6G for Connected Sky
“6G-SKY”

Work Package 5

Integration, Demonstrations and Lab Emulations

Deliverable 5.2 –

Demonstrators, Reports of Results and Videos

6G-SKY Project

2025-06-12

for full publication

[6G-SKY]

[D5.2 – Demonstrators, Report of Results and Videos

Editor: Christian Raffelsberger, Lakeside Labs GmbH

Reviewer: Irshad A. Meer; Ozan A. Topal, KTH Royal Institute of Technology

Project coordinator: Dominic Schupke, Airbus Defence and Space GmbH

Technical Project Coordinator: Cicek Cavdar, KTH Royal Institute of Technology

CELTIC published project result

Ericsson AB	Ericsson Hungary
Ericsson Antenna Systems	PTS
KTH	Deutsche Telekom
SAS	Lakeside Labs
Fraunhofer IIS	Logistik Center Austria Süd
RED Bernard	twins
AITIA	Skysense
Meshmerize	Motius
Airbus	

Disclaimer

This document contains material, which is the copyright of certain PARTICIPANTS, and may not be reproduced or copied without permission.

All PARTICIPANTS have agreed to full publication of this document.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the PARTICIPANTS nor CELTIC-NEXT warrant that the information contained in the report is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

Table of Contents

Document History

Version	Date	Author(s)/Reviewer	Comment
V0.1	18.11.2024	C. Raffelsberger (LAKE)	Document Template
v1.0	12.06.2025	M. Sende (LAKE), C. Raffelsberger (LAKE), J. Münzberg (RED), B. Lackner (TWINS), V. Ortman (SKYSENSE), P. Gradin (SKYSENSE), J. Kolb (Fraunhofer IIS), V.-M. Riepula (DT), A. Baltaci (Airbus)	Final Version

Executive Summary

The 6G-SKY project conducted a series of demonstrations to showcase its key achievements and to gather real-world performance data essential for system validation. This report summarizes the final demonstrator architectures and measurement results from lab emulation and real-world evaluations.

During the course of the 6G-SKY project, several modular drone platforms were developed and used across different demonstrations. They provide custom mount points and antenna configuration to support TN and NTN communication.

Lab emulation enabled early validation of mesh, 5G and 5G-NTN technologies within a multi-technology, -path, -link communication architecture. This allowed for systematic analysis of handovers and communication quality. A custom 5G Sidelink implementation for D2D communication was evaluated and compared with a mesh technology based on Wi-Fi, optimized for mobility. 5G NTN UE tests simulated data transmissions with a GEO satellite using a 5G Narrowband IoT link via an IP tunnel. A hybrid TN/NTN network integrating a High-Altitude Platform and satellite backhaul was tested. Handover between terrestrial and non-terrestrial links (voice and data) succeeded with 100% reliability. Voice calls remained stable, and latency figures aligned with expectations (<200 ms TN, >500 ms NTN). Round-trip-time spikes during handover remained below 1 second, confirming consistent performance.

A multi-link setup incorporating Wi-Fi mesh, 5G campus networks, and LEO satellite connectivity was demonstrated. All technologies were integrated on a drone, with Multipath TCP managing real-time link selection. While MPTCP ensured redundancy, it caused data spikes due to buffering, limiting its suitability for high-data-rate applications like 10 Mbit/s video streaming. Lower-rate traffic performed reliably, and future protocols (e.g., MP-DCCP, MP-QUIC) may better serve latency-sensitive applications.

An explainable AI framework for drone (UAV) handover decisions was developed. It combines deep reinforcement learning, SHAP-based feature attribution, and large language models to generate interpretable decisions. Tested on real LTE flight data, it enables users to understand selected and rejected handovers through a natural language interface. Initial expert feedback on the tool has been positive, and a formal user study is planned.

A drone swarm equipped with Wi-Fi mesh and 5G connectivity was deployed for autonomous container inspection. The system uses ROS2 for task allocation and image capture, with cloud offloading for analysis. Initial scalability issues caused by ROS2 service discovery were mitigated through middleware bridging. The swarm achieved reliable operation and detection accuracies of up to 90% at the LCA logistics center in Austria. A second use case at the same site demonstrated how a drone can act as a mobile relay, supporting live multimedia data streaming from an autonomous ground vehicle in obstructed environments. Using a Wi-Fi mesh, the drone ensured stable, low-latency connectivity between the rover and control station along the entire mission path.

Airspace safety was addressed through a sense and avoid demonstration. A 6G-SKY drone autonomously reacted to an intruding drone based on proximity data extracted by a ground sensor. The intruder's telemetry was passively intercepted and transmitted via MQTT to the friendly 6G-SKY drone, which paused or landed based on configurable safety radii. The system functioned in real time with a 100% detection rate.

Finally, a HAPS networking testbed was established to evaluate high-frequency and multi-connectivity communication links critical to future 6G systems. While flight trials and KPI evaluations are ongoing, the integrated setup is ready for testing HAPS-Ground, HAPS-UAV, HAPS-HAPS, and HAPS–Satellite links.

List of Authors

Name	Affiliation
Aygün Baltaci	Airbus
Petter Gradin	Skysense
Julia Kolb	Fraunhofer IIS
Bernd Lackner	Twins
Veli-Matti Riepula	Deutsche Telekom
Julian Münzberg	RED Bernard
Victor Ortman	Skysense
Christian Raffelsberger (Editor)	Lakeside Labs
Micha Sende	Lakeside Labs

1 Introduction

1.1 Objective of the Document

This deliverable, *D5.2 Demonstrators, reports of results and Videos,* presents a comprehensive overview of the demonstration activities carried out within the scope of the 6G-SKY project. It documents the outcomes of each demonstration case, highlighting the motivations, the experimental setup, and the targeted performance metrics. Furthermore, each case includes success criteria and the specific 6G features under evaluation. This report details the procedures followed during the demonstrations, the environments in which they were conducted, and the partner contributions. Both hardware and software components involved in each demonstration are identified. The accompanying videos serve to visually support and validate the reported findings.

