

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Dipartimento d'Informatica - Scienza e Ingegneria

Corso di Laurea in Ingegneria e Scienze Informatiche

# Sviluppo e Ottimizzazione di un Sistema di Telemetria per Razzi

## Architettura e Trasmissione Dati con Tecnologia LoRa©

**Relatore:**
**Chiar.mo Prof.**
**Andrea Piroddi**

**Presentata da:**
**Alessandro Monticelli**

---

**Sessione luglio 2025**
**Anno Accademico 2025/2026**

# Abstract

I razzi sperimentali sono essenziali per la ricerca aerospaziale e meteorologica, nonché per la validazione di nuove tecnologie in ambienti estremi. Per garantire il monitoraggio in tempo reale e l'analisi post-missione in tale contesto, la disponibilità di sistemi di telemetria affidabili, a bassa latenza e basso costo è di fondamentale importanza.

Questa tesi descrive lo sviluppo di un innovativo sistema di telemetria embedded, basato su un microcontrollore ESP32-S3 e sulla modulazione LoRa©, per il tracciamento dei dati di volo di un razzo sonda sperimentale. Tale sistema garantisce comunicazioni stabili anche su lunghe distanze verso una ground station. Per ovviare all'assenza di meccanismi di ritrasmissione e assicurare robustezza ed efficienza della trasmissione, è stato progettato un protocollo di comunicazione personalizzato, che include segmentazione del payload, header identificativi, codice CRC e gestione dei timeout per la ricostruzione dei dati.

I test in campo aperto hanno convalidato l'efficacia del sistema, evidenziando una portata effettiva superiore ai 2 km, un'alta stabilità del segnale con valori di *RSSI* non inferiori al 60% in campo aperto, e un throughput medio di 640 Byte/s. Queste prestazioni, in linea con i requisiti di missioni suborbitali a bassa quota, dimostrano la validità della soluzione proposta per applicazioni reali non limitate all'ambito aerospaziale, ma estendibili anche ad altri settori.

# Indice


**1 Introduzione** **1**
- 1.1 Presentazione del progetto Borealis di Aurora Rocketry . . . . . 1
- 1.2 Contesto storico della creazione e dell'impiego di razzi sperimentali . 3
- 1.3 Contesto e importanza dei sistemi di telemetria nei razzi sperimentali 3
- 1.4 Obiettivi della tesi e contributo personale . . . . . 4

**2 Descrizione del Computer di Volo** **5**
- 2.1 Funzioni principali e componenti chiave . . . . . 5
- 2.2 Integrazione e ruolo del sistema di telemetria . . . . . 6

**3 Tecnologia LoRa© e la sua Applicazione** **7**
- 3.1 Panoramica della tecnologia LoRa© . . . . . 7
  - 3.1.1 Modulazione *Chirp Spread Spectrum* . . . . . 8
- 3.2 Specifiche tecniche dei moduli utilizzati . . . . . 9
- 3.3 Specifiche delle antenne utilizzate . . . . . 10
- 3.4 Motivazioni della scelta di LoRa© per il progetto . . . . . 12

**4 Implementazione del Sistema di Telemetria** **13**
- 4.1 Architettura del sistema di trasmissione dati . . . . . 13
- 4.2 Protocollo di comunicazione e gestione dati . . . . . 15
- 4.3 Implementazione software e algoritmi di gestione della trasmissione . 16
  - 4.3.1 Architettura software . . . . . 16
  - 4.3.2 Algoritmo di ricostruzione del *payload* . . . . . 21

**5 Valutazione delle Prestazioni** **23**
- 5.1 Metodologia di test e criteri di valutazione . . . . . 24
  - 5.1.1 Test preliminari . . . . . 25
  - 5.1.2 Test sul protocollo di comunicazione . . . . . 26
- 5.2 Ottimizzazioni apportate durante lo sviluppo . . . . . 30
  - 5.2.1 Configurazione personalizzata dei moduli radio . . . . . 30

5.2.2 Compressione dei dati . . . . . 30
5.2.3 Gestione dei pacchetti . . . . . 31

**6 Risultati e Analisi** **33**
6.1 Analisi dei dati raccolti durante le prove sperimentali . . . . . 33
6.1.1 Test preliminari . . . . . 33
6.1.2 Risultati dei test sul protocollo di comunicazione . . . . . 35
6.2 Limiti e possibili miglioramenti futuri . . . . . 49
6.3 Implicazioni del sistema sviluppato per progetti simili . . . . . 49

**7 Conclusioni** **51**
7.1 Riepilogo dei risultati ottenuti . . . . . 51
7.2 Riflessioni sull'impatto del progetto nell'ambito della ricerca e dello sviluppo di razzi sperimentali . . . . . 51
7.3 Prospettive di sviluppo futuro . . . . . 52
7.3.1 Miglioramenti al sistema di telemetria . . . . . 52
7.3.2 Miglioramenti al protocollo di comunicazione . . . . . 53
7.3.3 Parsing dei dati e visualizzazione . . . . . 53

**Appendici**

**Appendice A Librerie**

**Appendice B Datasheet**

**Appendice C Dati di test**

# Elenco delle tabelle


3.1 Caratteristiche tecniche del modulo Long Range (LoRa©) EByte E220-900T22D. . . . . 9
3.2 Specifiche tecniche dell'antenna omnidirezionale e dell'antenna Yagi. . 11

5.1 Schema riassuntivo dei test sperimentali . . . . . 24
5.2 Condizioni del test del throughput. . . . . 26
5.3 Condizioni del test di segmentazione e ricostruzione dei dati. . . . . 28
5.4 Condizioni del test del meccanismo di *drop*. . . . . 29
5.5 Condizioni del test del meccanismo di riordinamento. . . . . 29
5.6 Condizioni del test di robustezza e affidabilità. . . . . 30

6.1 Intensità del segnale (%) in funzione del disallineamento angolare dell'antenna direzionale e della distanza . . . . . 34
6.2 Intensità del segnale (%) in funzione del disallineamento angolare dell'antenna omnidirezionale e della distanza . . . . . 34
6.3 Statistiche sperimentali del throughput in funzione della distanza . . 35
6.4 Tasso di perdita dei pacchetti durante il test del meccanismo di *drop*. 44
6.5 Verifica dell'ordine di arrivo dei chunk per ogni pacchetto . . . . . 45
6.6 Confronto delle metriche statistiche tra test senza interferenze e con interferenze . . . . . 48

# Elenco delle figure


1.1 Schema rappresentativo di *Borealis*. . . . . 1
1.2 *Electronics bay*. . . . . 2

3.1 Up-chirp e down-chirp . . . . . 8
3.2 Grafici di radiazione antenna omnidirezionale . . . . . 11
3.3 Grafici di radiazione dell'antenna Yagi . . . . . 11

4.1 Schema del pacchetto di dati trasmesso dal nodo trasmittente. . . . . 16
4.2 Macchina a stati finiti del computer di volo. . . . . 17
4.3 Interfaccia per la raccolta dati da un sensore. . . . . 18
4.4 Interfaccia per il logging dei dati. . . . . 18
4.5 Interfaccia per la telemetria. . . . . 19
4.6 Architettura della *ground station*. . . . . 20

5.1 Test di disallineamento angolare dell'antenna direzionale. . . . . 25
5.2 Test di disallineamento angolare dell'antenna omnidirezionale. . . . . 26
5.3 Posizioni di test da 10m a 250m con ostacoli. . . . . 27
5.4 Posizione di test a 250m senza ostacoli. . . . . 27
5.5 Posizione di test a 500m. . . . . 27

6.1 Intensità del segnale in funzione della distanza e dell'angolo di disallineamento delle antenne. . . . . 34
6.2 Andamento del *Received Signal Strength Indicator* (RSSI) in funzione della distanza (regressione logaritmica). . . . . 36
6.3 Distribuzione del RSSI per ciascuna distanza testata. . . . . 36
6.4 Throughput in funzione della distanza (distribuzione puntuale). . . . 37
6.5 RSSI e throughput medi in funzione della distanza. . . . . 37
6.6 Tempo medio di ricostruzione per numero di *chunk*. . . . . 38
6.7 Deviazione standard dei tempi di ricostruzione per numero di *chunk*. . 39
6.8 Distribuzione dei tempi di ricostruzione per 2 *chunk* . . . . . 39
6.9 Distribuzione dei tempi di ricostruzione per 3 *chunk* . . . . . 39

6.10 Distribuzione dei tempi di ricostruzione per 4 *chunk* . . . . . 39
6.11 Distribuzione dei tempi di ricostruzione per 5 *chunk* . . . . . 39
6.12 Tempo medio di ricostruzione per numero di *chunk*. . . . . 40
6.13 Deviazione standard dei tempi di ricostruzione per numero di *chunk*. . 40
6.14 Distribuzione dei tempi di ricostruzione per 2 *chunk* . . . . . 41
6.15 Distribuzione dei tempi di ricostruzione per 3 *chunk* . . . . . 41
6.16 Distribuzione dei tempi di ricostruzione per 4 *chunk* . . . . . 41
6.17 Distribuzione dei tempi di ricostruzione per 5 *chunk* . . . . . 41
6.18 Confronto del tempo medio di ricostruzione del payload per numero di *chunk* . . . . . 42
6.19 Differenza percentuale tra i tempi di ricostruzione del payload, configurazione di default vs configurazione personalizzata . . . . . 42
6.20 Confronto delle distribuzioni dei tempi di ricostruzione del payload, configurazione di default vs configurazione personalizzata . . . . . 43
6.21 Stabilità della comunicazione in assenza di interferenza diretta. . . . . 46
6.22 Stabilità della comunicazione in presenza di interferenza diretta. . . . 47
6.23 Confronto tra i due test di robustezza. . . . . 47

# Capitolo 1

# Introduzione

## 1.1 Presentazione del progetto Borealis di Aurora Rocketry

Aurora Rocketry è un'associazione studentesca dell'Università di Bologna nata nel 2024, che si occupa di progettare e costruire razzi sperimentali per partecipare a competizioni internazionali.
Borealis è il primo prototipo di razzo sperimentale sviluppato dall'associazione, progettato per raggiungere un'altitudine di 1500m.

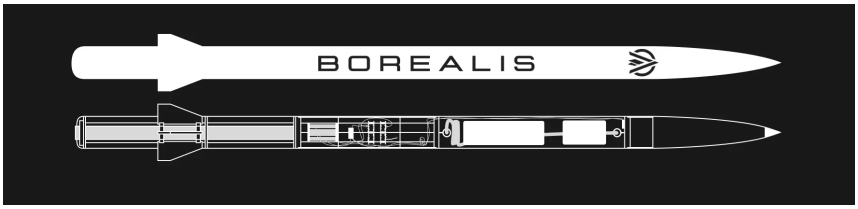



**Figura 1.1:** Schema rappresentativo di *Borealis*.

Il razzo è formato da una sezione propulsiva (*Motor bay*), una fusoliera in alluminio divisa in tre *bays*:

- ***Electronics bay***: contenente l'elettronica di bordo,
- ***Payload bay***: contenente un carico pagante sperimentale,
- ***Parachute tube***: contenente i due paracadute.

Infine un *nose cone* apribile per permettere l'espulsione del paracadute.

La *Electronics bay* é inserita in una finestra in polimetilmetacrilato, per permettere la trasmissione del segnale radio della telemetria, che sarebbe altrimenti schermato dal corpo in alluminio del razzo.

Nella *Electronics bay* è presente il computer di volo, che si occupa di gestire il lancio e il recupero del razzo e di raccogliere dati durante il volo.
Il computer di volo è dotato di un sistema di telemetria che trasmette in tempo reale i dati raccolti a una *ground station*, permettendo il monitoraggio del volo da terra.



**Figura 1.2:** *Electronics bay.*

## 1.2 Contesto storico della creazione e dell'impiego di razzi sperimentali

I razzi sperimentali svolgono un ruolo cruciale nella ricerca e nello sviluppo tecnologico sin dalla fine del XIX secolo. Sebbene siano stati impiegati per scopi militari sin dal XIII secolo, fu tra il 1926 e il 1929 che il fisico americano Robert H. Goddard, ispirato dagli studi di Hermann Oberth e Konstantin Tsiolkovsky sulla velocità di fuga terrestre e sulla propulsione a razzo, realizzò il primo razzo a propellente liquido per scopi scientifici [SB06].

Da allora, i razzi sperimentali sono stati utilizzati per una vasta gamma di applicazioni, tra cui:

- La ricerca spaziale e l'esplorazione planetaria, come i razzi-sonda impiegati per esperimenti scientifici in microgravità.
- Lo studio dell'atmosfera e dei fenomeni meteorologici, come i razzi impiegati per analizzare la composizione degli strati superiori dell'atmosfera terrestre.
- I test di nuove tecnologie aerospaziali, inclusi materiali innovativi, sistemi di propulsione sperimentali, sistemi di navigazione autonoma e nuove tecnologie di telecomunicazione.
- Applicazioni educative e accademiche, che permettono a università e centri di ricerca di sviluppare esperimenti e formare studenti nel campo dell'ingegneria aerospaziale.

## 1.3 Contesto e importanza dei sistemi di telemetria nei razzi sperimentali

Le motivazioni per cui è importante sviluppare un sistema di telemetria per un razzo sperimentale sono molteplici.

In primo luogo, il monitoraggio del volo in tempo reale permette di avere un *feedback* immediato sulle prestazioni del razzo, sulle condizioni ambientali e sulle eventuali anomalie che possono verificarsi durante il volo. Inoltre, la raccolta di dati durante il volo è fondamentale per l'analisi post missione e per l'ottimizzazione del design del razzo. Infine, la telemetria è essenziale per poter analizzare il volo in caso di problemi o incidenti, al fine d'identificarne le cause se il sistema di *logging* a bordo dovesse andare perso.

Ciò pone delle sfide tecniche significative tra cui:

- L'affidabilità della trasmissione dati, che deve essere mantenuta nonostante le condizioni critiche a cui il sistema è soggetto, tra cui accelerazioni estreme, eventi atmosferici e interferenze elettromagnetiche.
- L'ottimizzazione della raccolta dati, che deve avvenire in tempo reale e con una frequenza sufficiente per garantire la precisione delle misurazioni.
- La gestione del consumo energetico, che deve essere minimizzato per garantire l'autonomia del sistema durante il volo.

## 1.4 Obiettivi della tesi e contributo personale

L'obiettivo di questa tesi è la descrizione delle fasi di progettazione, sviluppo e ottimizzazione del sistema di telemetria per il razzo Borealis di Aurora Rocketry, offrendo una panoramica dettagliata delle tecnologie utilizzate e delle scelte progettuali effettuate, valutando le prestazioni del sistema e identificando possibili miglioramenti futuri. Si vuole inoltre fornire un confronto tra le soluzioni adottate da Aurora Rocketry e quelle utilizzate in progetti simili, al fine di valutare l'impatto del sistema sviluppato nell'ambito della ricerca e dello sviluppo di razzi sperimentali.

Il mio contributo personale al progetto ha riguardato tutte le fasi di sviluppo del computer di volo, dal design iniziale alla realizzazione del prototipo, fino alla fase di test e ottimizzazione. Ho contribuito alla scelta delle tecnologie utilizzate, alla progettazione dell'architettura del computer di volo e del sistema di telemetria e alla scrittura del software per la raccolta, la strutturazione e la trasmissione dati. Inoltre, ho partecipato attivamente alle prove sperimentali e all'analisi dei dati raccolti, apportando modifiche e ottimizzazioni al sistema in base ai risultati ottenuti.

