Output satellite positions
    fprintf(report_file, '\nSatellite Positions (km),,,\n');
    fprintf(report_file, 'Time Step,Latitude (deg),Longitude
(deg),SNR (dB),Distance of Sat <> GS (km),BER,BER (%%),SER,SER
(%%),PER,PER (%%),FSPL (dB),Doppler Shift,Propagation Delay
(seconds),Packet Delivery Ratio,Average Packet Delay,Average
Throughput,Energy Efficiency,Spectral Efficiency,Link
Reliability,Jitter\n');

    for CtrSNR = 1 : length(SNR)
        fprintf(report_file,
'%d,%.4f,%.4f,%.2f,%.4f,%.6f,%.4f%%,%.6f,%.4f%%
,%.6f,%.4f%%,%f,%.4f,%.6f,%.8f,%.8f,%.8f,%.8f,%.8f,%.8f,%.8f\n',
CtrSNR, satellite_data.Latitude_deg(CtrSNR),
satellite_data.Longitude_deg(CtrSNR), SNR(CtrSNR),
Doppler_Result.distance_km(CtrSNR),BER(CtrSNR),
BER_percentage(CtrSNR), SER(CtrSNR), SER_percentage(CtrSNR)
,PER(CtrSNR), PER_percentage(CtrSNR), FSPL_dB(CtrSNR),
Doppler_Result.doppler_shift_Hz(CtrSNR),
propagation_delay_sec(CtrSNR), transmitted_data.PDR(CtrSNR),
transmitted_data.avg_packet_delay,
transmitted_data.avg_throughput(CtrSNR),
transmitted_data.energy_efficiency(CtrSNR),
transmitted_data.spectral_efficiency(CtrSNR),
transmitted_data.link_reliability(CtrSNR), transmitted_data.jitter);
    end

    % Close the report file
    fclose(report_file);

    % Display a message indicating where the report is saved
    fprintf('Simulation report saved to: %s\n', report_filename);
end
```

APPENDIX I: main.m

```
% Main function
% This main function serves to pass variables and numericalvalues
between functions
% Once triggered, the main fucntion executes all of the other
related functions
% After all simulations are complete, the results are printed to a
.txt and .csv file

%% Main Function with definitions
function main
    % Define Satellite parameters to be called by
satellite_simulation script
    satellite_parameters = struct();
    satellite_parameters.OrbitAltitude_km = 800;    % Satellite
orbit altitude in kilometers
    satellite_parameters.LinkFrequency_GHz = 2.4;   % Communication
link frequency in GHz
    satellite_parameters.PowerTransmitted_dBm = 30; % Transmit power
in dBm
    satellite_parameters.AntennaGain_dBi = 10;      % Satellite
antenna gain in dBi
    satellite_parameters.Inclination_deg = 45;      % Satellite
orbit inclination in degrees
    satellite_parameters.eccentricity = 0.2;         % Satellite
Eccentricity from 0 to 1
    satellite_parameters.speed_of_light_kmps = 299792;  % Speed of
light in kilometres per second
    % satellite_parameters.OrbitPeriod_min = 90;      % Satellite
orbit period in minutes (Calculated in satellite_simulation)

    % Define Ground station parameters
    ground_station_parameters = struct();
    ground_station_parameters.GroundStationLocation = [-33.9034,
18.6246];  % Ground station location (latitude and longitude)
    ground_station_parameters.AntennaGain_dBi = 20;             %
Ground station antenna gain in dBi
    ground_station_parameters.Height_m = 15;                    %
Ground station height in meters
    ground_station_parameters.TransmitterPower_dBm = 25;        %
Transmitter power in dBm
    ground_station_parameters.ReceiverSensitivity_dBm = -110;   %
Receiver sensitivity in dBm

    % Simulate satellite constellation and assign the function
locally
    satellite_data = satellite_simulation(satellite_parameters,
ground_station_parameters);

    % Pass both satellite_data and ground_station_parameters to
BERLoRa to simulate LoRaWAN communication
    % At the same time, Extract relevant data processing and
analysis from the BERLoRa function
    [transmitted_data, SF, CtrSNR, SNR] =
BERLoRa(satellite_parameters, satellite_data,
ground_station_parameters);
```