1.2 Structure of the Document

The further structure of the document is as follows: In Chapter 2 we give a short overview of the 6G-SKY work package 5 and the overall demo scenarios. Chapter 3 is dedicated to hardware components such as drones and antennas that are used for several demonstrators. Chapter 4 to Chapter 8 describe the demonstration cases in detail:

- Chapter 4: Task 5.2: Lab Emulation
- Chapter 5: Task 5.3: Multi-technology network integration
- Chapter 6: Task 5.4: 3D network demonstration with multiple drones flying as a swarm in coordination
- Chapter 7: Task 5.5: Demonstration sense and avoid mechanisms
- Chapter 8: Task 5.6: Demonstration of HAPS networking

Each chapter describes motivation, setup, goals (including metrics and success criteria), schedules and procedures. Chapter 9 describes contributions to sustainability aspects before Chapter 10 concludes this deliverable. The annex contains the Glossary, List of Figures and List of Tables.

2 WP5 Overview

The overall goal of this work package WP5 is to integrate the components developed in the work packages WP1 to WP4 into functional demonstrators as well as to demonstrate the functionality of these components and the 6G-SKY architecture on a system level. This is accomplished in different environments depending on the underlying use cases. Necessary software and hardware (e.g., drones, or antenna modules) are developed. Test procedures, performance metrics and success criteria for each test/demonstration are applied to the testing and demonstrations. Particular objectives with associated tasks within WP5 are outlined as follows:

- Development of dedicated UAV hardware and antenna modules for use in the tests (Task 5.1).

- In-lab interoperability testing and validation of communication links and applications as used in Tasks 5.3 and 5.6 (Task 5.2).
- Evaluation and Proof-of-Concept of the results from WP2, WP3 and WP4. Multi-Technology Connectivity Links and the resilience of the adaptive multi-technology network are tested (Task 5.3).
- Proof of concept demonstration of networking and swarming technology with a real swarm of drones applied to the mobility use case defined in WP1 (Task 5.4).
- Demonstration of safety in Urban Air Mobility and U-space by providing "see & be seen" capability to all types of low flying aircraft including manned aircraft, collaborative drones and non-collaborative drones (Task 5.5).
- Test and demonstration of HAPS-Ground, HAPS-low altitude UAV, HAPS-HAPS and HAPS – Satellite links (Task 5.6).

3 Task 5.1: Development of dedicated hardware modules (UAVs, Antennas)

Lead: Twins, **Contributors:** Lakeside Labs, Meshmerize

3.1 Subtask 5.1.1: UAV Platform

As part of the system development process, three *twinFOLD GEO* drones (see Figure 1) were individually constructed and carefully adapted to support the specialized hardware required for the intended aerial operations. Each drone underwent a series of modifications to ensure compatibility with the mission's technical and functional requirements.

To facilitate the integration of on-board computing and imaging systems, a modular mounting plate was custom-designed and securely fitted to the underside of each drone. This modular design not only provided a robust structure for mounting but also allowed for easy installation, removal, and reconfiguration of various hardware components. Most importantly, it served as the mounting point for the companion board, a crucial element responsible for in-flight navigation (serial connection to the flight controller) and communication as well as image processing and data handling.

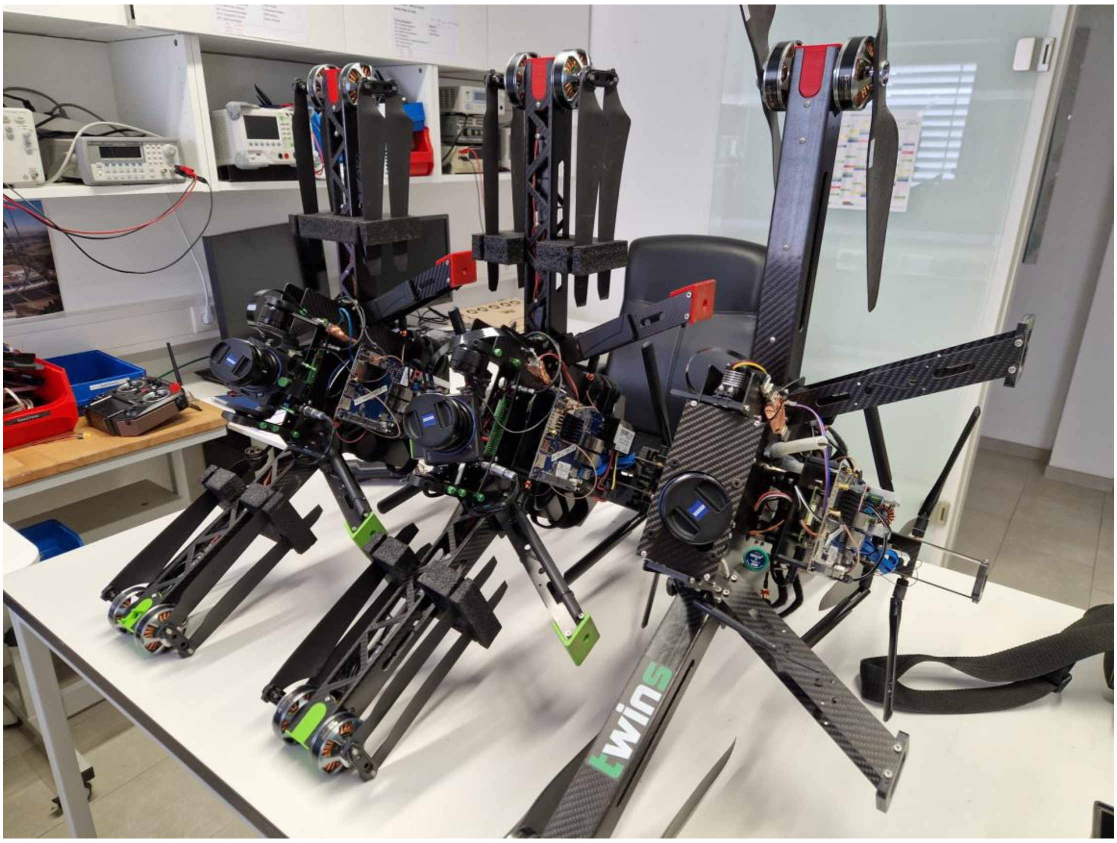


Figure 1. twinFOLD GEO drone platform

In order to capture high-quality aerial imagery with flexible orientation control, each drone was equipped with a two-axis gimbal. The cameras were mounted directly onto these gimbals, enabling them to pivot smoothly along both pitch and yaw axes. This setup allowed for real-time control of the camera orientation during flight, ensuring that specific angles and viewpoints could be captured accurately, even while the drone was in motion or subjected to environmental disturbances such as wind.