# Capitolo 2

# Descrizione del Computer di Volo

Il computer di volo di Borealis è basato su un microcontrollore ESP32-S3, montato su una scheda di sviluppo Arduino Nano ESP32, a cui sono collegati un modulo di raccolta dati, contenente i sensori, e un modulo di attuatori responsabile dell'attivazione di cariche pirotecniche tramite un circuito di innesco controllato dal microcontrollore.

I sensori utilizzati sono:

- Una Inertial Measurement Unit (IMU) *Bosch BNO055*[3] (Giroscopio, Accelerometro e Magnetometro).
- Due barometri *Adafruit MPRLS* con presa statica[4].

## 2.1 Funzioni principali e componenti chiave

Il computer di volo si occupa del monitoraggio del volo. Riceve i dati dal modulo dei sensori, tramite algoritmi di *sensor fusion* li elabora e contemporaneamente li trasmette alla *ground station*. In particolare, fa uso di un Extended Kalman Filter per filtrare e combinare i dati ottenuti dalla IMU e dei due barometri, ottenendo una stima affidabile dell'orientamento, della velocità verticale e dell'altitudine del razzo. Il filtro mitiga gli errori dei sensori, corregge la deriva della IMU e riduce le incertezze delle misurazioni barometriche, combinando i dati di accelerazione, orientamento e pressione atmosferica.

L'algoritmo di rilevamento dell'apogeo monitora l'altitudine e la velocità verticale del razzo. Quando la velocità verticale si annulla e il razzo inizia la discesa, il computer di volo rileva l'apogeo e attiva la sequenza di recupero, espellendo prima il paracadute *drogue* per stabilizzare il razzo, e, una volta raggiunta la quota

di sicurezza minima di 450 m, il paracadute principale per garantire una discesa controllata.

## 2.2 Integrazione e ruolo del sistema di telemetria

Per la telemetria è stato impiegato un modulo LoRa© *EByte E220-900T22D* a 868MHz collegato al microcontrollore tramite un'interfaccia Universal Asynchronous Receiver-Transmitter (UART) per trasmettere i dati di volo alla *ground station*.

Il sistema di telemetria è stato integrato nel computer di volo per permettere il monitoraggio del volo in tempo reale e la raccolta dati durante la missione, fornendo un'interfaccia di comunicazione tra il razzo e la *ground station*. Il sistema di telemetria è stato progettato per garantire l'affidabilità della trasmissione dati, la precisione delle misurazioni e l'ottimizzazione del consumo energetico.

Il software di gestione della telemetria è stato sviluppato in C++17 utilizzando una libreria open-source[(1)] per l'interfaccia con il modulo LoRa© e un protocollo di comunicazione personalizzato per la trasmissione e la ricezione dei dati.

# Capitolo 3

# Tecnologia LoRa© e la sua Applicazione

## 3.1 Panoramica della tecnologia LoRa©

LoRa© è una tecnologia proprietaria per la trasmissione wireless di dati a lungo raggio e basso consumo energetico. Impiega la modulazione Chirp Spread Spectrum (CSS), che garantisce un'elevata robustezza alle interferenze e consente comunicazioni su lunghe distanze a basso consumo energetico, a costo di una ridotta velocità di trasmissione, che rimane comunque adeguata per applicazioni di telemetria e monitoraggio in tempo reale.
LoRa© opera all'interno delle bande di frequenza Industrial, Scientific and Medical (ISM) non licenziate, con specifiche allocazioni in base alla regione geografica:

- **868 MHz** in Europa.
- **915 MHz** in Nord America.
- **433 MHz** in alcune aree dell'Asia.

Questa tecnologia può essere utilizzata sia in modalità *point-to-point*, per la comunicazione diretta tra dispositivi, sia in un'architettura di rete più estesa. In quest'ultimo caso, è comunemente adottata all'interno di un'infrastruttura Long Range Wide Area Network (LoRaWAN©), che consente la trasmissione di dati a lunga distanza tramite una rete di gateway interconnessi a server centralizzati.

### 3.1.1 Modulazione *Chirp Spread Spectrum*

La modulazione CSS è una tecnica di modulazione che fa uso di segnali a *chirp* per rappresentare informazioni digitali. I *chirp* sono segnali la cui frequenza varia linearmente nel tempo: quando la frequenza cresce si parla di *up-chirp*, mentre quando decresce si parla di *down-chirp*.

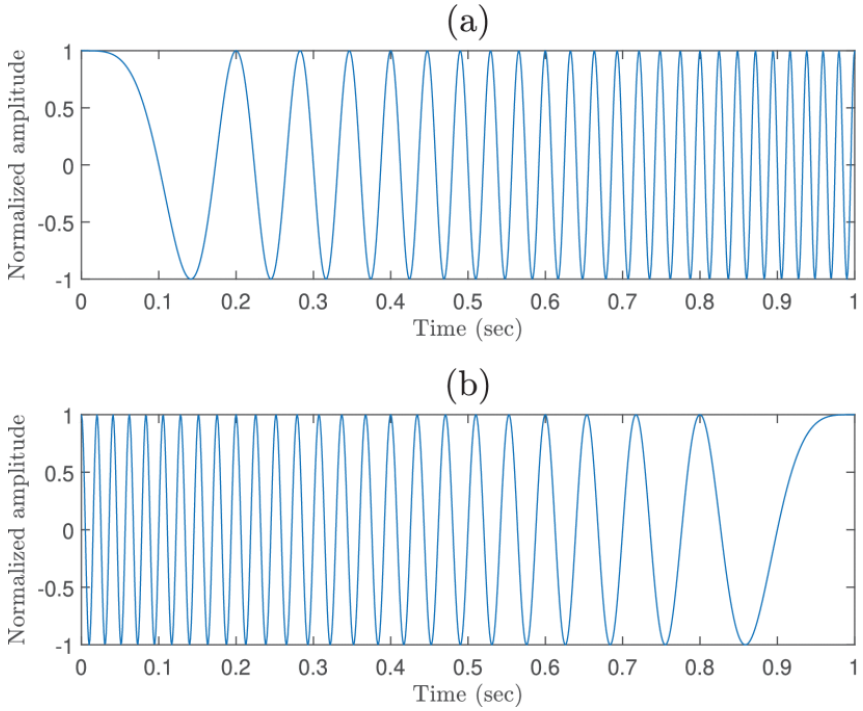



**Figura 3.1:** Esempio temporale di un *up-chirp* (a) e di un *down-chirp* (b). *Immagine adattata da* [MNBB24].

In un sistema CSS, ogni simbolo viene rappresentato da un singolo *chirp* e la quantità di bit che esso trasporta è determinata dallo Spreading Factor (SF), secondo la relazione $M = 2^{SF}$, dove $M$ rappresenta il numero di simboli ortogonali (cioè distinti tra loro).

La modulazione CSS si basa sulla generazione di un insieme di *chirp* ortogonali ottenuti mediante traslazioni circolari nel tempo di un *chirp* di riferimento (detto *chirp base*). La posizione temporale del *chirp* traslato è proporzionale al valore numerico del simbolo da trasmettere, permettendo la generazione di $M$ forme d'onda tra loro ortogonali.

Questa proprietà consente al ricevitore di recuperare il simbolo trasmesso mediante correlazione con i *chirp* possibili, tipicamente implementata in maniera efficiente

tramite la trasformata di Fourier (FFT), rendendo il sistema robusto a rumore, interferenze e attenuazione multipath [MNBB24].

## 3.2 Specifiche tecniche dei moduli utilizzati

Il sistema di telemetria di Borealis impiega due moduli LoRa© *EByte E220-900T22D*, uno in trasmissione a bordo del razzo e uno in ricezione nella *Ground Station*. Il modulo impiega la ricetrasmittente LoRa© *Semtech LLCC68* e opera nella banda di frequenza compresa tra 850.15 MHz e 930.15 MHz.

Le specifiche tecniche principali del modulo utilizzato sono riportate nella Tabella 3.1. Queste caratteristiche includono la potenza di trasmissione massima di +22 dBm (160 mW), la sensibilità del ricevitore pari a -129 dBm, e la configurabilità dello Spreading Factor (SF) nell'intervallo SF5 - SF12, che influisce direttamente sulla velocità di trasmissione (da 0.3 kbps a 62.5 kbps). Inoltre, il consumo energetico del modulo varia tra 120 mA in trasmissione alla potenza massima e 10 mA in ricezione[Datasheet 1], permettendo una gestione efficiente delle risorse in missioni di lunga durata.

| **Caratteristica** | **Valore** |
|---|---|
| **Banda di frequenza** | 850.15 MHz - 930.15 MHz |
| **Potenza di trasmissione** | +22 dBm (160 mW) |
| **Sensibilità del ricevitore** | -129 dBm |
| **Spreading Factor (SF)** | SF5 - SF12 |
| **Velocità di trasmissione** | 0.3 kbps - 62.5 kbps |
| **Consumo in trasmissione (a +22 dBm)** | 120 mA |
| **Consumo in ricezione** | 10 mA |
| **Interfaccia di comunicazione** | UART (3.3V TTL) |
| **Baud rate** | 1200 bps - 115200 bps |
| **Connettore antenna** | SMA femmina |
| **Impedanza connettore** | 50 Ω |

**Tabella 3.1:** Caratteristiche tecniche del modulo LoRa© EByte E220-900T22D.

## 3.3 Specifiche delle antenne utilizzate

La scelta delle antenne per il sistema telemetrico del razzo è stata guidata da esigenze tecniche specifiche, nonostante le opzioni disponibili fossero limitate. Le caratteristiche fondamentali del sistema richiedevano un'antenna a bordo in grado di garantire una copertura di radiazione uniforme su una superficie quanto più ampia possibile. Per la *ground station*, invece, è stata selezionata un'antenna direzionale.

### Antenna omnidirezionale (Razzo)

Per il razzo, la soluzione ottimale si è rivelata essere un'antenna omnidirezionale elicoidale. Questa scelta è praticamente obbligata a causa della variazione costante dell'orientamento del razzo durante il volo, rendendo impraticabile l'utilizzo di antenne direzionali, a meno di non implementare sistemi complessi di controllo dell'assetto, che tuttavia non rientrano nell'ambito del progetto attuale.

### Antenna direzionale (Ground Station)

Per la *ground station*, è stata selezionata un'antenna direzionale. Le due tecnologie principali prese in considerazione sono state:

- Antenna Yagi
- Antenna Log-Periodica

La differenza principale tra questi due tipi di antenne riguarda la direzionalità. L'antenna Log-Periodica offre una direzionalità maggiore rispetto alla Yagi, caratteristica che, in linea di principio, sarebbe desiderabile per il sistema di terra, in quanto consente di concentrare l'energia di ricezione in un angolo ristretto, aumentando il guadagno e riducendo il rumore di fondo. Tuttavia, l'eccessiva direzionalità della Log-Periodica avrebbe introdotto delle complicazioni operative. Poiché il tracciamento del razzo durante il volo sarà eseguito manualmente, una direzionalità troppo elevata avrebbe richiesto un'alta precisione di puntamento, difficile da mantenere senza strumenti automatizzati, aumentando il rischio di perdita temporanea del segnale.

L'antenna Yagi, con la sua direzionalità moderata, si è rivelata una scelta più equilibrata. Essa offre un buon compromesso tra guadagno e facilità di puntamento, consentendo un tracciamento sufficientemente preciso del razzo senza la necessità di strumenti avanzati d'inseguimento automatico.

## Dettagli tecnici delle antenne

| | **Antenna Omnidirezionale** | **Antenna Yagi** |
|---|---|---|
| Marca e modello | *Siretta Delta 5A* | *Siretta Oscar 3A* |
| Tipo | Omnidirezionale elicoidale | Direzionale Yagi |
| Frequenza di lavoro | 433 MHz - 5.9 GHz | 850 MHz - 930 MHz |
| Guadagno | 3 dBi | 10.55 dBi |
| Angolo di copertura | 360° | 60° |
| Impedenza | 50 Ω | 50 Ω |
| Polarizzazione | Verticale | Verticale |
| Tipo di connettore | SMA maschio | FME maschio |

**Tabella 3.2:** Specifiche tecniche dell'antenna omnidirezionale e dell'antenna Yagi.

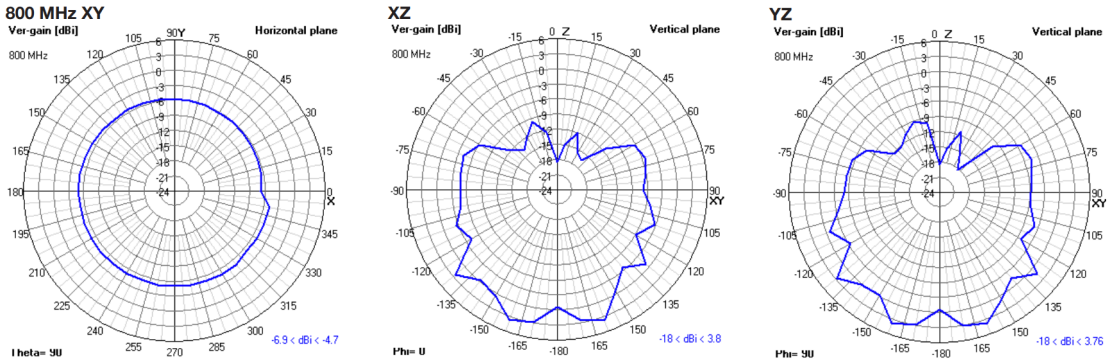



**Figura 3.2:** Grafici di radiazione dell'antenna omnidirezionale. Immagine adattata da (Datasheet 2)

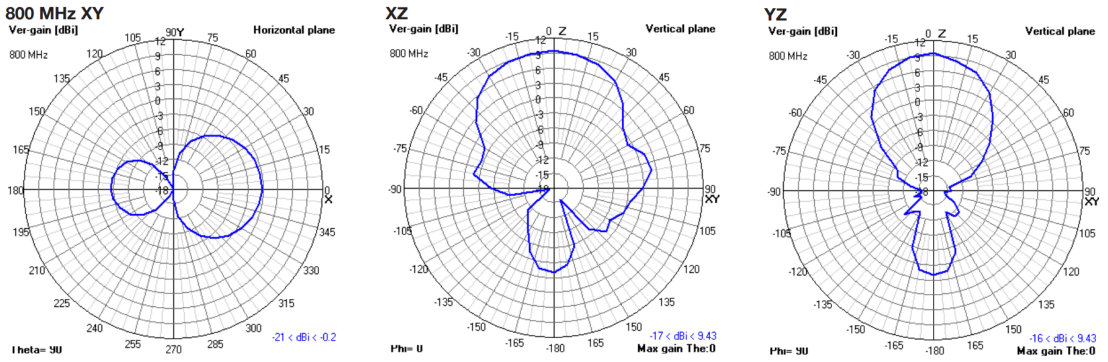



**Figura 3.3:** Grafici di radiazione dell'antenna Yagi. Immagine adattata da (Datasheet 3)

## 3.4 Motivazioni della scelta di LoRa© per il progetto

Il progetto richiedeva di individuare una tecnologia di trasmissione dati che garantisse un'elevata affidabilità e copertura, mantenendo un consumo energetico contenuto, anche in un ambiente operativo complesso come quello di un razzo sperimentale.

Le caratteristiche tecniche discusse nelle sezioni precedenti rendono LoRa© una scelta ideale per il progetto. I test effettuati in laboratorio hanno confermato la capacità dei moduli ricetrasmittenti impiegati di soddisfare i requisiti di trasmissione dati del sistema di telemetria.