```
    % Re-assign desired varaibles from BERLoRa(transmitted_data)
    BER = transmitted_data.BER.nCOH_AWGN; % Adjust this based on the
specific BER value you want
    SER = transmitted_data.SER.nCOH_AWGN; % Similarly, adjust this
for SER
    PER = transmitted_data.PER.nCOH_AWGN; % And this for PER
    % SNR = transmitted_data.SNR;
    % SF = transmitted_data.SF;
    CtrSNR = transmitted_data.CtrSNR;

    % Trigger Doppler shift calculation
    % Doppler_Shift = transmitted_data.Doppler_Shift
    Doppler_Result = Doppler_Shift(transmitted_data,
satellite_parameters, satellite_data, ground_station_parameters);

    % Call Free Space Path Loss function
    FSPL_dB = FSPL(transmitted_data, satellite_data,
satellite_parameters, ground_station_parameters);

    % Simulate and calculate propagation delay
    propagation_delay_sec =
propagation_delay(ground_station_parameters, satellite_data,
satellite_parameters);

    % Generate txt and csv reports of the outputs
    generate_txt_report(transmitted_data, satellite_data,
satellite_parameters, ground_station_parameters, Doppler_Result,
propagation_delay_sec, FSPL_dB, CtrSNR, SF, SNR, BER, SER, PER);
    generate_word_report(transmitted_data, satellite_data,
satellite_parameters, ground_station_parameters, Doppler_Result,
propagation_delay_sec, FSPL_dB, CtrSNR, SF, SNR, BER, SER, PER);
    generate_csv_report(transmitted_data, satellite_data,
satellite_parameters, ground_station_parameters, Doppler_Result,
propagation_delay_sec, FSPL_dB, CtrSNR, SF, SNR, BER, SER, PER);
end
```

APPENDIX J: Simulation Report Sample

Simulation Report - 2023-10-16_22-33-51

| Ground Station Parameters |

|_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _|

Antenna Height (m): 15.00

Antenna Gain (dBi): 20.00

Transmitter Power (dBm): 25.00

Receiver Sensitivity (dBm): -110.00

^

Ground Station Position(deg):

Latitude -33.9034,

Longitude 18.6246

| Satellite Parameters |

|_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _|

Link Frequency (GHz): 2.40

Transmit Power (dBm): 30.00

Antenna Gain (dBi): 10.00

Orbit Altitude (km): 800.00

Orbit Inclination (deg): 45.00

Orbit Eccentricity: 0.20

Orbit Period (min): 100.72

^

Initial Satellite Position:

Latitude (deg): 0.0421

Longitude (deg): 45.0596

| Simulation Parameters - Performance Metrics |

|_ _|

Spreading Factor: 7

SNR Range (dB): -30.00 : 30.00

Initial BER: 0.497710 > (49.7710%)

Initial SER: 0.987700 > (98.7700%)

Initial PER: 1.000000 > (98.7700%)

- -

Simulation Results

- -

Telemetry Per Time Step

—

Time Step 1

Satellite Latitude (deg): 0.0421

Satellite Longitude (deg): 45.0596

Distance of Sat <> GS: 4673.5978 km

BER: 0.497710 > (49.7710%)

SER: 0.987700 > (98.7700%)

PER: 1.000000 > (98.7700%)

SNR (dB): -30.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015589

| QoS Metrics |

Packet Delivery Ratio: 0.00723333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 2

Satellite Latitude (deg): 0.0842

Satellite Longitude (deg): 45.1191

Distance of Sat <> GS: 4681.1598 km

BER: 0.497433 > (49.7433%)

SER: 0.986767 > (98.6767%)

PER: 1.000000 > (98.6767%)

SNR (dB): -29.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015615

| QoS Metrics |

Packet Delivery Ratio: 0.00870000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 3

Satellite Latitude (deg): 0.1264

Satellite Longitude (deg): 45.1787

Distance of Sat <> GS: 4688.7233 km

BER: 0.495190 > (49.5190%)

SER: 0.983000 > (98.3000%)

PER: 1.000000 > (98.3000%)

SNR (dB): -28.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015640

| QoS Metrics |

Packet Delivery Ratio: 0.00793333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 4

Satellite Latitude (deg): 0.1685

Satellite Longitude (deg): 45.2383

Distance of Sat <> GS: 4696.2883 km

BER: 0.494129 > (49.4129%)

SER: 0.982100 > (98.2100%)

PER: 1.000000 > (98.2100%)

SNR (dB): -27.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015665

| QoS Metrics |

Packet Delivery Ratio: 0.00806667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 5

Satellite Latitude (deg): 0.2106

Satellite Longitude (deg): 45.2978

Distance of Sat <> GS: 4703.8547 km

BER: 0.493205 > (49.3205%)

SER: 0.978700 > (97.8700%)

PER: 1.000000 > (97.8700%)

SNR (dB): -26.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015690

| QoS Metrics |

Packet Delivery Ratio: 0.00736667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 6

Satellite Latitude (deg): 0.2527

Satellite Longitude (deg): 45.3574

Distance of Sat <> GS: 4711.4227 km

BER: 0.490414 > (49.0414%)

SER: 0.972467 > (97.2467%)

PER: 1.000000 > (97.2467%)

SNR (dB): -25.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015716

| QoS Metrics |

Packet Delivery Ratio: 0.00710000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 7

Satellite Latitude (deg): 0.2948