To enhance the overall data management workflow, an interface was implemented between the camera systems and the companion board PC. This interface enabled direct access to the cameras' on-board storage, thereby allowing the companion computer to retrieve, store, and transfer image data to the processing server midair. This capability significantly improved operational efficiency by reducing the need for manual data handling after flight and supported real-time analysis capabilities.

Furthermore, to ensure uninterrupted communication during drone operation, the remote control (RC) antennas were strategically repositioned. This adjustment was made to reduce the risk of radio frequency interference between the RC communication system and the mesh network used for inter-drone or ground-to-drone communication. By minimizing such disturbances, the stability and reliability of both control signals and data transmission were greatly enhanced, supporting smoother and safer flight operations. Figure 2 shows the integrated platform.

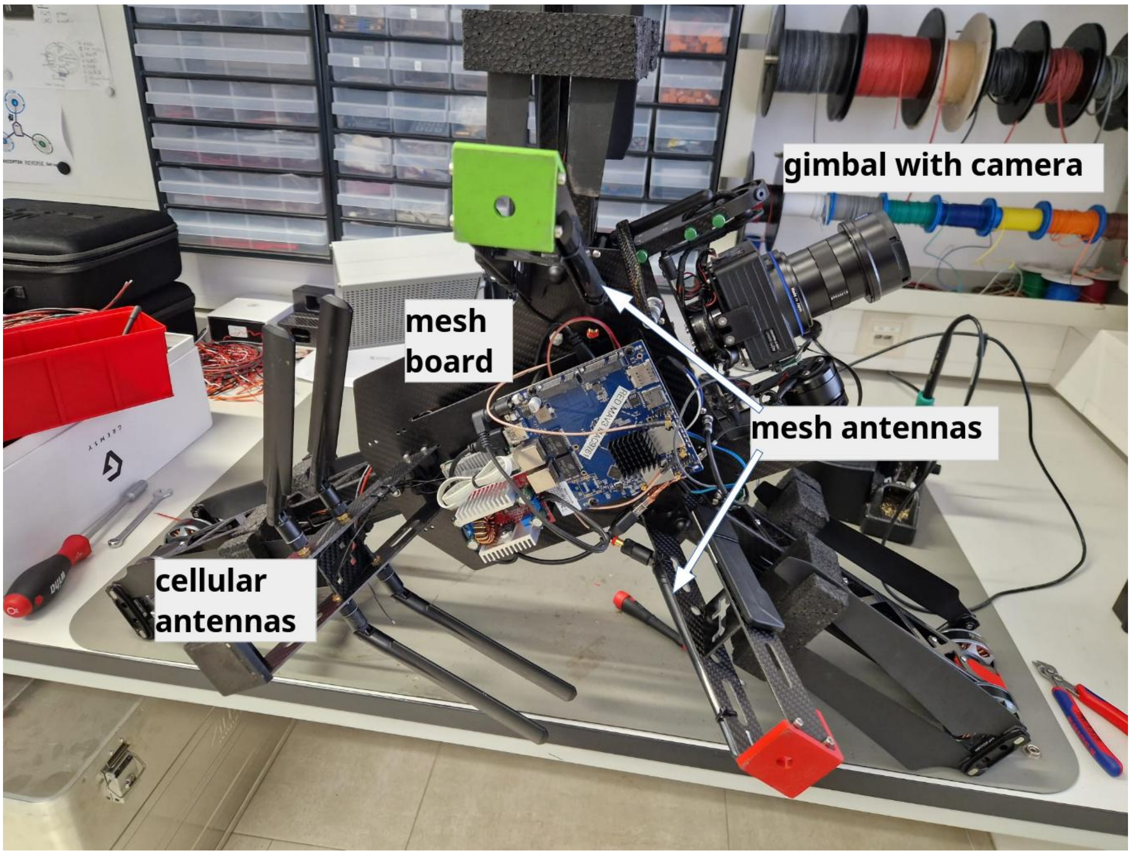


Figure 2. twinFOLD GEO drone platform with additional components for a demonstration

3.2 Subtask 5.1.2: Communication and Antenna Module Design

Each drone was equipped with a Wallys DR4019 embedded board to enable mesh-based communication within the swarm. The key specifications of the DR4019 include:

- SoC: Qualcomm IPQ4019 (Quad-core ARM Cortex-A7, 716.8 MHz)
- Wi-Fi: 802.11ac (Wi-Fi 5), 2x2 MIMO
- Radios: Dual-band support (2.4 GHz and 5 GHz)

The DR4019 boards ran a custom firmware developed by Meshmerize. Communication between drones was established over the 5 GHz band using a 20 MHz channel bandwidth to balance throughput and range and to avoid interference with other on-board modules operating in the 2.4 GHz band. The 5 GHz radio was connected to two standard omni antennas, which were mounted parallel to two of the legs of the drone.

Each DR4019 board was connected to a Raspberry Pi 4 companion board (not visible on Figure 2 as it is below the mesh board) via Ethernet. Similarly, the ground control station was connected to a DR4019 board over Ethernet. The Ethernet interfaces were bridged with the wireless mesh network, creating a single Layer 2 domain. This setup allowed seamless, transparent communication between the drones and the ground controller, enabling efficient command-and-control as well as telemetry exchange across the network.

4 Task 5.2: Lab Emulation

4.1 Overview

Lead: Fraunhofer IIS, **Contributors:** Deutsche Telekom, Airbus, Meshmerize, Ericsson Germany, Lakeside Labs

4.1.1 Scenario Description

The scope of Task 5.2 is the in-lab interoperability testing and validation of communication links and applications used in Tasks 5.3 and 5.6. The lab emulation task aims to integrate, test, and validate the communication equipment and applications in the lab under reproducible conditions before performing the over-the-air tests with drones, HAPS, and satellites in Tasks 5.3 and 5.6.