In letteratura sono stati documentati numerosi casi di applicazione di LoRa© in sistemi simili [MTS22; MLS+24], che hanno dimostrato la sua efficacia in ambienti operativi critici, garantendo trasmissioni dati stabili e affidabili.

Infine, l'ampia diffusione dell'uso di LoRa© nell'Internet of Things (IoT) e la conseguente vasta disponibilità di hardware a costi contenuti hanno reso questa tecnologia una scelta conveniente e accessibile per il progetto.

# Capitolo 4

# Implementazione del Sistema di Telemetria

## 4.1 Architettura del sistema di trasmissione dati

Il sistema di trasmissione dati impiegato su Borealis è basato su un'architettura punto-punto monodirezionale.

Come discusso nel Capitolo 3 il sistema è composto dai seguenti componenti principali:

- **Modulo LoRa© trasmettitore (Razzo)**: EByte E220-900T22D per la trasmissione della telemetria alla *ground station* con antenna omnidirezionale (vedi Sezione 3.3). Collegato tramite interfaccia UART al computer di volo.
- **Modulo LoRa© ricevitore (Ground Station)**: EByte E220-900T22D, identico al ricevente, impiegato nella *ground station* per ricevere i dati con antenna direzionale Yagi (vedi Sezione 3.3). Collegato a un microcontrollore ESP32-S3 tramite interfaccia UART che si occupa di inviare i dati ricevuti a un PC tramite connessione seriale.

Il flusso dei dati nel sistema è il seguente:

1. I dati di volo vengono raccolti dai sensori a bordo del razzo e inviati al computer di volo.
2. Il computer di volo elabora i dati attraverso un *Extended Kalman Filter* (EKF) che permette di derivare l'altitudine e la velocità verticale del razzo.
3. I dati elaborati vengono inseriti in un oggetto JSON.

4. L'oggetto JSON viene serializzato e segmentato per rispettare i limiti di dimensione del payload dei moduli LoRa©.
5. Ogni segmento viene inserito in un pacchetto ad-hoc per gestire la logica di trasmissione e ricezione.
6. I dati serializzati vengono trasmessi al modulo LoRa© trasmettitore via UART
7. Il modulo LoRa© trasmettitore invia i pacchetti dati via wireless.
8. Il modulo LoRa© ricevitore riceve i dati trasmessi.
9. I dati ricevuti vengono deserializzati, ricomponendo il *payload* originale.
10. Il *payload* viene inviato al PC per essere salvati e in seguito analizzati.

Vista la presenza di forti vibrazioni, accelerazioni estreme, variazioni continue di assetto e orientamento, condizioni atmosferiche variabili e possibili interferenze elettromagnetiche, sono stati valutati gli impatti che questi fattori possono avere sulla trasmissione dei dati e sulla loro integrità.
Secondo i calcoli la banda di frequenza di 868 MHz, utilizzata dai moduli LoRa© impiegati, non è soggetta a interferenze significative da parte di altri dispositivi e la modulazione CSS impiegata garantisce una robustezza adeguata contro le interferenze, il rumore ambientale e l'effetto Doppler, permettendo una trasmissione affidabile anche in condizioni difficili. Bisogna inoltre tenere conto che nella *ground station* non é previsto alcun sistema di inseguimento, quindi l'antenna direzionale Yagi deve essere puntata manualmente verso il razzo durante il volo, per garantire una ricezione ottimale del segnale.

## 4.2 Protocollo di comunicazione e gestione dati

Il protocollo di comunicazione è stato progettato per garantire efficienza e semplicità, consentendo la trasmissione di pacchetti dati strutturati e adattabili a diversi formati.

Il pacchetto generato dal nodo trasmittente è composto dalle seguenti sezioni:

- ***Header* fisico** (3 byte), che include:
  - Byte alto dell'indirizzo del destinatario
  - Byte basso dell'indirizzo del destinatario
  - Identificativo del canale di trasmissione
- ***Payload*** (194 byte): contiene i dati applicativi da trasmettere.
- ***Footer*** (3 byte): include il valore del RSSI, utilizzato per la diagnostica della qualità del segnale e il Controllo di Ridondanza Ciclica (CRC) per garantire l'integrità dei dati.

Considerando le limitazioni fisiche del modulo LoRa© impiegato (in particolare, la dimensione massima dei pacchetti), il protocollo è stato esteso per supportare la segmentazione del *payload*. Quando il dato da trasmettere eccede la dimensione massima consentita, esso viene suddiviso in sotto-segmenti detti *chunk*, ciascuno trasmesso come pacchetto indipendente.

Per supportare questa funzionalità è stato introdotto un **header logico** aggiuntivo di 11 byte in testa al *payload*, che include le seguenti informazioni di controllo:

- ***Packet Number***: identificatore univoco della sequenza di pacchetti riferita a una singola lettura.
- ***Total Chunks***: numero totale di chunk in cui è stato suddiviso il *payload* originale.
- ***Chunk Index***: indice progressivo del chunk corrente all'interno della sequenza, utilizzato per la ricostruzione corretta del dato.
- ***Payload Size***: dimensione effettiva del *payload* trasportato nel chunk, espressa in byte.
- ***Timestamp***: riferimento temporale associato all'invio del pacchetto, utile per analisi temporali e sincronizzazione.
- ***Protocol Version***: campo riservato per la gestione di versioni successive del protocollo, utile in ottica di retrocompatibilità.

Considerando la dimensione del *header* logico e del *footer*, il *payload* massimo trasportabile in un singolo pacchetto è di 183 byte usando i moduli LoRa© EByte E220-900T22D, che hanno un limite di 200 byte per pacchetto.

Uno schema del pacchetto è riportato nella Figura 4.1.

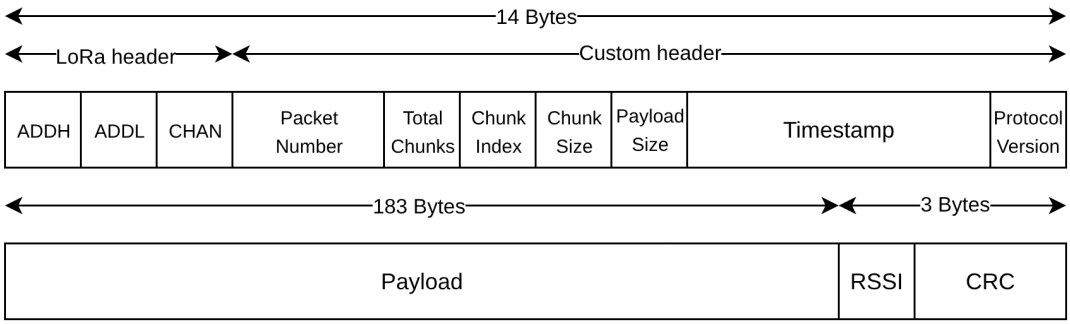



**Figura 4.1:** Schema del pacchetto di dati trasmesso dal nodo trasmittente.

## 4.3 Implementazione software e algoritmi di gestione della trasmissione

Il software è stato implementato in C++17. Nei seguenti paragrafi verrà illustrata l'architettura dei principali moduli software: il computer di volo, il modulo trasmettitore e il modulo ricevitore.

### 4.3.1 Architettura software

#### Computer di volo

Il computer di volo è responsabile della raccolta e dell'elaborazione dei dati provenienti dai sensori, della gestione della telemetria e dell'attivazione degli attuatori per il recupero del razzo.

Il sistema è orchestrato da una macchina a stati finiti, che gestisce le diverse fasi del volo, rappresentata nella Figura 4.2.

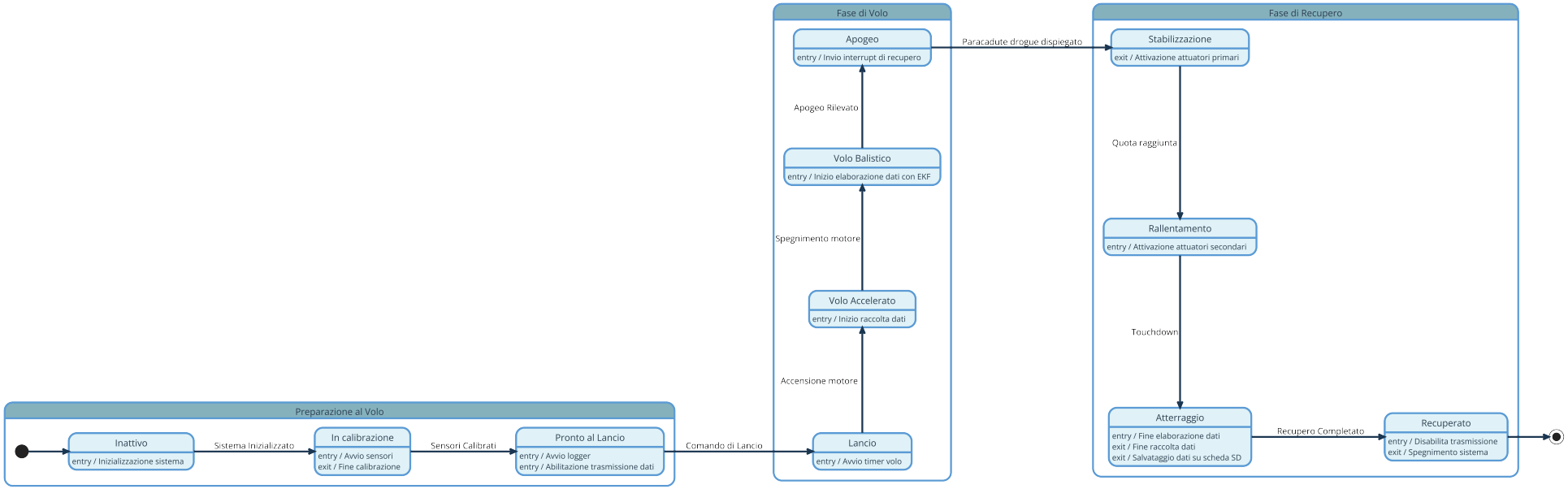



**Figura 4.2:** Macchina a stati finiti del computer di volo.

Il computer di volo è composto da 4 elementi principali:

**Modulo di raccolta dati:** Gestisce la lettura dei sensori.

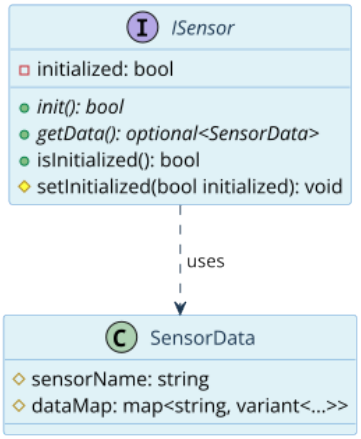



**Figura 4.3:** Interfaccia per la raccolta dati da un sensore.

**Modulo di logging:** Si occupa della serializzazione dei dati in formato JSON.

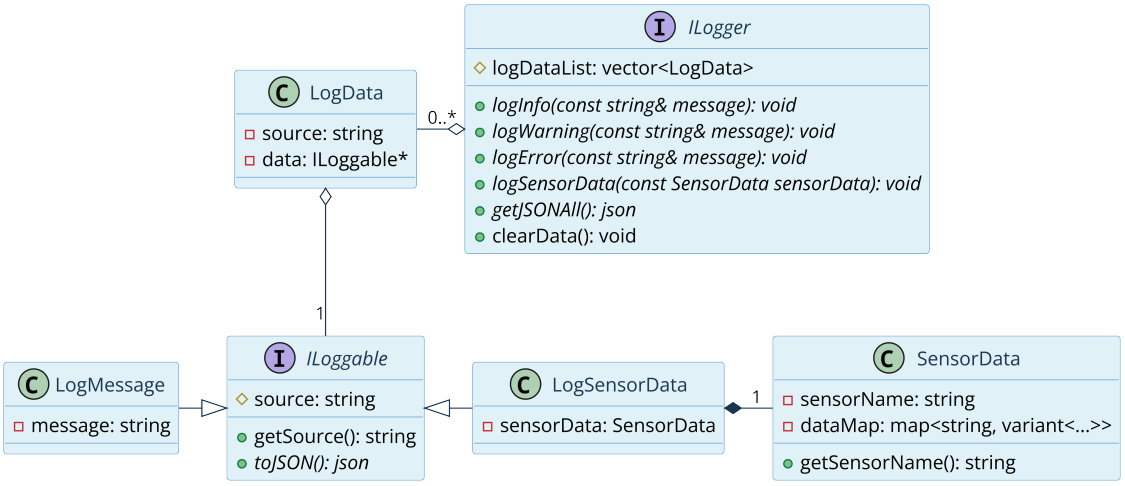



**Figura 4.4:** Interfaccia per il logging dei dati.

**Modulo di elaborazione dati:** Si occupa dell'elaborazione dei dati raccolti dai sensori, applicando un EKF e algoritmi di *sensor fusion* per la stime dell'altitudine e della velocità verticale, che sono utilizzate per il rilevamento dell'apogeo e la gestione della sequenza di recupero.

**Modulo di telemetria:** Si occupa della serializzazione, segmentazione e trasmissione dei dati via LoRa©.

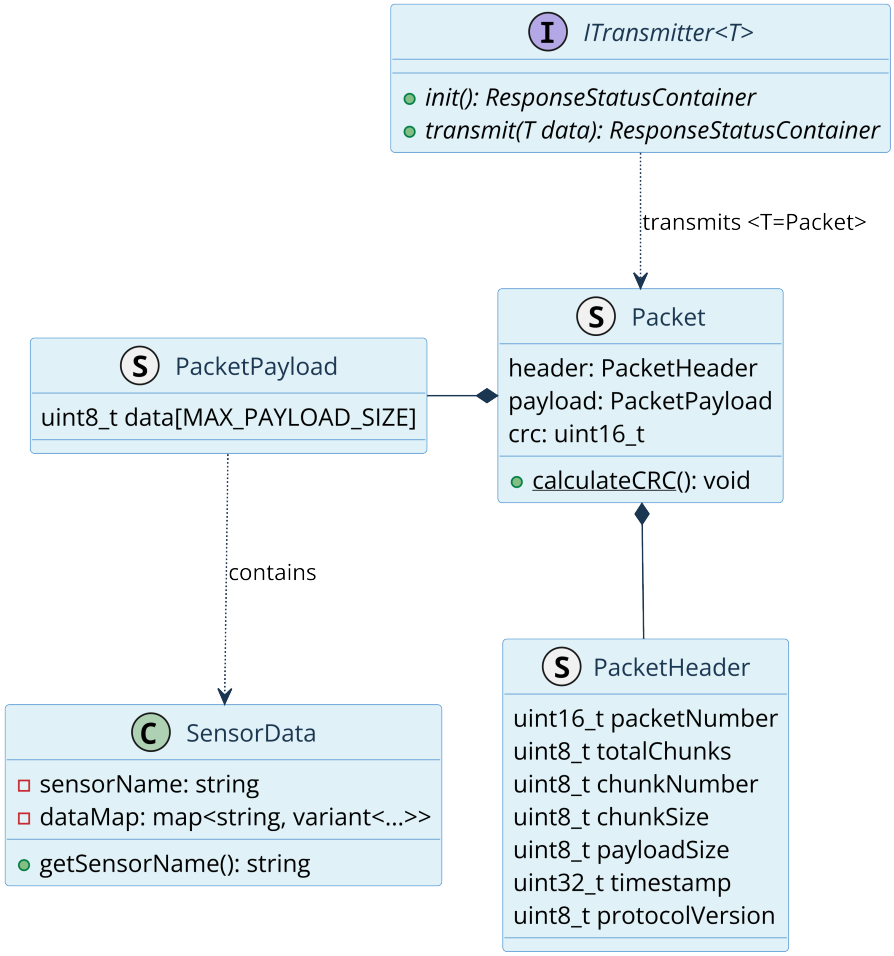



**Figura 4.5:** Interfaccia per la telemetria.

### Ground Station

La *ground station* è responsabile della ricezione dei dati trasmessi dal razzo, della loro deserializzazione e dell'invio al PC su cui vengono salvati.