Satellite Longitude (deg): 45.4170

Distance of Sat <> GS: 4718.9921 km

BER: 0.489757 > (48.9757%)

SER: 0.969233 > (96.9233%)

PER: 1.000000 > (96.9233%)

SNR (dB): -24.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015741

| QoS Metrics |

Packet Delivery Ratio: 0.00783333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 8

Satellite Latitude (deg): 0.3370

Satellite Longitude (deg): 45.4765

Distance of Sat <> GS: 4726.5629 km

BER: 0.484529 > (48.4529%)

SER: 0.960700 > (96.0700%)

PER: 1.000000 > (96.0700%)

SNR (dB): -23.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015766

| QoS Metrics |

Packet Delivery Ratio: 0.00813333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 9

Satellite Latitude (deg): 0.3791

Satellite Longitude (deg): 45.5361

Distance of Sat <> GS: 4734.1352 km

BER: 0.478686 > (47.8686%)

SER: 0.949233 > (94.9233%)

PER: 1.000000 > (94.9233%)

SNR (dB): -22.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015791

| QoS Metrics |

Packet Delivery Ratio: 0.00786667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 10

Satellite Latitude (deg): 0.4212

Satellite Longitude (deg): 45.5957

Distance of Sat <> GS: 4741.7089 km

BER: 0.471305 > (47.1305%)

SER: 0.934533 > (93.4533%)

PER: 1.000000 > (93.4533%)

SNR (dB): -21.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015817

| QoS Metrics |

Packet Delivery Ratio: 0.00813333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 11

Satellite Latitude (deg): 0.4633

Satellite Longitude (deg): 45.6553

Distance of Sat <> GS: 4749.2840 km

BER: 0.461971 > (46.1971%)

SER: 0.915900 > (91.5900%)

PER: 1.000000 > (91.5900%)

SNR (dB): -20.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015842

| QoS Metrics |

Packet Delivery Ratio: 0.00813333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 12

Satellite Latitude (deg): 0.5054

Satellite Longitude (deg): 45.7148

Distance of Sat <> GS: 4756.8605 km

BER: 0.445971 > (44.5971%)

SER: 0.884033 > (88.4033%)

PER: 1.000000 > (88.4033%)

SNR (dB): -19.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015867

| QoS Metrics |

Packet Delivery Ratio: 0.00803333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 13

Satellite Latitude (deg): 0.5476

Satellite Longitude (deg): 45.7744

Distance of Sat <> GS: 4764.4384 km

BER: 0.422533 > (42.2533%)

SER: 0.839067 > (83.9067%)

PER: 1.000000 > (83.9067%)

SNR (dB): -18.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015892

| QoS Metrics |

Packet Delivery Ratio: 0.00800000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 14

Satellite Latitude (deg): 0.5897

Satellite Longitude (deg): 45.8340

Distance of Sat <> GS: 4772.0177 km

BER: 0.394362 > (39.4362%)

SER: 0.781300 > (78.1300%)

PER: 1.000000 > (78.1300%)

SNR (dB): -17.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015918

| QoS Metrics |

Packet Delivery Ratio: 0.00820000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 15

Satellite Latitude (deg): 0.6318

Satellite Longitude (deg): 45.8935

Distance of Sat <> GS: 4779.5984 km

BER: 0.350219 > (35.0219%)

SER: 0.696633 > (69.6633%)

PER: 1.000000 > (69.6633%)

SNR (dB): -16.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015943

| QoS Metrics |

Packet Delivery Ratio: 0.00813333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 16

Satellite Latitude (deg): 0.6739

Satellite Longitude (deg): 45.9531

Distance of Sat <> GS: 4787.1805 km

BER: 0.299990 > (29.9990%)

SER: 0.596533 > (59.6533%)

PER: 1.000000 > (59.6533%)

SNR (dB): -15.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015968

| QoS Metrics |

Packet Delivery Ratio: 0.00806667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 17

Satellite Latitude (deg): 0.7160

Satellite Longitude (deg): 46.0127

Distance of Sat <> GS: 4794.7640 km

BER: 0.236048 > (23.6048%)

SER: 0.468433 > (46.8433%)

PER: 0.998333 > (46.8433%)

SNR (dB): -14.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.015994

| QoS Metrics |

Packet Delivery Ratio: 0.00710000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 18

Satellite Latitude (deg): 0.7582

Satellite Longitude (deg): 46.0722

Distance of Sat <> GS: 4802.3487 km

BER: 0.166924 > (16.6924%)

SER: 0.331100 > (33.1100%)

PER: 0.980333 > (33.1100%)

SNR (dB): -13.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016019

| QoS Metrics |

Packet Delivery Ratio: 0.00693333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 19

Satellite Latitude (deg): 0.8003

Satellite Longitude (deg): 46.1318

Distance of Sat <> GS: 4809.9349 km

BER: 0.103343 > (10.3343%)

SER: 0.205533 > (20.5533%)

PER: 0.892000 > (20.5533%)

SNR (dB): -12.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016044

| QoS Metrics |

Packet Delivery Ratio: 0.00730000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 20