4.1.2 Setup and Environment

The principal setup is depicted in Figure 3. The lab setup is based on a commercial channel emulator (Keysight F64, including aerospace options for GEO, MEO, and LEO satellites) for hybrid scenarios, which reproduces the propagation conditions from various links (satellite, HAPS, A2G, A2A, V2V) in real-time. Integration and interoperability testing of the communication links (e.g., based on the software defined radio 5G RAN stack OpenAirInterface (OAI) or commercial wireless devices) is possible in the lab without the need for expensive satellite, HAPS capacity, or drone operations, as all link characteristics, such as delay and frequency drifts, multi-path transmissions, and interference, can be emulated. Additionally, testing equipment, such as a cellular communication tester (Keysight UXM 5G Wireless Test Platform), is available for detailed analysis of a UE regarding performance and specification compliance. The test site is located in Erlangen, Germany.

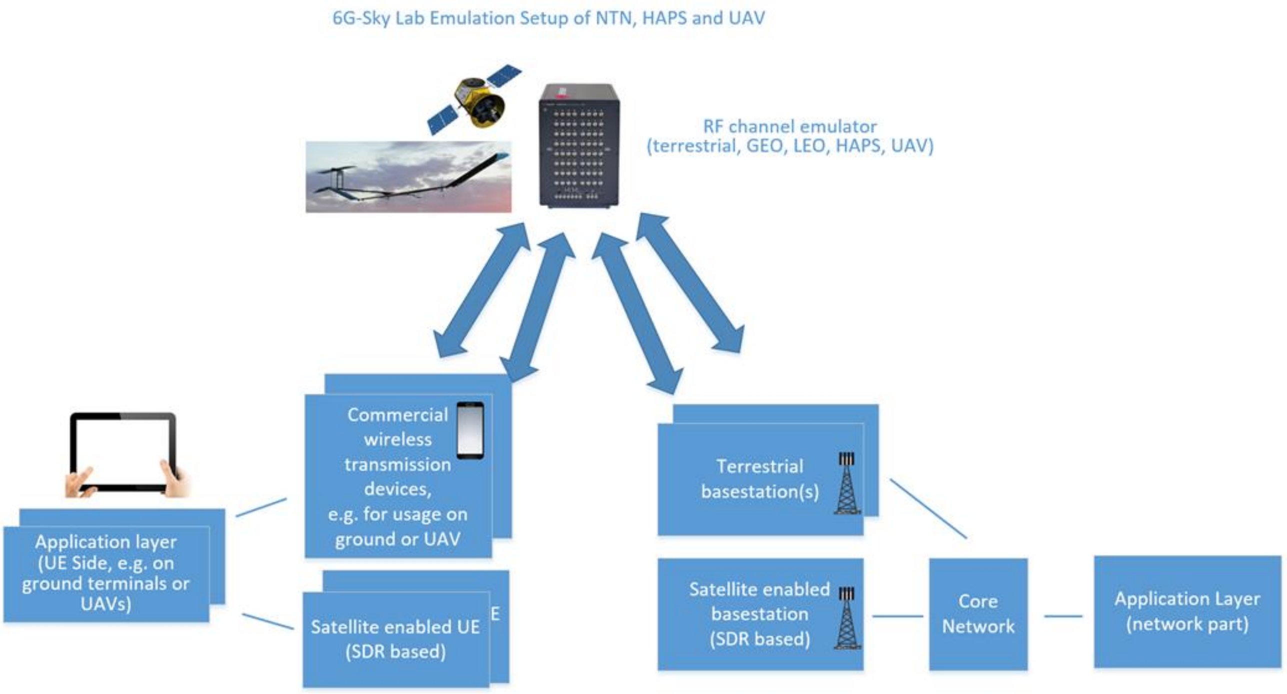

6G-Sky Lab Emulation Setup of NTN, HAPS and UAV
RF channel emulator
(terrestrial, GEO, LEO, HAPS, UAV)
Application layer
(UE Side, e.g. on ground terminals or UAVs)
Commercial wireless transmission devices,
e.g. for usage on ground or UAV
Satellite enabled UE
(SDR based)
Terrestrial basestation(s)
Satellite enabled basestation
(SDR based)
Core Network
Application Layer
(network part)

Figure 3. Lab setup at Fraunhofer IIS for emulation of testing of link performance in terrestrial and non-terrestrial scenarios

4.1.3 Goals, Performance Metrics and Success Criteria

The primary goal is to showcase the integration of the partners' technologies in mesh, 5G, and 5G-NTN into a multilink-technology and multipath communication system to support data transfer within a drone swarm and between the swarm and the application server on ground using NTN support.

Once the communication was stable, the maximum end-to-end (E2E) throughput in downlink and uplink were measured, along with the round-trip delay. The quality and stability of communication within the drone swarm and between the swarm and the ground network via NTN satellite connection were assessed.

The experiment is considered successful if we showcase stable data transmission with expected throughput and latency over several minutes (demo duration).

4.2 Experiment 1: Two-hop mesh communication

4.2.1 Experimental Setup

The experiment focuses on a hybrid network combining 5G terrestrial and meshed Wi-Fi networks (operating on 2.4GHz or 5GHz) to establish interconnections among the drones. In this context, the term “drone” refers to the role or configuration of each node in the network emulation. Although no actual drones were used during this phase, the communication hardware corresponds exactly to that intended for drone flights. The two primary use cases explored are:

- **Range Extension**: Involving direct communication between two drones or sequentially extending the range to the furthest drone. For the range extension, Meshmerize Wi-Fi mesh is run on a DR4019 board, connected via Gigabit Ethernet to a Raspberry Pi 4 (RPi4), while the 5G User Equipment (UE) runs on the 5G RM500Q USB Modem. This replicates the communication setup of the drone described in Chapter 3.2.
  Additionally, 5G Sidelink is implemented using the Software Define Radio (SDR) approach based on OAI on an Intel NUC PC to test D2D Communication in the 3GPP Standard. In Sidelink, we tested the data transmissions on the lower 5G protocol stack layers only (without mesh). This setup was not replicated on a drone but only evaluated in lab.
- **Dynamic link switching in the mesh network**: Allowing drones to maintain connectivity within a purely mesh environment, where link changes happen dynamically.