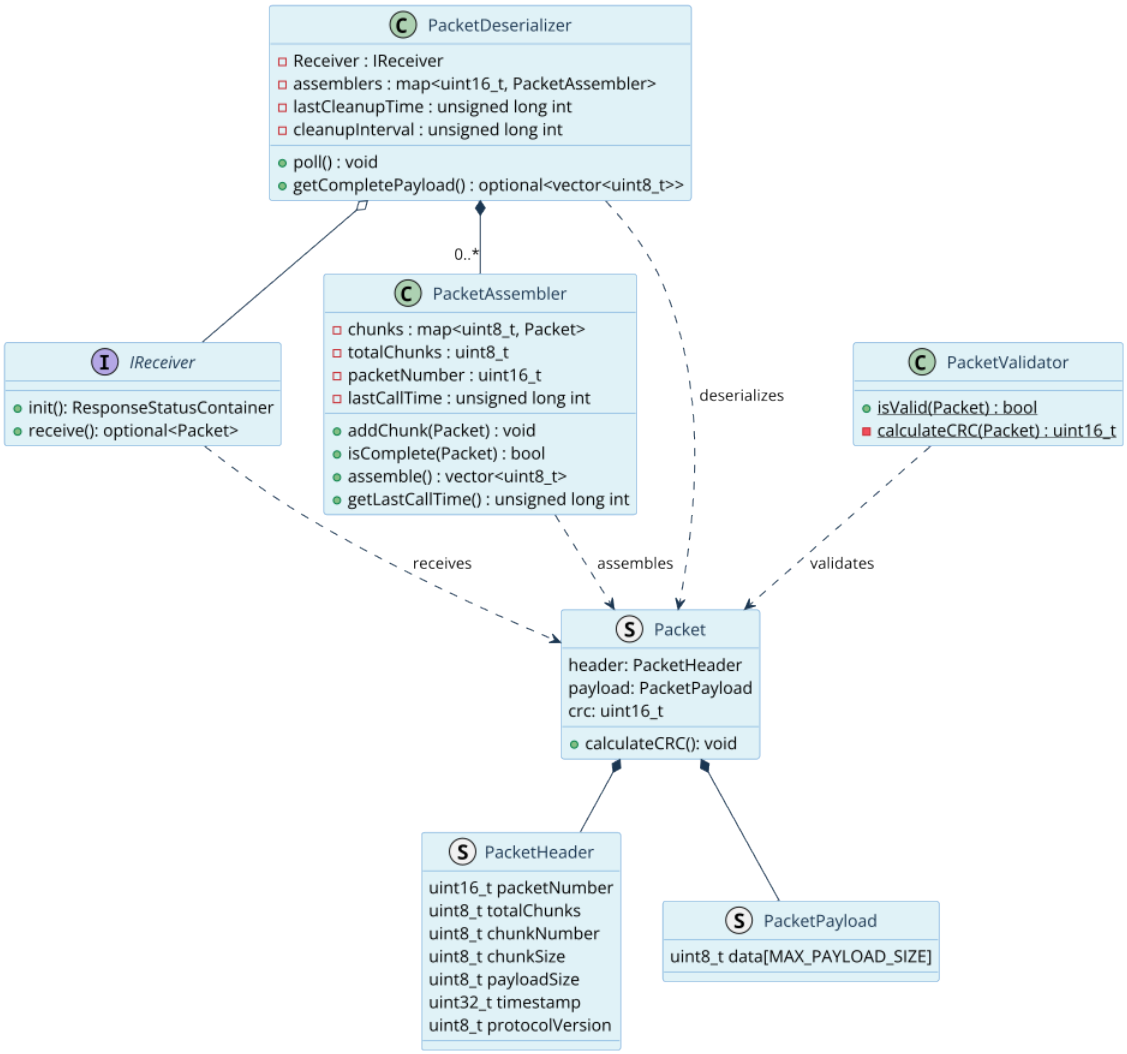



**Figura 4.6:** Architettura della *ground station*.

### 4.3.2 Algoritmo di ricostruzione del *payload*

L'algoritmo di ricostruzione del *payload* originale a partire dai pacchetti ricevuti, permette di gestire la segmentazione dei dati e garantire l'integrità delle informazioni trasmesse.

L'algoritmo organizza i pacchetti ricevuti in base al loro identificativo di sequenza (*Packet Number*), consentendo la ricostruzione dei messaggi frammentati. Ogni pacchetto in arrivo viene validato, verificando la consistenza dell'*header* logico e il CRC per assicurare l'integritá del contenuto.

I pacchetti validati vengono inseriti in una struttura dati temporanea, corrispondente alla sequenza a cui appartengono, indicizzandoli in base al *Chunk Number*. Una volta ricevuti tutti i *chunk* previsti per una sequenza, il *payload* originale viene ricostruito, concatenando i dati dei *chunk*.

Per ottimizzare l'uso delle risorse, é stato implementato un meccanismo di *drop* temporizzato per le sequenze incomplete, prevenendo cosí l'accumulo di dati parziali, sfruttando la tolleranza del sistema alla perdita occasionale di letture dei dati dei sensori e la garanzia di disponibilità dei dati grazie al sistema di logging secondario a bordo del computer di volo.

# Capitolo 5

# Valutazione delle Prestazioni

## 5.1 Metodologia di test e criteri di valutazione

| **Test** | **Obiettivo** | **Metrica misurata** |
|---|---|---|
| Affidabilitá della trasmissione | Verificare la capacitá del sistema di trasmettere e ricevere pacchetti su distanze variabili, in condizioni ambientali diverse, con ostacoli, rumore e disallineamento tra le antenne. | RSSI, angolo di disallineamento tra le antenne |
| Throughput | Valutare la velocità effettiva di trasmissione su diversi scenari di distanza e ambiente | Pacchetti/s, bit/s, RSSI |
| Segmentazione e ricostruzione | Verificare l'efficienza e la scalabilità del protocollo nel ricostruire correttamente i payload segmentati | Tempo medio di ricostruzione |
| Meccanismo di *drop* | Verificare la capacità del sistema di identificare e scartare pacchetti incompleti o corrotti | Loss rate (%), pacchetti scartati, |
| Riordinamento dei *chunk* | Verificare la capacità del sistema di gestire chunk ricevuti in ordine scorretto e ripristinare l'ordine corretto prima della ricostruzione | Stato di ordinamento |
| Robustezza e affidabilità | Valutare la stabilità del sistema durante trasmissioni prolungate e in condizioni realistiche di rumore e interferenza | Stabilità throughput, variazione delay |

**Tabella 5.1:** Schema riassuntivo dei test sperimentali

### 5.1.1 Test preliminari

La valutazione delle prestazioni del sistema è stata preceduta da una serie di test preliminari, condotti per determinare le capacità effettive dei moduli prima dell'implementazione del protocollo di comunicazione. Utilizzando una configurazione punto-punto, sono state effettuate misurazioni del RSSI in diverse condizioni ambientali. I risultati di questi test sono stati favorevoli, dimostrando una comunicazione efficace anche in presenza di ostacoli come alberi, edifici e rilievi, fornendo una base solida per lo sviluppo successivo.

I moduli sono stati configurati con i parametri di default descritti in (Datasheet 1).

#### Disallineamento angolare dell'antenna direzionale

Per ogni distanza si sono effettuate 8 misurazioni, con angoli di disallineamento dell'antenna direzionale compresi tra 0° e 30°, e un ulteriore test a 90° per verificare la perdita di segnale in caso di disallineamento completo.

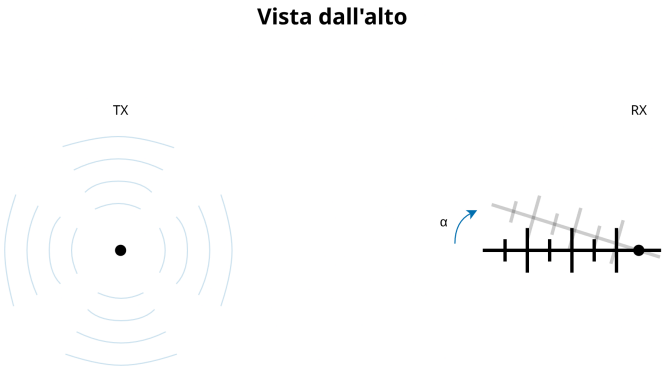



**Figura 5.1:** Test di disallineamento angolare dell'antenna direzionale.

### Disallineamento angolare dell'antenna omnidirezionale

In seguito é stato eseguito lo stesso test, stavolta variando l'angolo di disallineamento dell'antenna omnidirezionale sul piano verticale tra 0° e 30°, e un ulteriore test a 90°, per simulare l'effetto della variazione di assetto del razzo durante il volo.

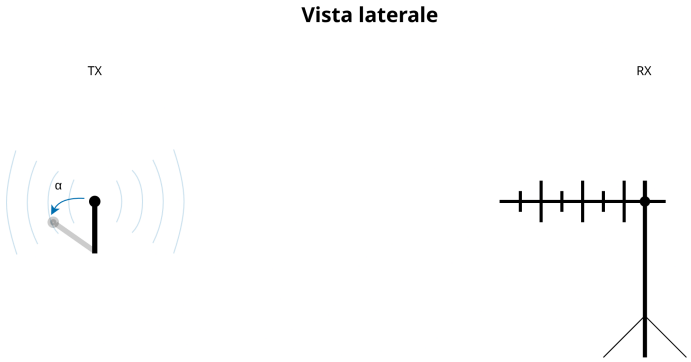



**Figura 5.2:** Test di disallineamento angolare dell'antenna omnidirezionale.

## 5.1.2 Test sul protocollo di comunicazione

Dopo aver verificato le prestazioni dei moduli LoRa© e delle antenne, si è proceduto a testare il protocollo di comunicazione implementato, valutando l'affidabilità della trasmissione dei dati, la capacità di gestire la segmentazione e la ricostruzione dei messaggi, e la robustezza del sistema in condizioni di rumore e interferenze.

### Test del *throughput*

| Distanza | Frequenza di trasmissione | Potenza trasmissiva | Variabili di controllo |
|---|---|---|---|
| 10m | 873.125 MHz | +17 dBm | Dimensione del payload |
| 50m | | | |
| 100m | | | |
| 150m | | | |
| 200m | | | |
| 250m | | | |
| 500m | | | |

**Tabella 5.2:** Condizioni del test del throughput.

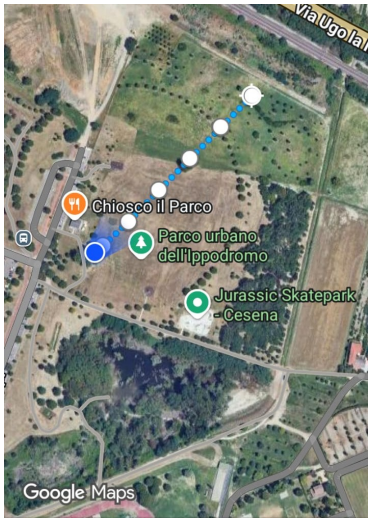



**Figura 5.3:** Posizioni di test da 10m a 250m con ostacoli.

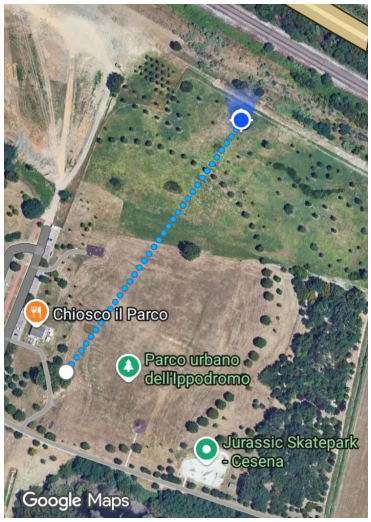



**Figura 5.4:** Posizione di test a 250m senza ostacoli.

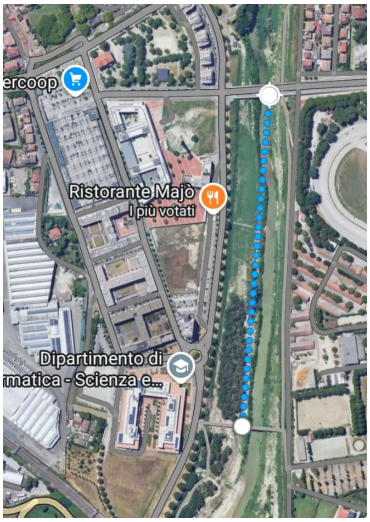



**Figura 5.5:** Posizione di test a 500m.

### Test del *throughput*

Il test ha avuto l'obiettivo di misurare il *throughput* effettivo del sistema di telemetria, valutando la quantità di dati ricevuti correttamente al variare della distanza tra le antenne, in condizioni ambientali reali.

Per ciascuna prova è stato trasmesso un *payload* di dimensione fissa, suddiviso in 5 *chunk*, secondo il criterio di frammentazione descritto nel Capitolo 4.

**Dettagli operativi del setup sperimentale:** Sulla base dei risultati dei test preliminari, discussi nel Capitolo 6, i moduli LoRa© sono stati riconfigurati, impostando un *Air Data Rate* (ADR) di 9.2 kbps e configurando il *baud rate* della linea UART a 115200 bps. Questa configurazione ha consentito di aumentare il *throughput* effettivo del sistema, a fronte di una riduzione della portata massima di trasmissione.

Il test è stato eseguito in configurazione punto-punto, con l'antenna ricevente mantenuta in posizione fissa, mentre l'antenna trasmittente è stata progressivamente allontanata lungo una direttrice lineare, posizionandola alle distanze indicate nella Tabella 5.2 e visibili nelle Figure 5.3-5.5.

Per la distanza di 250 m sono state effettuate due misurazioni distinte:

- **Scenario con ostacoli:** la trasmittente era posizionata dietro un leggero rilievo e fitta vegetazione, in assenza di linea di vista diretta (non-*Line of Sight* (LoS));

- **Scenario parzialmente libero:** la trasmittente è stata spostata lateralmente di alcuni metri, ottenendo una visibilità migliorata, con la presenza di pochi alberi isolati lungo la traiettoria.

Il test alla distanza di 200 m è stato effettuato sulla cima di un piccolo rilievo naturale.

Il test alla distanza di 500 m è stato invece eseguito in una seconda location, scelta per garantire una condizione di LoS più favorevole: la trasmissione è avvenuta tra due ponti stradali separati da un fiume, con l'antenna ricevente posizionata su un ponte e la trasmittente sull'altro. Nonostante la presenza di vegetazione lungo gli argini, la visibilità tra le antenne è rimasta sostanzialmente diretta.

Queste configurazioni hanno permesso di analizzare l'impatto reale degli ostacoli naturali sulla qualità del collegamento radio e sul throughput risultante, mantenendo il più possibile costanti le condizioni operative (altezza delle antenne, orientamento, parametri di trasmissione).

Per ogni distanza, sono state misurate le seguenti metriche:

- RSSI medio rilevato;
- Numero di pacchetti ricevuti;
- Pacchetti al secondo (*pkt/s*);
- Byte al secondo (*Byte/s*) ricevuti correttamente.