Satellite Latitude (deg): 0.8424

Satellite Longitude (deg): 46.1914

Distance of Sat <> GS: 4817.5224 km

BER: 0.051014 > (5.1014%)

SER: 0.101967 > (10.1967%)

PER: 0.656333 > (10.1967%)

SNR (dB): -11.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016070

| QoS Metrics |

Packet Delivery Ratio: 0.00716667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 21

Satellite Latitude (deg): 0.8845

Satellite Longitude (deg): 46.2509

Distance of Sat <> GS: 4825.1112 km

BER: 0.018714 > (1.8714%)

SER: 0.037233 > (3.7233%)

PER: 0.318333 > (3.7233%)

SNR (dB): -10.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016095

| QoS Metrics |

Packet Delivery Ratio: 0.00823333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 22

Satellite Latitude (deg): 0.9266

Satellite Longitude (deg): 46.3105

Distance of Sat <> GS: 4832.7013 km

BER: 0.004562 > (0.4562%)

SER: 0.009133 > (0.9133%)

PER: 0.086667 > (0.9133%)

SNR (dB): -9.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016120

| QoS Metrics |

Packet Delivery Ratio: 0.00786667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 23

Satellite Latitude (deg): 0.9687

Satellite Longitude (deg): 46.3701

Distance of Sat <> GS: 4840.2927 km

BER: 0.000881 > (0.0881%)

SER: 0.001767 > (0.1767%)

PER: 0.017333 > (0.1767%)

SNR (dB): -8.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016146

| QoS Metrics |

Packet Delivery Ratio: 0.00733333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 24

Satellite Latitude (deg): 1.0109

Satellite Longitude (deg): 46.4296

Distance of Sat <> GS: 4847.8855 km

BER: 0.000033 > (0.0033%)

SER: 0.000067 > (0.0067%)

PER: 0.000667 > (0.0067%)

SNR (dB): -7.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016171

| QoS Metrics |

Packet Delivery Ratio: 0.00680000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 25

Satellite Latitude (deg): 1.0530

Satellite Longitude (deg): 46.4892

Distance of Sat <> GS: 4855.4795 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): -6.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016196

| QoS Metrics |

Packet Delivery Ratio: 0.00833333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 26

Satellite Latitude (deg): 1.0951

Satellite Longitude (deg): 46.5488

Distance of Sat <> GS: 4863.0748 km

BER: 0.000005 > (0.0005%)

SER: 0.000033 > (0.0033%)

PER: 0.000333 > (0.0033%)

SNR (dB): -5.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016221

| QoS Metrics |

Packet Delivery Ratio: 0.00713333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 27

Satellite Latitude (deg): 1.1372

Satellite Longitude (deg): 46.6083

Distance of Sat <> GS: 4870.6714 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): -4.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016247

| QoS Metrics |

Packet Delivery Ratio: 0.00796667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 28

Satellite Latitude (deg): 1.1793

Satellite Longitude (deg): 46.6679

Distance of Sat <> GS: 4878.2693 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): -3.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016272

| QoS Metrics |

Packet Delivery Ratio: 0.00786667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 29

Satellite Latitude (deg): 1.2214

Satellite Longitude (deg): 46.7275

Distance of Sat <> GS: 4885.8684 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): -2.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016298

| QoS Metrics |

Packet Delivery Ratio: 0.00780000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 30

Satellite Latitude (deg): 1.2635

Satellite Longitude (deg): 46.7871

Distance of Sat <> GS: 4893.4688 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): -1.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016323

| QoS Metrics |

Packet Delivery Ratio: 0.00756667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 31

Satellite Latitude (deg): 1.3056

Satellite Longitude (deg): 46.8466

Distance of Sat <> GS: 4901.0704 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 0.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016348

| QoS Metrics |

Packet Delivery Ratio: 0.00776667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 32

Satellite Latitude (deg): 1.3478

Satellite Longitude (deg): 46.9062

Distance of Sat <> GS: 4908.6733 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 1.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016374

| QoS Metrics |

Packet Delivery Ratio: 0.00853333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 33

Satellite Latitude (deg): 1.3899

Satellite Longitude (deg): 46.9658

Distance of Sat <> GS: 4916.2773 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 2.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016399

| QoS Metrics |