The initial phase involves testing single links to ensure stable performance. Following this, we will evaluate the integration of these single links with handover capabilities between the Wi-Fi mesh network and 5G.

A visualization of the network topology is provided in Figure 4.

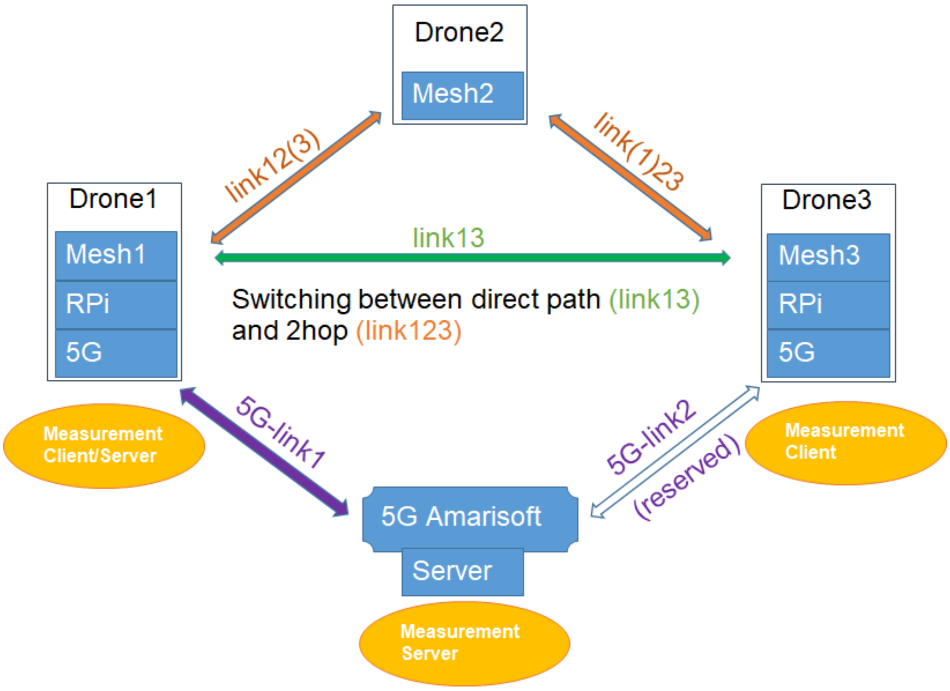


Figure 4. Setup for testing two-hop communication via drone-to-drone links and 5G connectivity to ground

Drone Configuration

Drone 1 carries a Meshmerize mesh-network board (DR4019) connected to a RPi4 via Gigabit Ethernet, along with a 5G UE (Quectel RM500Q) connected via a USB 3 interface to the RPi4.

Drone 2 serves as an optional repeater if Link 13 (drone 1 to drone 3) is unavailable, but Links 12 (drone 1 to drone 2) and 23 (drone 2 to drone 3) remain within reachable distance.

Drone 3 utilizes a RPi4 and a Meshmerize mesh-network router connected via Gigabit Ethernet, along with a 5G UE (Quectel RM500Q) connected via USB 3 for access to the 5G terrestrial TN network.

To test direct communication in 5G Sidelink, we are utilizing the implementation of 5G Sidelink (Release 16) developed by Fraunhofer IIS on the lower protocol layers of the 5G stack, as no commercial equipment for ad-hoc networks in 5G was available. This implementation is based on Software-Defined Radio (SDR) and is programmed using OpenAirInterface. We are employing Intel NUCs and USRP devices as our SDR components to facilitate signal transmission across the frequency range.

Network Infrastructure

For the 5G TN gNB, a computer with Amarisoft SW together with an AW2S Radio Unit (band n78), featuring 4x4 MIMO and 100 MHz bandwidth, is used (Fraunhofer IIS 5G mobile campus network equipment).

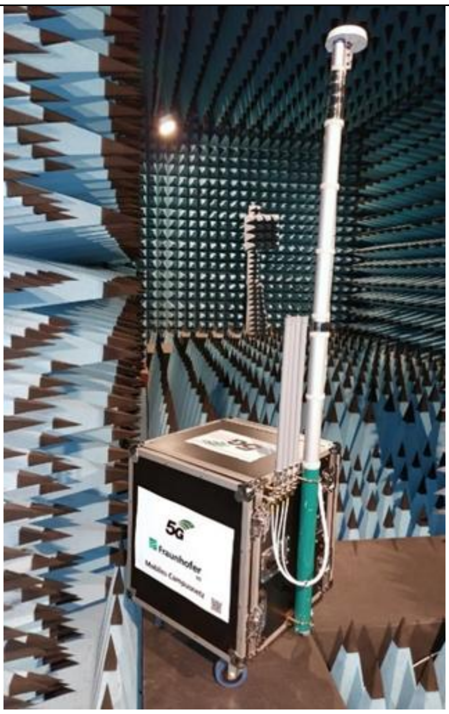
The UE can be served by a direct terrestrial 5G link, while the Wi-Fi mesh network can bridge communications between all drones. If a direct Wi-Fi connection between two drones becomes impossible (e.g., due to distance), the additional drone can act as a repeater, relaying information to the destination that would otherwise be out of reach.

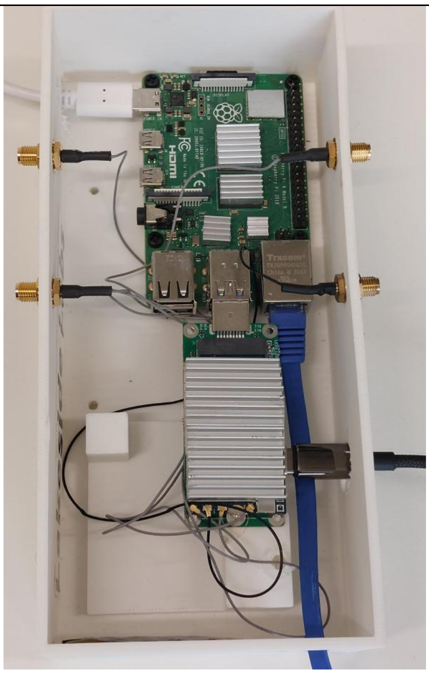
The project partners Fraunhofer IIS, Meshmerize, Lakeside Labs, and Airbus have successfully integrated the various technologies, enabling Fraunhofer IIS to carry out the tests for Task 5.2.