### Test sulla segmentazione e ricostruzione dei dati

| Distanza | Frequenza di trasmissione | Potenza trasmissiva | Variabili di controllo |
|---|---|---|---|
| 5m | 873.125 MHz | +17 dBm | Dimensione del payload |

**Tabella 5.3:** Condizioni del test di segmentazione e ricostruzione dei dati.

Per valutare le prestazioni del protocollo di trasmissione nella gestione di pacchetti segmentati, è stato condotto un test sperimentale mirato a misurare il tempo necessario alla ricostruzione del *payload* originale a partire dai *chunk* ricevuti.

Il test è stato eseguito considerando *payload* di dimensioni crescenti, suddivisi in 2, 3, 4 e 5 *chunk*. L'obiettivo era analizzare l'evoluzione del tempo medio di ricostruzione al variare del numero di *chunk*, al fine di valutare l'efficienza e la scalabilità del protocollo nella fase di riassemblaggio.

Questa analisi permette di valutare le prestazioni del sistema in scenari di trasmissione con pacchetti di lunghezza variabile e di evidenziare eventuali criticità legate alla latenza introdotta dalla segmentazione.

Il tempo di ricostruzione di un pacchetto è calcolato come:

$$\delta_r = (t_{UltimoChunk} - t_{PrimoChunk}) \tag{5.1}$$

Il test è stato effettuato sia con la configurazione di default che con la configurazione descritta nella Sezione 5.1.2.

### Test del meccanismo di *drop*

| Distanza | Frequenza di trasmissione | Potenza trasmissiva | Variabili di controllo |
|---|---|---|---|
| 5m | 873.125 MHz | +17 dBm | Frequenza di mancata trasmissione di un *chunk* |

**Tabella 5.4:** Condizioni del test del meccanismo di *drop*.

La valutazione del meccanismo di *drop* dei pacchetti è stata effettuata per verificare la capacità del sistema di gestire situazioni in cui i pacchetti ricevuti non sono completi o presentano errori di trasmissione. Per eseguire il test, è stata simulata la mancata trasmissione di alcuni *chunk*, consentendo di verificare che il sistema identificasse e scartasse correttamente i pacchetti incompleti.

### Test del riordinamento dei pacchetti

| Distanza | Frequenza di trasmissione | Potenza trasmissiva | Variabili di controllo |
|---|---|---|---|
| 5m | 873.125 MHz | +17 dBm | Frequenza di invio di *chunk* fuori ordine |

**Tabella 5.5:** Condizioni del test del meccanismo di riordinamento.

Per garantire l'integritá dei dati, il protocollo mantiene i *chunk* ricevuti in una struttura dati ordinata. Per verificare l'efficacia del meccanismo di riordinamento, sono stati inviati alcuni *chunk* in ordine invertito, consentendo di valutare la capacità del sistema di riconoscere le sequenze fuori ordine, e riordinare i pacchetti prima della ricostruzione del *payload* originale.

**Test di robustezza e affidabilità**

| Distanza | Frequenza di trasmissione | Potenza trasmissiva | Variabili di controllo |
|---|---|---|---|
| 5m | 873.125 MHz | +17 dBm | Durata di trasmissione |
| | | | Dimensione del *payload* |
| | | | Frequenza di trasmissione della seconda antenna |

**Tabella 5.6:** Condizioni del test di robustezza e affidabilità.

Per valutare la robustezza del sistema, sono stati condotti due test sperimentali. Il primo ha riguardato la trasmissione di pacchetti in condizioni ottimali, a distanza ravvicinata, con un *payload* di dimensioni fisse, per verificare la stabilità della comunicazione e studiare eventuali degradazioni delle prestazioni nel tempo. Il test ha previsto l'invio di pacchetti frammentati per un periodo di 15 minuti, monitorando la regolarità della trasmissione e il tempo di ricostruzione del *payload* originale.

Il secondo test ha previsto l'attivazione di un secondo nodo trasmittente, operante sulla stessa frequenza della prima antenna, per verificare eventuali effetti di interferenze co-canalari. Il test è stato condotto nelle stesse condizioni del primo, aggiungendo l'invio continuo di pacchetti da parte della seconda antenna. I risultati dei due test sono stati successivamente confrontati per valutare l'effettivo impatto delle interferenze sulla qualità del segnale e sulla stabilità della trasmissione.

## 5.2 Ottimizzazioni apportate durante lo sviluppo

### 5.2.1 Configurazione personalizzata dei moduli radio

Come descritto nella Sezione 5.1.2, in seguito ai test preliminari svolti, i parametri di configurazione dei moduli sono stati modificati per aumentare l'ADR e il *baud rate* dell'interfaccia UART del modulo. Questo ha permesso di migliorare il *throughput*, con i risultati descritti nella Sezione 3.

### 5.2.2 Compressione dei dati

Considerando le limitazioni intrinseche dei moduli LoRa© utilizzati, i tempi di ricostruzione osservati risultano comunque compatibili con i requisiti applicativi. Per ridurre ulteriormente il carico trasmissivo, è stata implementata una compressione delta per ridurre la dimensione complessiva dei *payload* mantenendo inalterato il contenuto informativo.

### 5.2.3 Gestione dei pacchetti

Come discusso nella Sezione 4.3.2, é stato implementato un meccanismo di *drop* dei pacchetti, che consente di scartare i pacchetti incompleti o corrotti, evitando di accumulare dati parziali che potrebbero compromettere le prestazioni e l'affidabilità del sistema. Questo meccanismo si basa su un timeout configurabile, che permette di scartare i pacchetti che non sono stati completati entro un certo intervallo di tempo, evitando l'accumulo di dati inutili e garantendo una gestione efficiente della memoria.

Inoltre, per garantire l'integrità dei dati, il protocollo implementa un meccanismo di riordinamento dei pacchetti, che consente di gestire situazioni in cui i *chunk* arrivano in ordine non sequenziale.

# Capitolo 6

# Risultati e Analisi

## 6.1 Analisi dei dati raccolti durante le prove sperimentali

### 6.1.1 Test preliminari

I test preliminari hanno dimostrato che i moduli LoRa© EByte E220-900T22D sono in grado di mantenere una comunicazione stabile anche in presenza di ostacoli naturali (rilievi collinari, vegetazione) e artificiali, fino a distanze dell'ordine dei 2000 metri. Questi risultati evidenziano la solidità della modulazione CSS, in grado di garantire una buona resilienza al rumore ambientale e alle interferenze radio locali.

Come mostrato in Figura 6.1, anche con un disallineamento angolare tra le antenne, l'intensità del segnale ricevuto è rimasta superiore al 60% in tutti i casi testati, con assenza di errori di decodifica o perdite significative di pacchetti.

I risultati sperimentali raccolti confermano la validità della scelta architetturale adottata per il sistema radio, consentendo di procedere con lo sviluppo del protocollo di comunicazione e della logica di trasmissione, assumendo una robusta copertura anche in scenari *outdoor* complessi.

## Disallineamento angolare dell'antenna direzionale

| Distanza (m) | 0° | 5° | 10° | 15° | 20° | 25° | 30° | 90° |
|---|---|---|---|---|---|---|---|---|
| **100** | 83.92 | 83.92 | 83.14 | 83.14 | 83.14 | 83.14 | 83.14 | 78.43 |
| **250** | 80.39 | 78.43 | 80.78 | 79.61 | 79.61 | 76.86 | 76.86 | 74.51 |
| **780** | 78.43 | 76.47 | 77.25 | 76.47 | 74.51 | 74.51 | 74.51 | 70.59 |
| **1000** | 69.02 | 68.63 | 66.67 | 65.49 | 68.63 | 71.37 | 71.37 | 69.80 |
| **1500** | 63.53 | 63.14 | 63.53 | 62.75 | 63.53 | 62.35 | 62.75 | 63.92 |
| **2000** | 62.75 | 62.75 | 62.75 | 63.14 | 62.75 | 63.14 | 62.75 | 62.75 |

**Tabella 6.1:** Intensità del segnale (%) in funzione del disallineamento angolare dell'antenna direzionale e della distanza

## Disallineamento angolare dell'antenna omnidirezionale

| Distanza (m) | 0° | 5° | 10° | 15° | 20° | 25° | 30° | 90° |
|---|---|---|---|---|---|---|---|---|
| **100** | 80.00 | 78.43 | 77.65 | 78.43 | 80.39 | 82.35 | 82.35 | 84.31 |
| **250** | 76.47 | 79.61 | 79.61 | 78.43 | 78.43 | 78.82 | 78.82 | 82.35 |
| **780** | 73.73 | 72.55 | 69.80 | 72.55 | 72.94 | 73.73 | 71.37 | 76.47 |
| **1000** | 66.27 | 67.84 | 66.67 | 68.24 | 67.06 | 68.63 | 68.24 | 68.63 |
| **1500** | 62.75 | 62.75 | 63.14 | 62.75 | 62.35 | 62.35 | 62.35 | 64.71 |
| **2000** | 63.14 | 63.14 | 63.14 | 63.14 | 63.14 | 63.14 | 63.14 | 63.14 |

**Tabella 6.2:** Intensità del segnale (%) in funzione del disallineamento angolare dell'antenna omnidirezionale e della distanza

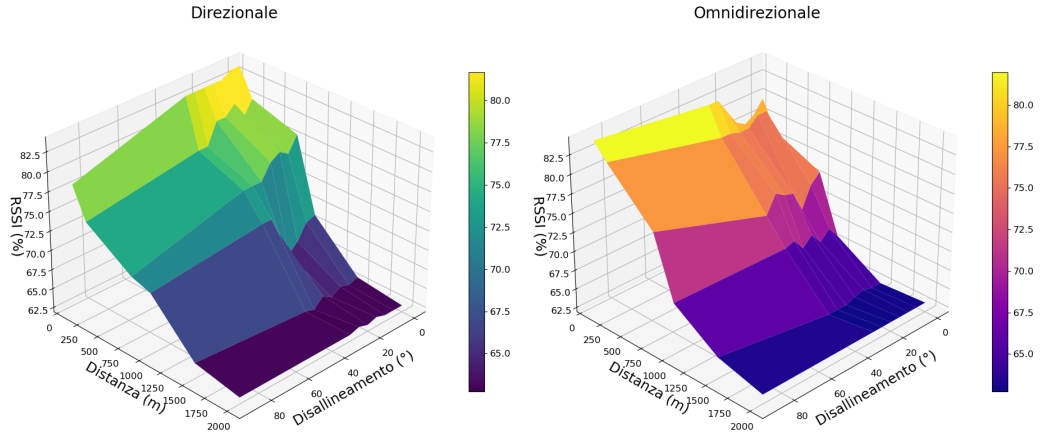



**Figura 6.1:** Intensità del segnale in funzione della distanza e dell'angolo di disallineamento delle antenne.

### 6.1.2 Risultati dei test sul protocollo di comunicazione

#### Test del *throughput*

I risultati del test di *throughput* indicano che il sistema di telemetria mantiene prestazioni adeguate anche in condizioni operative complesse, con buona robustezza alla variazione della distanza e alla presenza di ostacoli naturali.

Come riportato nella Tabella 6.3, il throughput si è mantenuto stabile (3.30 pkt/s e 646 B/s medi) per tutte le distanze testate, con una leggera riduzione nello scenario a 250 m in presenza di ostacoli (vegetazione e rilievi). In quel caso, la variabilità del throughput risulta più elevata (deviazione standard: 10.85 B/s), confermando l'impatto degli ostacoli sulla stabilità del collegamento. Al contrario, lo stesso scenario a 250 m con linea di vista libera ha mostrato prestazioni sovrapponibili a quelle rilevate a 50–100–500 m.

| **Distanza** | **RSSI [%]** | | | | **Throughput [Byte/s]** | | | | **Pkt/s** | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Media** | $\sigma$ | **Min** | **Max** | **Media** | $\sigma$ | **Min** | **Max** | **Media** | $\sigma$ |
| 10 m | 86.25 | 0.54 | 84.31 | 87.84 | 646.46 | 0.19 | 644.74 | 646.86 | 3.30 | 0.00 |
| 50 m | 73.43 | 0.78 | 69.41 | 75.29 | 646.44 | 0.16 | 644.74 | 646.86 | 3.30 | 0.00 |
| 100 m | 70.18 | 0.86 | 69.02 | 71.76 | 646.49 | 0.09 | 646.15 | 646.86 | 3.30 | 0.00 |
| 150 m | 65.79 | 1.73 | 60.78 | 69.02 | 646.45 | 0.15 | 644.74 | 646.86 | 3.30 | 0.00 |
| 200 m | 71.17 | 1.15 | 67.06 | 72.94 | 646.45 | 0.07 | 645.80 | 646.86 | 3.30 | 0.00 |
| 250 m (ostacoli) | 64.90 | 1.37 | 60.39 | 67.06 | 634.54 | 10.85 | 587.71 | 646.51 | 3.24 | 0.06 |
| 250 m (libero) | 65.21 | 1.33 | 59.61 | 67.84 | 646.49 | 0.09 | 646.44 | 646.86 | 3.30 | 0.00 |
| 500 m | 73.64 | 0.67 | 71.76 | 75.29 | 646.42 | 0.12 | 644.74 | 646.51 | 3.30 | 0.00 |

**Tabella 6.3:** Statistiche sperimentali del throughput in funzione della distanza

La qualità del segnale, rappresentata dal valore di RSSI, è risultata elevata in tutti i test, con valori medi sempre superiori al 65%, ad eccezione della configurazione 250 m con ostacoli. La Figura 6.2 mostra l'andamento del RSSI medio in funzione della distanza, con una regressione logaritmica che evidenzia un lieve decadimento complessivo all'aumentare della distanza. Il boxplot in Figura 6.3 evidenzia inoltre un'anomala crescita del valore medio del RSSI a 200 m, attribuibile alla posizione sopraelevata della trasmittente in quel punto, causata dalla conformazione naturale del terreno.