Packet Delivery Ratio: 0.00843333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 34

Satellite Latitude (deg): 1.4320

Satellite Longitude (deg): 47.0253

Distance of Sat <> GS: 4923.8826 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 3.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016424

| QoS Metrics |

Packet Delivery Ratio: 0.00766667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 35

Satellite Latitude (deg): 1.4741

Satellite Longitude (deg): 47.0849

Distance of Sat <> GS: 4931.4891 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 4.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016450

| QoS Metrics |

Packet Delivery Ratio: 0.00736667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 36

Satellite Latitude (deg): 1.5162

Satellite Longitude (deg): 47.1445

Distance of Sat <> GS: 4939.0968 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 5.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016475

| QoS Metrics |

Packet Delivery Ratio: 0.00826667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 37

Satellite Latitude (deg): 1.5583

Satellite Longitude (deg): 47.2040

Distance of Sat <> GS: 4946.7057 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 6.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016500

| QoS Metrics |

Packet Delivery Ratio: 0.00800000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 38

Satellite Latitude (deg): 1.6004

Satellite Longitude (deg): 47.2636

Distance of Sat <> GS: 4954.3158 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 7.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016526

| QoS Metrics |

Packet Delivery Ratio: 0.00793333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 39

Satellite Latitude (deg): 1.6425

Satellite Longitude (deg): 47.3232

Distance of Sat <> GS: 4961.9270 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 8.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016551

| QoS Metrics |

Packet Delivery Ratio: 0.00773333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 40

Satellite Latitude (deg): 1.6846

Satellite Longitude (deg): 47.3827

Distance of Sat <> GS: 4969.5395 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 9.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016577

| QoS Metrics |

Packet Delivery Ratio: 0.00793333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 41

Satellite Latitude (deg): 1.7267

Satellite Longitude (deg): 47.4423

Distance of Sat <> GS: 4977.1530 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 10.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016602

| QoS Metrics |

Packet Delivery Ratio: 0.00803333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 42

Satellite Latitude (deg): 1.7688

Satellite Longitude (deg): 47.5019

Distance of Sat <> GS: 4984.7677 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 11.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016627

| QoS Metrics |

Packet Delivery Ratio: 0.00906667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 43

Satellite Latitude (deg): 1.8109

Satellite Longitude (deg): 47.5614

Distance of Sat <> GS: 4992.3836 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 12.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016653

| QoS Metrics |

Packet Delivery Ratio: 0.00813333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 44

Satellite Latitude (deg): 1.8530

Satellite Longitude (deg): 47.6210

Distance of Sat <> GS: 5000.0006 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 13.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016678

| QoS Metrics |

Packet Delivery Ratio: 0.00733333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 45

Satellite Latitude (deg): 1.8951

Satellite Longitude (deg): 47.6806

Distance of Sat <> GS: 5007.6187 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 14.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016704

| QoS Metrics |

Packet Delivery Ratio: 0.00873333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 46

Satellite Latitude (deg): 1.9372

Satellite Longitude (deg): 47.7401

Distance of Sat <> GS: 5015.2380 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 15.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016729

| QoS Metrics |

Packet Delivery Ratio: 0.00860000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 47

Satellite Latitude (deg): 1.9793

Satellite Longitude (deg): 47.7997

Distance of Sat <> GS: 5022.8583 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 16.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016754

| QoS Metrics |

Packet Delivery Ratio: 0.00783333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 48

Satellite Latitude (deg): 2.0214

Satellite Longitude (deg): 47.8593

Distance of Sat <> GS: 5030.4797 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 17.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016780