Hardware List with Description and Visualization

The following HW-Elements are used in the tests:

- Terrestrial 5G gNB (Amarisoft gNB and AW2S-RU 4x4 MIMO 100 MHz) with integrated Terrestrial 5G Core Network (Amarisoft)
- Meshmerize router (Wi-Fi (2.4 GHz or 5 GHz) with 2 antennas each)
- 5G UEs (Quectel RM500 Modems connected via USB3 to Raspberry)
- Raspberry Pi to control Meshmerize Router and 5G-Modem (UE)
- 5G UEs (Sidelink on Intel NUC)

Fraunhofer 5G gNB (Mobile campus network): • Band n78 • Frequency: 3750MHz (3700MHz-3800MHz) • Bandwidth: 100 MHz • Time Division Duplex (TDD) • 4x4 MIMO • downlink_slots: 7 • downlink_symbols: 6 • uplink_slots: 2 • uplink_symbols: 4 • subcarrier_spacing: 30 kHz • ssb_pos_bitmap: "10000000" • period: 5 ms • prach_config_index: 16 (subframe 1 every frame) • tx_pwr: 20dBm • ant_gain: 3dBi • Linux 4.15.0-192-lowlatency iperf 3.14	
Meshmerize Mesh-router MM Router is configured for interaction with Lakeside Labs companion board and for handover. It is able to establish a mesh-network. **Parameters** • Wi-Fi-router DR40X90 • Wi-Fi 2,4GHz channel11 (ht-20) • Wi-Fi 5.0GHz channel36 (vht-40) • 2 antennas Wi-Fi 2.4GHz • 2 antennas Wi-Fi 5.0GHz • Chipset IPQ4029 • 2x2MIMO 2.4GHz high power radio module • 2x2MIMO 5GHz high power radio module • Frequency 2.400-2.482GHz • Frequency 4.940-5.825GHz • 5MHz/10MHz/20MHz/40MHz/80MHz bandwidth • Support Wi-Fi 802.11ABGN/AC	

Lakeside Labs Raspberry Pi 4 companion board to control MM Router and 5G UEs • Configured for automatic 5G connection to core network & for interaction with MM-Router • Lakesiderouter1: Linux geo-red 5.15.0-1059-raspi #62-Ubuntu iperf 3.9 (cJSON 1.7.13) TCP (Quectel UE RM500 (4x4MIMO)) • Lakesiderouter2: Linux geo-blue 5.15.0-1055-raspi #58-Ubuntu iperf 3.9 (cJSON 1.7.13) TCP (Quectel UE RM500 (4x4MIMO)) • **Supply Voltage Range:** 5V **Power Consumption (Typical):** 15W	
5G-Sidelink implementation for D2D Communication • PHY Layer + MAC layer + RRC layer • mode 2 (out of coverage) • Band n47 5.855-5.925 GHz • MCS9 • 30 kHz sub-carrier spacing, • 10 MHz bandwidth • 4 Sidelink slots per frame • 2 USRP B205mini-i connected to • 2 Intel NUC • connected cables setup with good SNR	

4.2.2 Evaluation results

This chapter systematically presents both the testing methodology and experimental results.

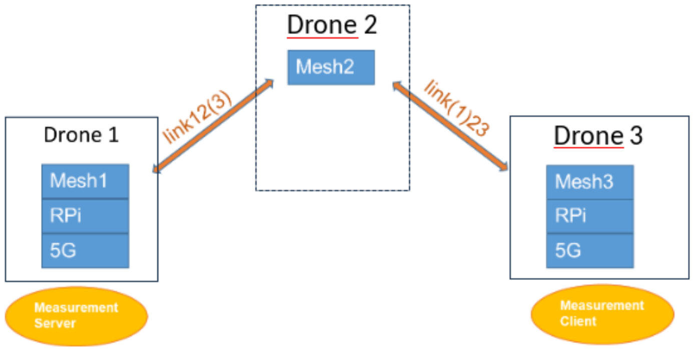
1.Range extension

3 Wi-Fi nodes are connected sequentially in a mesh network.

- Measurement results: link123
- Measurement server: Drone 1
- Measurement Client: Drone 3

2.4 GHz (channel 11)	5 GHz (channel 36)
Downlink rate: **16** Mbit /sec	Downlink rate: **103** Mbit/sec
Uplink rate: **16** Mbit / sec	Uplink rate: **101** Mbit/sec
Round Trip Time: **13** msec	Round Trip time: **13** msec

2.4 GHz band: ht20; 5 GHz band: vht-40



The data is transmitted from the drone 1 to drone 3 via drone 2 and vice versa.

2.Dynamic link switching

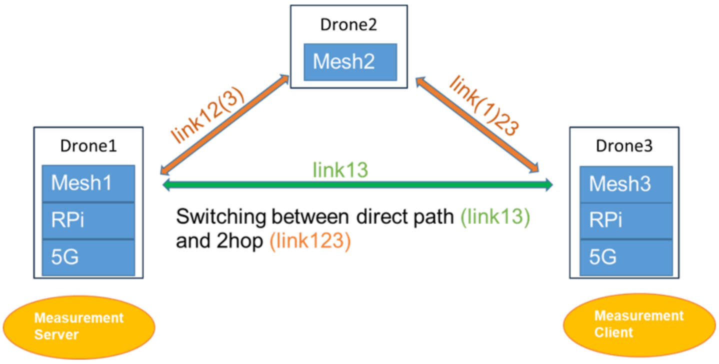
Measurements results for the link 13

2.4 GHz (channel 11)	5 GHz (channel 36)
Downlink rate: **40** Mbit /sec	Downlink rate: **163** Mbit/sec
Uplink rate: **40** Mbit / sec	Uplink rate: **172** Mbit/sec
Round Trip Time: **7** msec	Round Trip time: **7** msec

After switching link 13 to link 123 (one hop)

2.4 GHz (channel 11)	5 GHz (channel 36)
Downlink rate: **15** Mbit /sec	Downlink rate: **90** Mbit/sec
Uplink rate: **15** Mbit / sec	Uplink rate: **80** Mbit/sec
Round Trip Time: **12** msec	Round Trip time: **14** msec

2.4 GHz band: ht20; 5 GHz band: vht-40



After switching from direct link 13 to 123 (with one hop over drone 2) in channel 11, we observe a significant drop in throughput performance from 40 Mbit/s to 15 Mbit/s, with an increase in latency of almost 80%. By analogy, the performance is worsening in a similar way on channel 36.