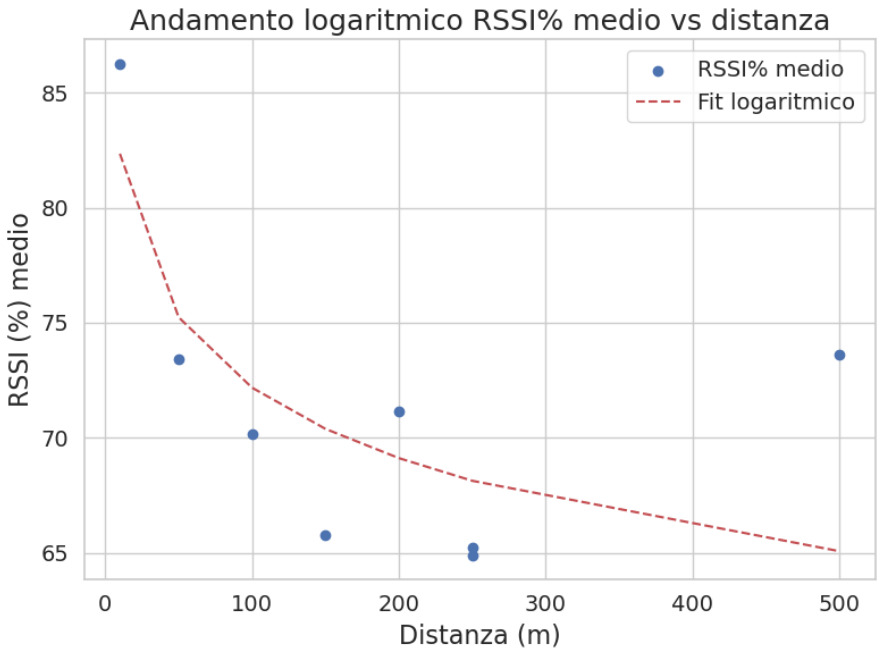



**Figura 6.2:** Andamento del RSSI in funzione della distanza (regressione logaritmica).

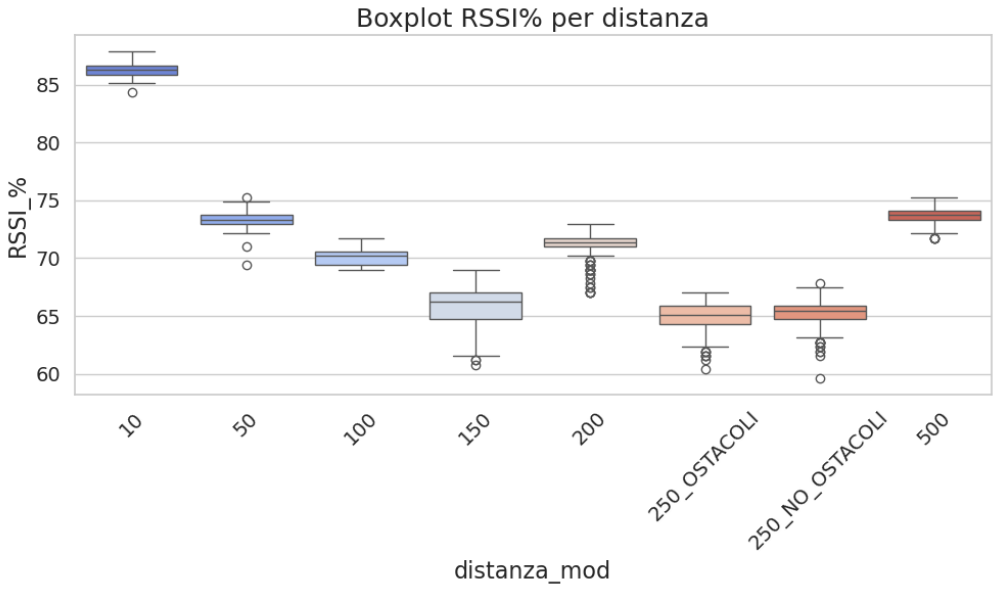



**Figura 6.3:** Distribuzione del RSSI per ciascuna distanza testata.

L'impatto degli ostacoli naturali è ancora più evidente osservando il comportamento del throughput. La Figura 6.4 mostra la dispersione dei valori di throughput in funzione della distanza: nello scenario con ostacoli a 250 m si osserva una maggiore variabilità e una riduzione rispetto alla configurazione equivalente con linea di vista. La Figura 6.5 mette a confronto l'RSSI e il throughput medi per ciascuna distanza, evidenziando la coerenza tra decadimento del segnale e leggera riduzione delle prestazioni.

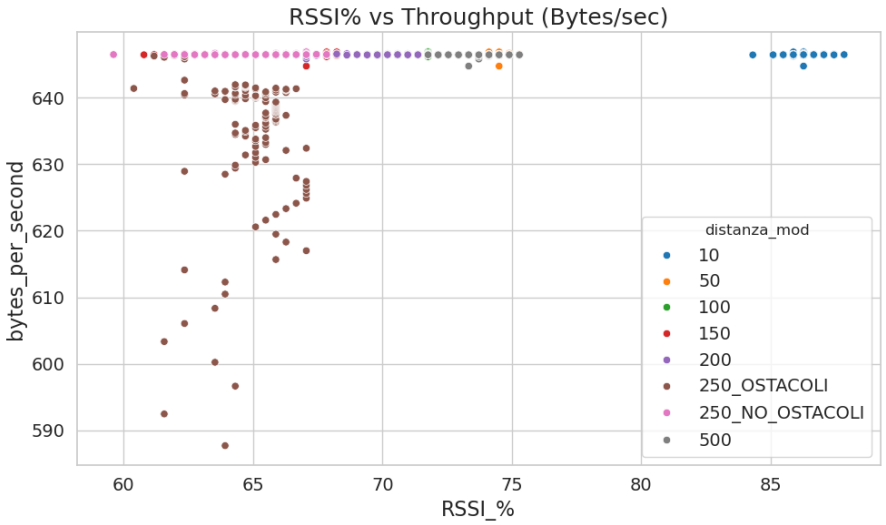



**Figura 6.4:** Throughput in funzione della distanza (distribuzione puntuale).

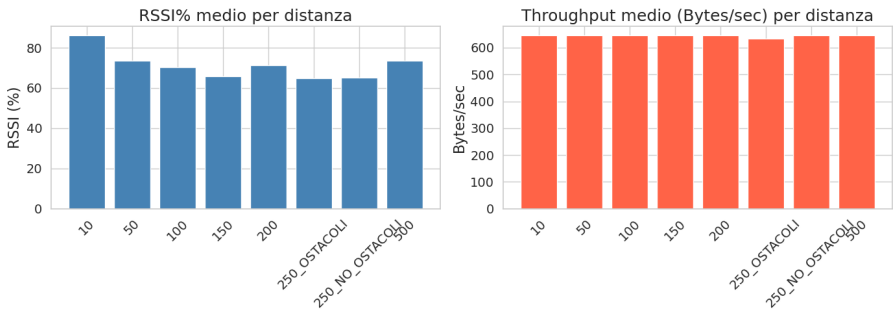



**Figura 6.5:** RSSI e throughput medi in funzione della distanza.

In conclusione, anche nello scenario con ostacoli a 250 m, il degrado delle prestazioni è risultato contenuto e non critico: il throughput medio si è mantenuto sopra i 3.2 pkt/s, dimostrando una buona resilienza del sistema alla degradazione del canale radio.

### Test sulla segmentazione e ricostruzione dei dati

I risultati dei test confermano che il sistema gestisce correttamente il riassemblaggio dei *payload* segmentati.

**Configurazione di default:** Il tempo medio di ricostruzione cresce in maniera approssimativamente lineare con il numero di *chunk*, come mostrato in Figura 6.6.

Come evidenziato in Figura 6.7, la deviazione standard dei tempi di ricostruzione si mantiene relativamente stabile al variare del numero di *chunk*, indicando una buona prevedibilità temporale del processo di riassemblaggio.

Le distribuzioni dei tempi di ricostruzione (Figure 6.8–6.11) mostrano che i tempi di ricostruzione sono concentrati attorno a valori medi, con una leggera variabilità, ma senza picchi significativi.

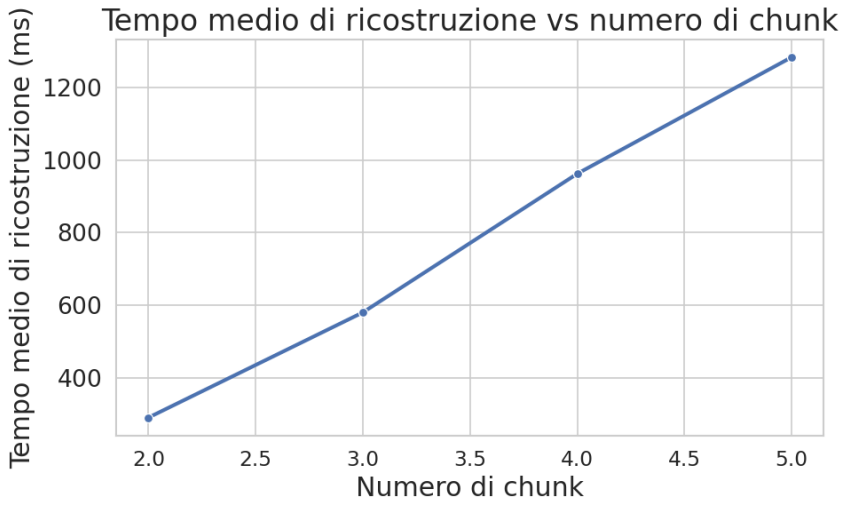



**Figura 6.6:** Tempo medio di ricostruzione per numero di *chunk*.

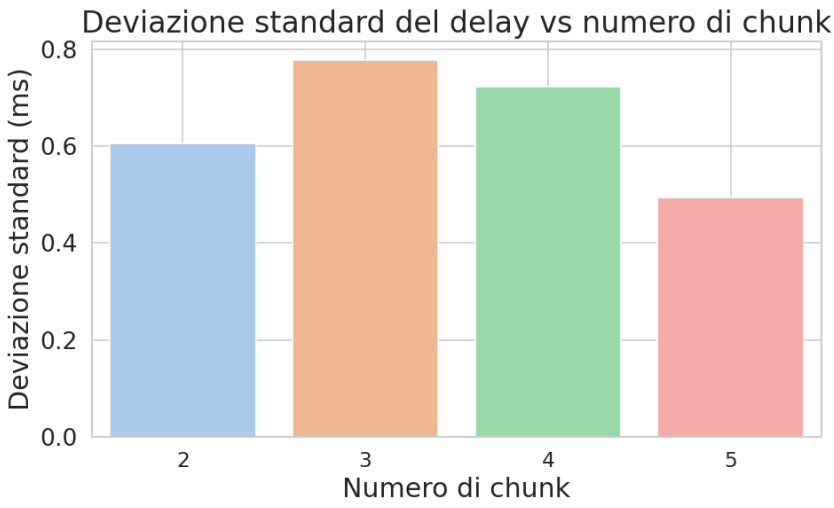



**Figura 6.7:** Deviazione standard dei tempi di ricostruzione per numero di *chunk*.

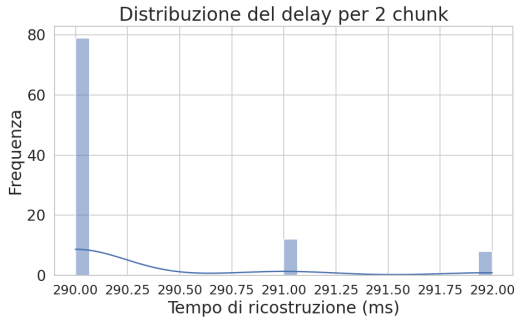



**Figura 6.8:** Distribuzione dei tempi di ricostruzione per 2 *chunk*

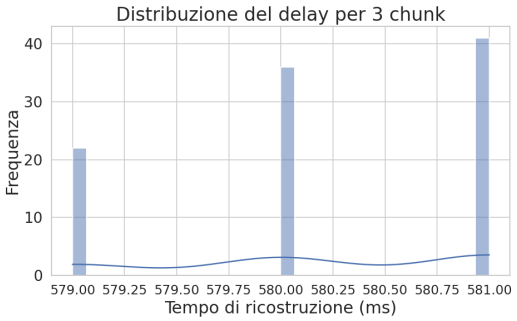



**Figura 6.9:** Distribuzione dei tempi di ricostruzione per 3 *chunk*

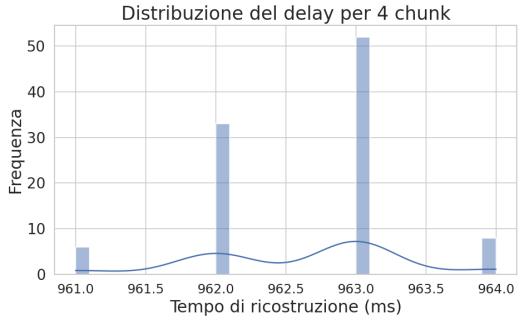



**Figura 6.10:** Distribuzione dei tempi di ricostruzione per 4 *chunk*

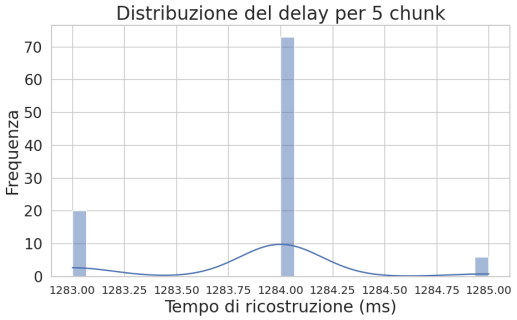



**Figura 6.11:** Distribuzione dei tempi di ricostruzione per 5 *chunk*

**Configurazione personalizzata:** La variabilità dei dati è rimasta contenuta, come si vede nelle Figure 6.13-6.17, confermando una buona stabilità nei tempi di ricostruzione dei *payload* segmentati.