| QoS Metrics |

Packet Delivery Ratio: 0.00736667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 49

Satellite Latitude (deg): 2.0635

Satellite Longitude (deg): 47.9189

Distance of Sat <> GS: 5038.1023 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 18.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016805

| QoS Metrics |

Packet Delivery Ratio: 0.00820000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 50

Satellite Latitude (deg): 2.1056

Satellite Longitude (deg): 47.9784

Distance of Sat <> GS: 5045.7259 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 19.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016831

| QoS Metrics |

Packet Delivery Ratio: 0.00746667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 51

Satellite Latitude (deg): 2.1477

Satellite Longitude (deg): 48.0380

Distance of Sat <> GS: 5053.3506 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 20.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016856

| QoS Metrics |

Packet Delivery Ratio: 0.00690000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 52

Satellite Latitude (deg): 2.1898

Satellite Longitude (deg): 48.0976

Distance of Sat <> GS: 5060.9763 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 21.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016882

| QoS Metrics |

Packet Delivery Ratio: 0.00773333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 53

Satellite Latitude (deg): 2.2319

Satellite Longitude (deg): 48.1571

Distance of Sat <> GS: 5068.6032 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 22.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016907

| QoS Metrics |

Packet Delivery Ratio: 0.00623333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 54

Satellite Latitude (deg): 2.2739

Satellite Longitude (deg): 48.2167

Distance of Sat <> GS: 5076.2310 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 23.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016933

| QoS Metrics |

Packet Delivery Ratio: 0.00840000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 55

Satellite Latitude (deg): 2.3160

Satellite Longitude (deg): 48.2763

Distance of Sat <> GS: 5083.8599 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 24.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016958

| QoS Metrics |

Packet Delivery Ratio: 0.00713333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 56

Satellite Latitude (deg): 2.3581

Satellite Longitude (deg): 48.3358

Distance of Sat <> GS: 5091.4899 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 25.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.016983

| QoS Metrics |

Packet Delivery Ratio: 0.00746667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 57

Satellite Latitude (deg): 2.4002

Satellite Longitude (deg): 48.3954

Distance of Sat <> GS: 5099.1209 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 26.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.017009

| QoS Metrics |

Packet Delivery Ratio: 0.00766667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 58

Satellite Latitude (deg): 2.4423

Satellite Longitude (deg): 48.4550

Distance of Sat <> GS: 5106.7528 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 27.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.017034

| QoS Metrics |

Packet Delivery Ratio: 0.00756667

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 59

Satellite Latitude (deg): 2.4844

Satellite Longitude (deg): 48.5145

Distance of Sat <> GS: 5114.3859 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 28.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.017060

| QoS Metrics |

Packet Delivery Ratio: 0.00760000

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 60

Satellite Latitude (deg): 2.5265

Satellite Longitude (deg): 48.5741

Distance of Sat <> GS: 5122.0199 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 29.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.017085

| QoS Metrics |

Packet Delivery Ratio: 0.00783333

Average Packet Delay: NaN seconds

Average Throughput: 0.00000000 bps

Energy Efficiency: 0.00000000 bits/Joule

Spectral Efficiency: 0.00000000 bps/Hz

Link Reliability: 0.00000000

Jitter: NaN

—

Time Step 61

Satellite Latitude (deg): 2.5685

Satellite Longitude (deg): 48.6337

Distance of Sat <> GS: 5129.6549 km

BER: 0.000000 > (0.0000%)

SER: 0.000000 > (0.0000%)

PER: 0.000000 > (0.0000%)

SNR (dB): 30.00

FSPL (dB) = 237.047831 dB

Doppler Shift 0.0013 Hz

Propagation Delay (seconds): 0.017111

| QoS Metrics |

Packet Delivery Ratio: 0.00826667

Average Packet Delay: NaN seconds

Average Throughput: 0.00007813 bps

Energy Efficiency: 1033333.33333333 bits/Joule

Spectral Efficiency: 7233.33333333 bps/Hz

Link Reliability: 0.00006667

Jitter: NaN