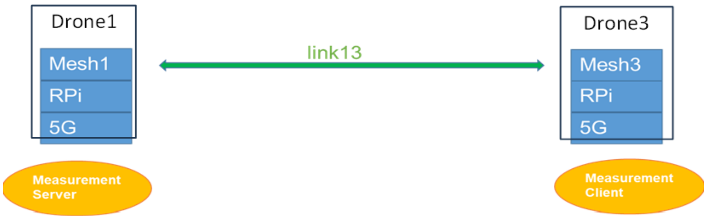
3.Range extension direct Wi-Fi vs. 5G-Sidelink (in Open Air Interface)

A. Measurements for the link 13 (Wi-Fi)

2.4 GHz (channel 11)	5 GHz (channel 36)
Downlink rate: **37** Mbit /sec	Downlink rate: **103** Mbit/sec
Uplink rate: **37** Mbit / sec	Uplink rate: **101** Mbit/sec
Round Trip Time: **9** msec	Round Trip time: **13** msec

2.4 GHz band: ht20; 5 GHz band: vht-40

Wi-Fi mesh link: Meshmerize1 - Meshmerize2

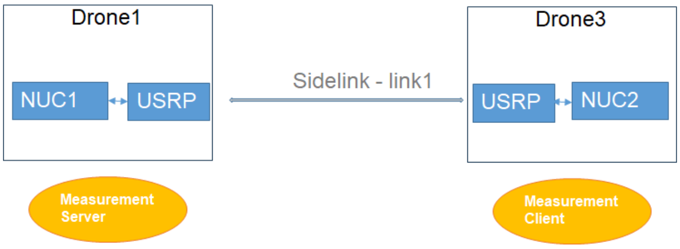


5G-Sidelink: NUC1 - USRP1 — USRP2 - NUC2

B. Measurements for the link 13 (5G-Sidelink)

Band n47	5.855-5.925 GHz
Downlink rate:	**1.75** Mbit/sec
Uplink rate:	**1.75** Mbit/sec
Round Trip Time:	**10** msec

The Mesh Wi-Fi technology is superior in terms of data transmission rates, while experimental 5G Sidelink delivers comparable performance in terms of Round-Trip-Time (RTT).



4.Connection to the terrestrial infrastructure 5G gNB Amarisoft

A. Separate connections. Drone 1 to 5G gNB / drone 2 to 5G gNB

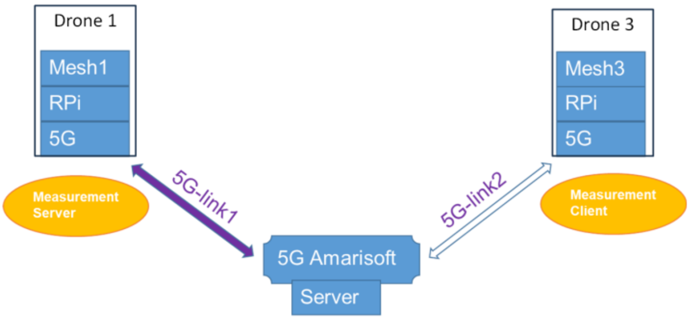
Measurements:	**5G-UE - link1**	**5G-UE - link2**
Downlink rate:	**255** Mbit/sec	**246** Mbit/sec
Uplink rate:	**62** Mbit/sec	**60** Mbit/sec
Round Trip Time:	**25** msec	**23** msec

B. The data is sent drone 1 / 2 to the server.

Both drones are connected to 5G gNB

Measurements:	5G-link1 <> 5G-link2
Downlink rate:	**40** Mbit/sec
Uplink rate:	**40** Mbit/sec
Round Trip Time:	**44** msec

The data is sent from drone 1 to drone 3 via the terrestrial 5G network (and vice versa).



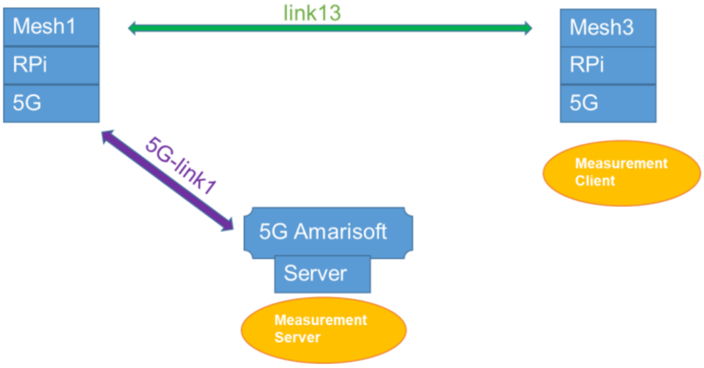
In a situation of one drone in the 5G network (link 1 or link 2 only), we measure prominent throughput in DL of 246 or 255 Mbit/s, which is more than twice as much as the throughput in the Wi-Fi network (channel 36, 5 GHz). However, when two drones are sending data to each other via ground 5G network we observe significant drop in DL performance to 40 Mbit/s, which is 6 times less than in the situation with one UE modem sends the data in the network. The 5G server "checks" whether the resource blocks are free, and only then sends the data. Afterwards, it waits for a response. This procedure results in higher latencies.