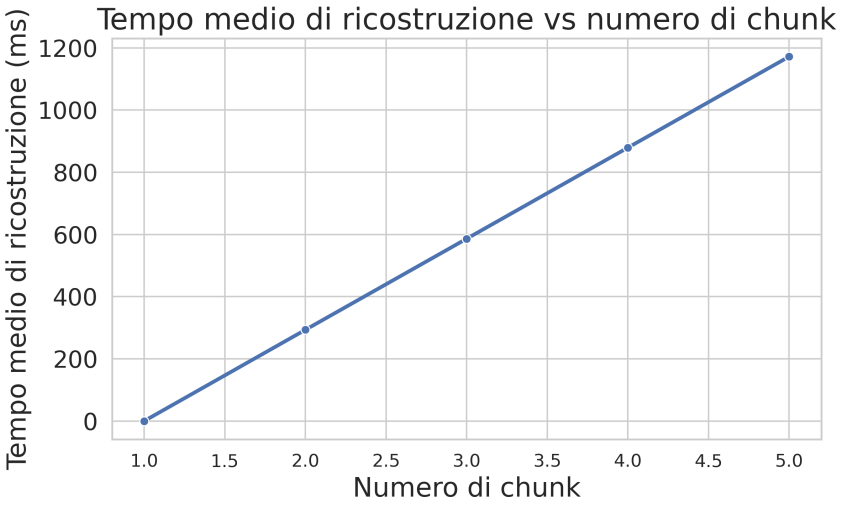



**Figura 6.12:** Tempo medio di ricostruzione per numero di *chunk*.

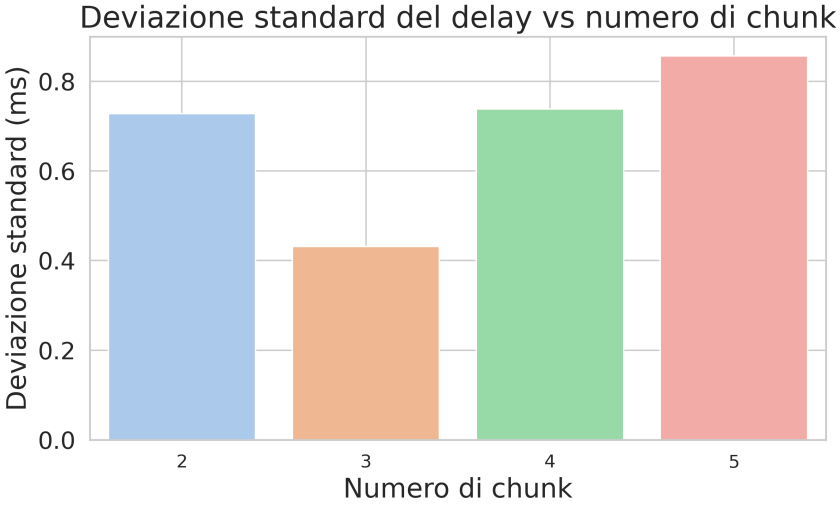



**Figura 6.13:** Deviazione standard dei tempi di ricostruzione per numero di *chunk*.

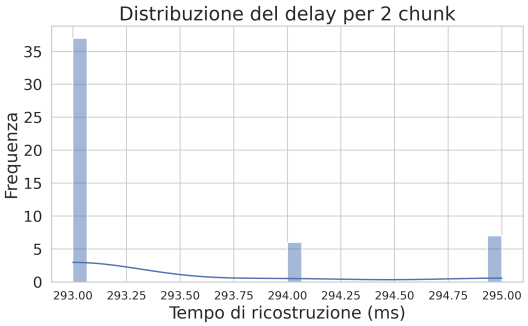



**Figura 6.14:** Distribuzione dei tempi di ricostruzione per 2 *chunk*

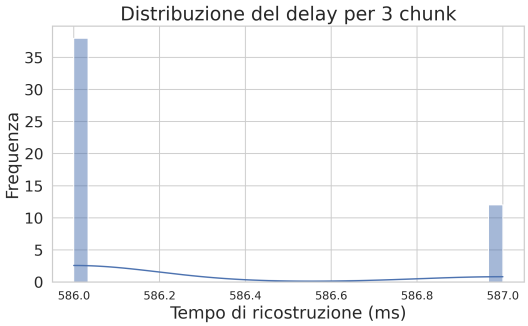



**Figura 6.15:** Distribuzione dei tempi di ricostruzione per 3 *chunk*

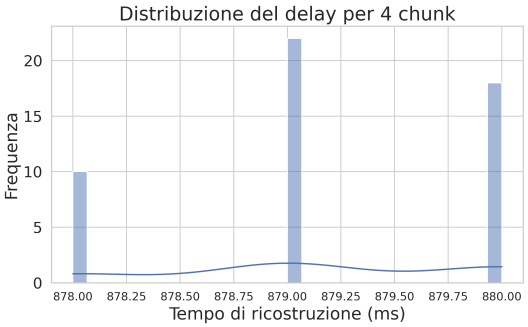



**Figura 6.16:** Distribuzione dei tempi di ricostruzione per 4 *chunk*

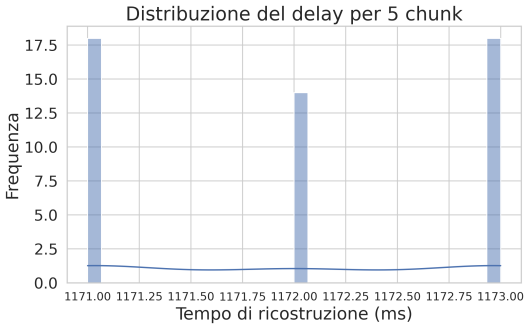



**Figura 6.17:** Distribuzione dei tempi di ricostruzione per 5 *chunk*

**Confronto tra le configurazioni:** Dopo aver analizzato separatamente le due configurazioni, è stato eseguito un confronto diretto. La configurazione personalizzata ha mostrato una riduzione dei tempi di ricostruzione fino all'8.7% per *payload* segmentati in 4 e 5 *chunk*, con penalità trascurabili (inferiore all'1%) sui payload più piccoli. Le Figure 6.18-6.20 mostrano il confronto dettagliato.

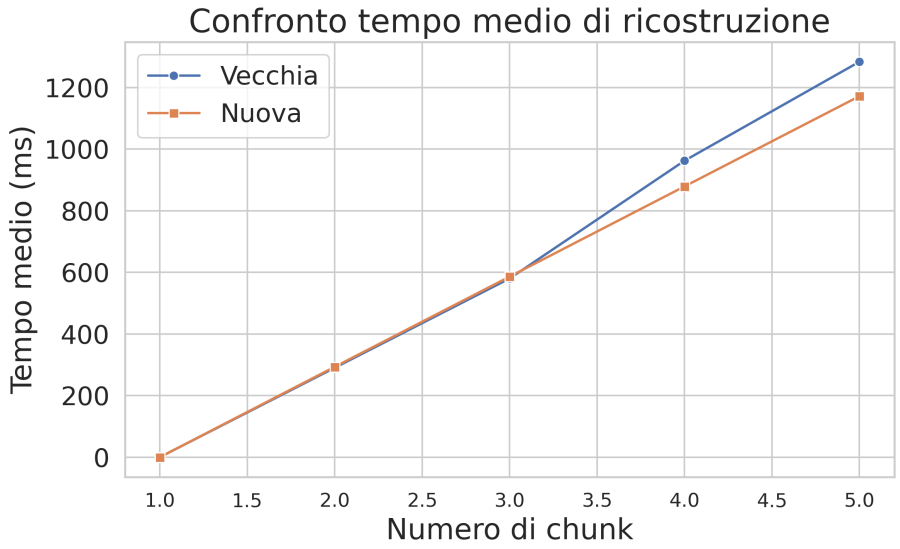



**Figura 6.18:** Confronto del tempo medio di ricostruzione del payload per numero di *chunk*

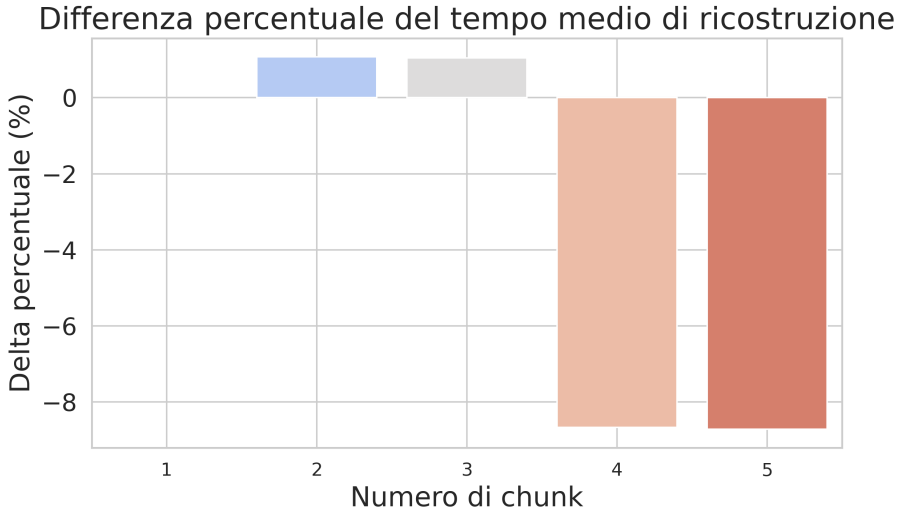



**Figura 6.19:** Differenza percentuale tra i tempi di ricostruzione del payload, configurazione di default vs configurazione personalizzata

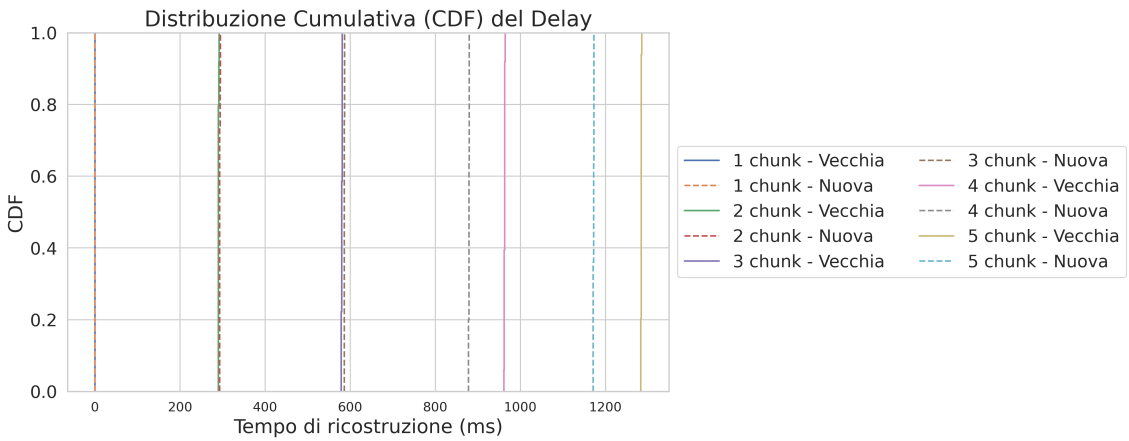



**Figura 6.20:** Confronto delle distribuzioni dei tempi di ricostruzione del payload, configurazione di default vs configurazione personalizzata

### Test del meccanismo di *drop*

Il meccanismo di *drop* si é dimostrato efficace nel gestire l'arrivo di pacchetti non validi o incompleti, evitando l'accumulo di dati parziali, garantendo un'efficace gestione della memoria e delle risorse. La simulazione prevedeva l'invio di pacchetti frammentati, con la perdita di un *chunk* ogni 3 pacchetti, con una percentuale di perdita prevista del 33%. I risultati ottenuti sono riportati nella Tabella 6.4, che conferma la percentuale di perdita dei pacchetti attesa.

| **Tempo** | ***Loss Rate* (%)** | ***Packet Number*** | **Pacchetti ricevuti** | **Pacchetti scartati** |
|---|---|---|---|---|
| 00:00:000 | 33,33 | 3 | 3 | 1 |
| 00:04:359 | 33,33 | 6 | 6 | 2 |
| 00:08:720 | 33,33 | 9 | 9 | 3 |
| 00:13:080 | 33,33 | 12 | 12 | 4 |
| 00:17:440 | 33,33 | 15 | 15 | 5 |
| 00:21:803 | 33,33 | 18 | 18 | 6 |
| 00:26:166 | 33,33 | 21 | 21 | 7 |
| 00:30:528 | 33,33 | 24 | 24 | 8 |
| 00:39:255 | 33,33 | 30 | 30 | 10 |
| 00:43:617 | 33,33 | 33 | 33 | 11 |
| 00:47:980 | 33,33 | 36 | 36 | 12 |
| 00:52:343 | 33,33 | 39 | 39 | 13 |
| 00:56:706 | 33,33 | 42 | 42 | 14 |
| 01:01:069 | 33,33 | 45 | 45 | 15 |
| 01:05:433 | 33,33 | 48 | 48 | 16 |
| 01:09:795 | 33,33 | 51 | 51 | 17 |
| 01:14:158 | 33,33 | 54 | 54 | 18 |
| 01:18:521 | 33,33 | 57 | 57 | 19 |
| 01:22:884 | 33,33 | 60 | 60 | 20 |
| 01:27:248 | 33,33 | 63 | 63 | 21 |
| 01:31:611 | 33,33 | 66 | 66 | 22 |
| 01:35:973 | 33,33 | 69 | 69 | 23 |
| 01:40:336 | 33,33 | 72 | 72 | 24 |
| 01:44:700 | 33,33 | 75 | 75 | 25 |
| 01:49:063 | 33,33 | 78 | 78 | 26 |
| 01:53:425 | 33,33 | 81 | 81 | 27 |
| 01:57:789 | 33,33 | 84 | 84 | 28 |
| 02:02:152 | 33,33 | 87 | 87 | 29 |
| 02:06:514 | 33,33 | 90 | 90 | 30 |

**Tabella 6.4:** Tasso di perdita dei pacchetti durante il test del meccanismo di *drop*.

### Test del riordinamento dei pacchetti

Il test di riordinamento dei pacchetti ha dimostrato che il sistema è in grado di riconoscere e gestire correttamente i *chunk* ricevuti in ordine sparso, riordinandoli in base al loro *Chunk Number* prima della ricostruzione del *payload* originale. I risultati sono riportati nella Tabella 6.5, che mostra l'ordine di arrivo dei pacchetti, e lo stato di ordinamento per ogni sequenza di *chunk*. Come nel test precedente, la simulazione prevedeva l'invio di pacchetti frammentati, con lo scambio dell'ordine di arrivo di due *chunk* ogni 3 pacchetti.

| **Tempo** | ***Packet Number*** | ***Total Chunks*** | ***Chunk* ricevuti** | **Ordinati** | **Ordine di arrivo** |
|---|---|---|---|---|---|
| 00:00:000 | 1 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:03:101 | 2 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:04:652 | 3 | 5 | 5 | NO | 1-3-2-4-5 |
| 00:06:203 | 4 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:07:753 | 5 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:09:305 | 6 | 5 | 5 | NO | 1-3-2-4-5 |
| 00:10:856 | 7 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:12:407 | 8 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:13:958 | 9 | 5 | 5 | NO | 1-3-2-4-5 |
| 00:15:509 | 10 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:17:059 | 11 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:18:610 | 12 | 5 | 5 | NO | 1-3-2-4-5 |
| 00:20:161 | 13 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:21:712 | 14 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:23:264 | 15 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:24:815 | 16 | 5 | 5 | NO | 1-3-2-4-5 |
| 00:26:366 | 17 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:27:916 | 18 | 5 | 5 | NO | 1-3-2-4-5 |
| 00:29:468 | 19 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:31:019 | 20 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:32:569 | 21 | 5 | 5 | NO | 1-3-2-4-5 |
| 00:34:121 | 22 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:35:671 | 23 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:37:223 | 24 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:38:773 | 25 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:40:325 | 26 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:41:875 | 27 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:43:426 | 28 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:44:978 | 29 | 5 | 5 | YES | 1-2-3-4-5 |
| 00:46:529 | 30 | 5 | 5 | NO | 1-3-2-4-5 |

**Tabella 6.5:** Verifica dell'ordine di arrivo dei chunk per ogni pacchetto

### Test di robustezza e affidabilità

Il primo test di robustezza ha permesso di verificare che il sistema è in grado di mantenere una comunicazione stabile e affidabile anche per periodi prolungati, senza cali di prestazioni né perdite di pacchetti.

I risultati mostrano che il sistema ha mantenuto un livello di prestazioni costante per tutta la durata dell'esperimento, evidenziato dalla crescita lineare del numero di pacchetti ricevuti e dalla stabilità del tempo di ricostruzione del *payload* originale, come illustrato nella Figura 6.21.

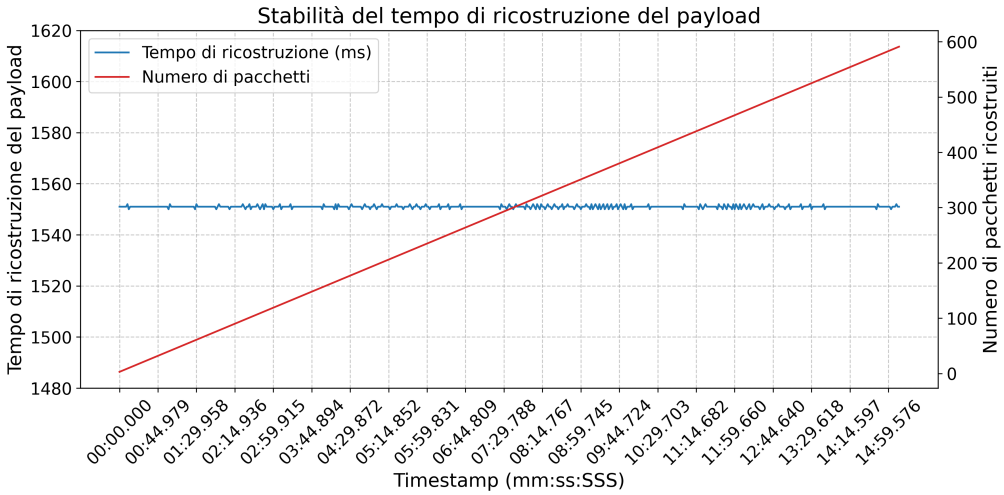



**Figura 6.21:** Stabilità della comunicazione in assenza di interferenza diretta.

Anche nel secondo test, nonostante la presenza di una seconda antenna trasmittente attiva sulla stessa frequenza, il sistema ha mantenuto una buona stabilità della comunicazione, con un numero di pacchetti ricevuti in linea con le attese e senza cali significativi di prestazioni, come mostrato nella Figura 6.22.

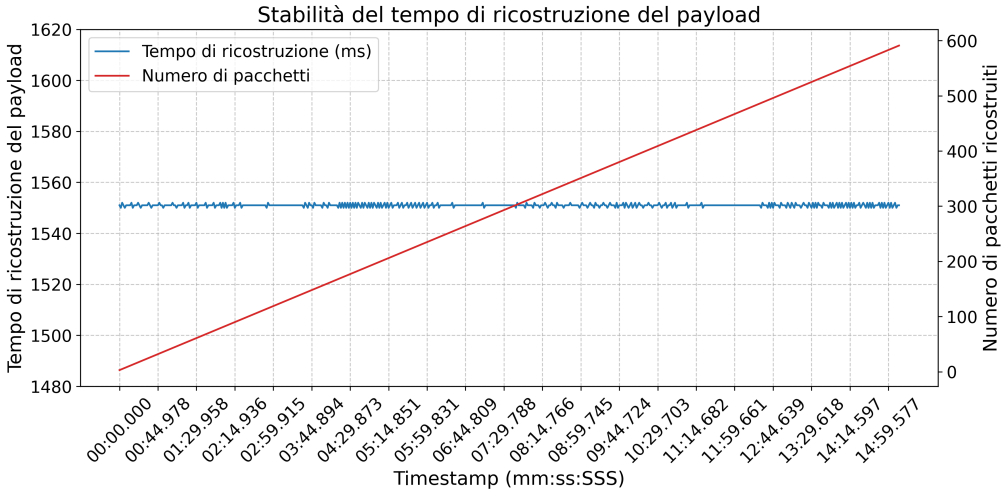



**Figura 6.22:** Stabilità della comunicazione in presenza di interferenza diretta.

Confrontando i risultati dei due test, si osserva che l'introduzione della seconda antenna non ha avuto un impatto significativo sulla stabilità della comunicazione né sulle prestazioni del sistema. Come illustrato nella Figura 6.23, i dati raccolti nei due scenari risultano sovrapponibili.

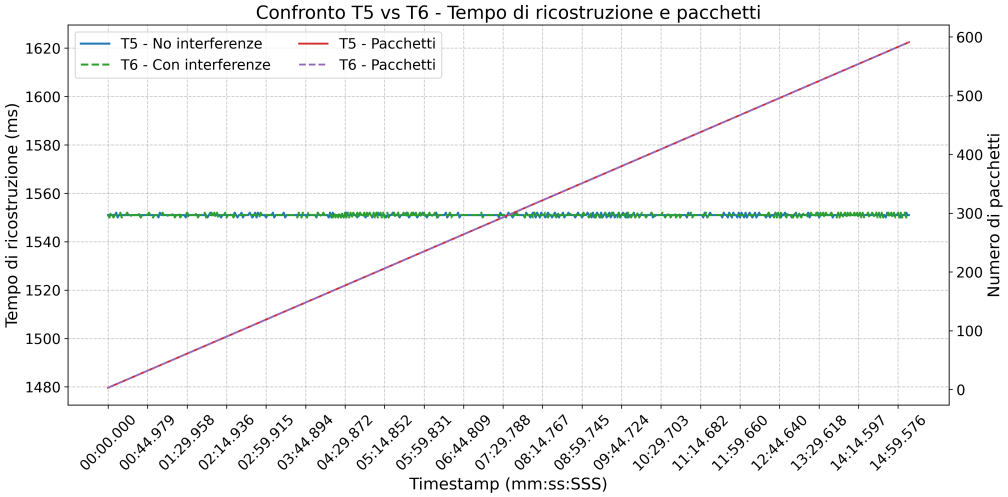



**Figura 6.23:** Confronto tra i due test di robustezza.

L'analisi numerica conferma che le differenze tra i due test sono trascurabili, sebbene rilevabili. Come riportato nella Tabella 6.6, la deviazione standard e la varianza dei tempi di ricostruzione sono leggermente superiori nel test con interferenze, indicando una maggiore variabilità della latenza di ricostruzione del *payload*, imputabile alla presenza di trasmissioni concorrenti.

| **Metrica** | **Senza interferenze** | **Con interferenze** |
|---|---|---|
| Tempo di ricostruzione medio ($\delta_r$) | 1550.99 | 1550.99 |
| Deviazione standard $\delta_r$ | 0.49 | 0.62 |
| Varianza $\delta_r$ | 0.24 | 0.38 |
| Numero di pacchetti ricevuti | 588 | 588 |
| Numero di misure | 589 | 589 |

**Tabella 6.6:** Confronto delle metriche statistiche tra test senza interferenze e con interferenze

## 6.2 Limiti e possibili miglioramenti futuri

Il sistema di telemetria sviluppato ha dimostrato buone prestazioni e un'elevata affidabilità, ma presenta alcuni limiti che potrebbero essere oggetto di miglioramento in sviluppi futuri.

Tra le criticità principali si segnala la necessità di ottimizzare la gestione della latenza nella ricostruzione dei *payload* segmentati, nonché il basso *throughput* dei moduli LoRa© impiegati, che limita la quantità di dati trasmissibili nell'unità di tempo. Tali vincoli derivano dalle caratteristiche intrinseche del protocollo LoRa e dall'architettura a pacchetti frammentati adottata.

Un ulteriore limite significativo riguarda la difficoltà di testare il sistema in condizioni operative reali: il lancio di razzi sperimentali comporta infatti complesse problematiche di natura legale, logistica, organizzativa ed economica, che rendono difficile la pianificazione e l'esecuzione di prove realistiche in volo.

## 6.3 Implicazioni del sistema sviluppato per progetti simili

Il sistema di telemetria sviluppato per il progetto *Borealis* rappresenta una soluzione economica, affidabile e facilmente replicabile, potenzialmente applicabile a un'ampia varietà di scenari nel campo della ricerca e dello sviluppo di razzi sperimentali, ma non solo.

La modularità dell'architettura e l'impiego di tecnologie open-source consentono una facile personalizzazione e integrazione con altri sottosistemi, rendendo il progetto un solido punto di partenza per ulteriori sviluppi o adattamenti.

Il protocollo di comunicazione implementato, basato su segmentazione e ricostruzione di *payload*, estende le capacità di trasmissione offerte dai moduli LoRa©, rendendolo adatto anche ad applicazioni che richiedono l'invio affidabile di pacchetti di dati di medie dimensioni, pur operando in ambienti a banda stretta e bassa latenza.

# Capitolo 7

# Conclusioni

## 7.1 Riepilogo dei risultati ottenuti

Il progetto di telemetria sviluppato per il razzo *Borealis* ha raggiunto tutti gli obiettivi operativi prefissati. I test sperimentali condotti hanno confermato la solidità del sistema in termini di affidabilità, stabilità e resilienza operativa, evidenziando una copertura adeguata, una gestione efficace della segmentazione dei dati e una comunicazione stabile anche in condizioni avverse, quali disallineamento angolare tra le antenne e presenza di interferenze co-canalari.

Il sistema ha dimostrato di essere tecnicamente valido, adattabile e compatibile con i vincoli imposti da contesti aerospaziali a basso budget, confermando la bontà delle scelte progettuali sia hardware che software.

## 7.2 Riflessioni sull'impatto del progetto nell'ambito della ricerca e dello sviluppo di razzi sperimentali

In letteratura sono documentati pochi progetti di telemetria per razzi sperimentali basati su tecnologia LoRa©, la maggior parte dei quali si trova ancora a livello di *proof-of-concept* (AMM22), oppure non risulta adeguata alle esigenze operative reali. Ad esempio, il sistema presentato in (MTS22), pur completo dal punto di vista funzionale, si è dimostrato sottodimensionato sul piano prestazionale, con un raggio operativo limitato a meno di 500 metri, un valore non compatibile con scenari di volo suborbitale o a lungo raggio.

Altri lavori più recenti, come[BKM23;dASBRG19], mostrano dati sperimentali promettenti, ma non approfondiscono lo sviluppo di protocolli di comunicazione né la gestione strutturata dei dati come affrontato nel presente progetto.

A confronto con altri approcci documentati in letteratura, basati su tecnologie differenti come WiFi[MKA24], XBee[KK17] con trasmettitori radio ad alta potenza[LRR17;FM22], il sistema sviluppato per *Borealis* offre un equilibrio ottimale tra portata, consumo energetico e semplicità di implementazione.

L'uso efficiente dei moduli LoRa©, abbinato a un protocollo personalizzato di segmentazione e ricostruzione, consente di ottenere distanze operative di diversi chilometri con requisiti hardware minimi e consumi contenuti, rendendo il sistema preferibile in molte applicazioni embedded in ambito aerospaziale sperimentale.

In questo contesto, il sistema sviluppato per il progetto *Borealis* si distingue per l'approccio metodico, la modularità dell'architettura e la scalabilità, oltre che per l'implementazione di un protocollo di comunicazione flessibile, progettato per estendere in modo efficiente le capacità dei moduli LoRa© anche in condizioni operative critiche. Queste caratteristiche rendono il sistema adattabile e riutilizzabile in una varietà di contesti, contribuendo in modo significativo all'avanzamento dello stato dell'arte nei sistemi di telemetria embedded a basso consumo.

## 7.3 Prospettive di sviluppo futuro

### 7.3.1 Miglioramenti al sistema di telemetria

#### Utilizzo di antenne *patch*

L'utilizzo di un'antenna elicoidale omnidirezionale si é rivelato efficace, tuttavia, per migliorare la copertura e la direzionalitá, e per ridurre le dimensioni del sistema, si potrebbe considerare l'uso di antenne *patch*, da applicare sulla fusoliera del razzo, in questo modo si elimina il problema del mascheramento del segnale dovuto al materiale della fusoliera (alluminio o fibra di carbonio), eliminando la necessità di inserire la finestra in plexiglass utilizzata su Borealis. In questo modo migliorerebbero anche le proprietá strutturali e aerodinamiche del razzo.

#### Utilizzo di moduli LoRa© con maggiore *throughput*

I moduli LoRa© EByte E220-900T22D utilizzati hanno dimostrato buone prestazioni in termini di portata e affidabilità, ma presentano un limite di *throughput* che può essere migliorato utilizzando moduli con capacità di trasmissione dati più elevate. Moduli come i moduli della serie Semtech SX126x offrono un *throughput* maggiore e

una maggiore flessibilità nella configurazione dei parametri di trasmissione, a fronte di un costo superiore e del minore support software rispetto ai moduli EByte.

### 7.3.2 Miglioramenti al protocollo di comunicazione

#### Implementazione di un sistema di *retransmission*

Attualmente il protocollo di comunicazione implementato non prevede un meccanismo di *retransmission* dei pacchetti persi, si potrebeb valutare l'implementazione di un sistema di *retransmission* da coadiuvare al meccanismo di *drop* temporizzato dei pacchetti, in modo da ridurre il numero di pacchetti persi in caso di errori di trasmissione o di congestione della rete. Questo potrebbe essere realizzato implementando un sistema di *acknowledgment* dei pacchetti, tuttavia, bisogna valutare l'impatto che questo potrebbe avere sul *throughput* del sistema.

### 7.3.3 Parsing dei dati e visualizzazione

#### Implementazione di un sistema di visualizzazione in tempo reale

É prevista l'implementazione di una *dashboard*, per poter monitorare il volo del razzo in tempo reale. Si prevede di utilizzare un sistema di visualizzazione basato su *Flask*, che permetterebbe di visualizzare i dati raccolti in tempo reale tramite un'interfaccia web.

# Appendici

# Appendice A

# Librerie

1. **EByte LoRa E220 Series** - Per la gestione dei moduli LoRa© E220.
2. **Adafruit Unified Sensor** - Per le API standard di gestione dei sensori.
3. **Adafruit BNO055** - Per la gestione della IMU.
4. **Adafruit MPRLS** - Per la gestione dei barometri MPRLS.
5. **Adafruit SDFat** - Per la gestione del logging su scheda SD.
6. **nlohmann-json** - Per la gestione di oggetti JSON in C++.

# Appendice B

# Datasheet

1. **Datasheet EByte E220-900T22D** - Datasheet del modulo EByte E220-900T22D.
2. **Datasheet Siretta Delta 5A** - Datasheet dell'antenna omnidirezionale Siretta Delta 5A.
3. **Datasheet Siretta Oscar 3A** - Datasheet dell'antenna Yagi Siretta Oscar 3A.

# Appendice C

# Dati di test

1. **Disallineamento delle antenne** - Dati raccolti durante i test di disallineamento delle antenne.
2. **Test del *throughput*** - Dati raccolti durante i test di throughput del protocollo di comunicazione.
3. **Test di segmentazione e ricostruzione dei dati** - Dati raccolti durante i test di segmentazione e ricostruzione dei dati.
4. **Test del meccanismo di *drop*** - Dati raccolti durante i test del meccanismo di *drop* dei pacchetti.
5. **Test di riordinamento dei pacchetti** - Dati raccolti durante i test di riordinamento dei pacchetti.
6. **Test di robustezza e affidabilità** - Dati raccolti durante i test di robustezza e affidabilità del protocollo di comunicazione.

# Riferimenti bibliografici


[AMM22] Rodrigo Andrade, Alison Moraes, and Felipe Motta. Iot-based sounding rocket telemetry system. In *2022 Symposium on Internet of Things (SIoT)*. IEEE, October 2022.

[BKM23] Marta Brzynska, Krzysztof Klimaszewski, and Krzysztof Martin. A localisation and telemetry lora node for rockets. In *2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, September 2023.

[dASBRG19] Gabriel de Almeida Souza, Larissa Barbosa, Glênio Ramalho, and Alexandre Guarato. Low cost real time rocket telemetry system design. 01 2019.

[FM22] Thomas Francois and Thomas Montano. Design of a radio telemetry system for use by university rocket teams, 2022.

[KK17] Seth Kitchen and Daniel Klinger. Telemetry system for intercollegiate rocket engineering competition vehicle, 2017.

[LRR17] Simón Martín Lombardozzo, Santiago Rodríguez, and Pedro Agustín Roncagliolo. Development of an experimental telemetry system. In *Congreso Argentino de Sistemas Embebidos (CASE)(Universidad de Buenos Aires, 9 al 11 de agosto de 2017)*, 2017.

[MKA24] I Ngethe Muchiri, Yator C Kiplimo, and Shohei Aoki. Implementation of message queuing telemetry transport protocol in model rocket. 2024.

[MLS+24] Yen-Huai Ma, Kuan-Hsun Lee, Tzong-Hann Shieh, Chun-Liang Yeh, and Hsien-Sheng Lin. Applying lora communication technology to uav telemetry system and payload design. In *2024 IEEE International Workshop on Electromagnetics: Applications and Student Innovation Competition (iWEM)*. IEEE, July 2024.

[MNBB24] Alireza Maleki, Ha H. Nguyen, Ebrahim Bedeer, and Robert Barton. A tutorial on chirp spread spectrum modulation for lorawan: Basics and key advances. *IEEE Open Journal of the Communications Society*, 5:4578–4612, 2024.

[MTS22] Misbahuddin Misbahuddin, Muhamat Taufik, and I Made Budi Suksmadana. Lora sv611-based communication system to monitor behaviour of rocket using inertial measurement unit sensor. *International Journal of Electrical, Energy and Power System Engineering*, 5(2), June 2022.

[SB06] Günther Seibert and Bruce T Battrick. *The history of sounding rockets and their contribution to European space research.* ESA Publications division Noordwijk, 2006.