5.Combined terrestrial infrastructure 5G gNB and mesh network (one hop)

5G gNB — 5G UE-Mesh1 — Mesh3

2.4 GHz (channel 11)	5 GHz (channel 36)
Downlink rate: **38** Mbit /sec	Downlink rate: **44** Mbit/sec
Uplink rate: **36** Mbit / sec	Uplink rate: **54** Mbit/sec
Round Trip Time: **33** msec	Round Trip time: **33** msec

The data is sent from the server on the ground first to drone 1 and then via the mesh Wi-Fi network to drone 3 (and vice versa).



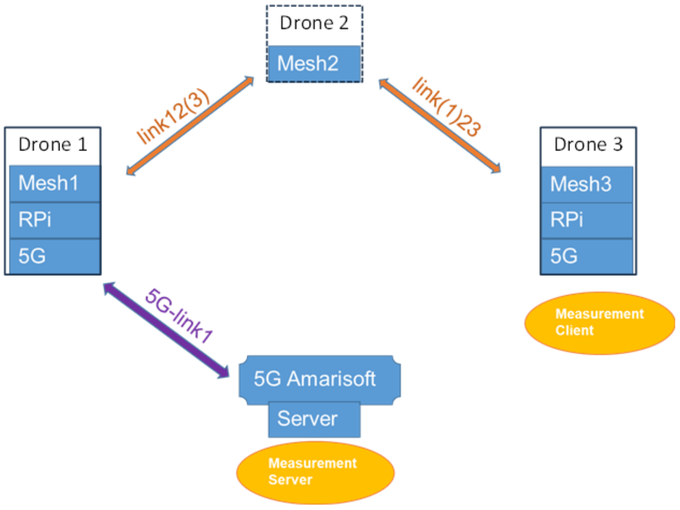
We observe an insignificant throughput drop (approximately 2 MBit/s in channel 11); however, in return, the drone mesh network experiences a significant extension of coverage. On 5 GHz frequencies, the 5G network provides more resources for data transmission than the mesh network in the current situation can offer on its side.

6.Combined terrestrial infrastructure 5G gNB and mesh network (two hops)

5GgNB — 5G UE-Mesh1 — Mesh2 —Mesh3

2.4 GHz (channel 11)	5 GHz (channel 36)
Downlink rate: **14** Mbit /sec	Downlink rate: **44** Mbit/sec
Uplink rate: **14** Mbit / sec	Uplink rate: **53** Mbit/sec
Round Trip Time: **38** msec	Round Trip time: **38** msec

The data is sent from the server on the ground first to drone 1 and then via the Wi-Fi mesh network to drone 2 and then to drone 3 (and vice versa).



We observe an insignificant throughput drop (approximately 2 MBit/s in channel 11); however, in return, the drone mesh network experiences a significant extension of coverage.

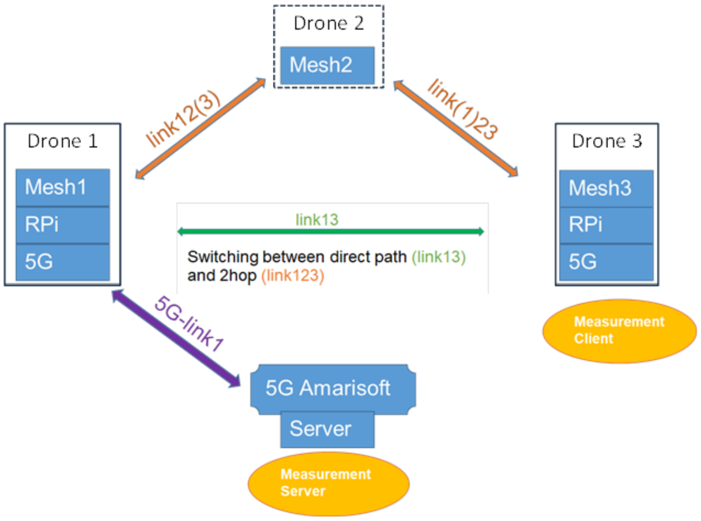
7.Combined terrestrial infrastructure 5G gNB and mesh network (two hops with switching option)

A.5G gNB — 5G UE-Mesh1— Mesh3 or — Mesh2 —Mesh3

2.4 GHz (channel 11) link13<>5G-link1	5 GHz (channel 36) link13<>5G-link1
Downlink rate: **34** Mbit /sec	Downlink rate: **41** Mbit/sec
Uplink rate: **28** Mbit / sec	Uplink rate: **51** Mbit/sec
Round Trip Time: **38** msec	Round Trip time: **33** msec

After the direct path (link13) is no longer available, the Wi-Fi mesh switches to the 2-hop link 12(3) and link (1)23 via the repeater drone (drone 2) to establish the data connection between drone 3 and the 5G server (5G core network) over all stations. The measured data rates and round-trip-times for the full link from the 5G network to drone3 (and vice versa) via drone1 and drone2 (repeater drone) are as follows:

2.4 GHz (channel 11) link123<>5G-link1	5 GHz (channel 36) link123<>5G-link1
Downlink rate: **14** Mbit /sec	Downlink rate: **41** Mbit/sec
Uplink rate: **12** Mbit / sec	Uplink rate: **51** Mbit/sec
Round Trip Time: **37** msec	Round Trip time: **38** msec



The measurement results are comparable to those measured for both cases 'one hop' and 'two hops' before without switching (cases 5 and 6). Here, we also observe a deterioration in data transmission on the 2.4 GHz band: Instead of 40 Mbit/s, which the 5G gNB provides in case 4B (two drones in the 5G network), we measure a 15% lower throughput in DL (34 Mbit/s) in the situation with two 5G UEs and a 65% lower throughput (14 Mbit/s) in the situation with two 5G UEs and an additional repeater drone. For the same scenarios on the 5 GHz frequency, a repeater drone shows no impact on the

	data rates in downlink, but the round-trip delay deteriorates by 15% (38 msec instead of 33 msec).

4.2.3 Discussion of the Results

In these tests, we aimed to evaluate multi-technology connectivity as preparation to the demonstration in the Task 5.3 within a reproducible laboratory environment. Our primary objectives were to ensure connectivity and interoperability between the different links, as well as to assess and measure the maximum achievable throughput in both uplink and downlink scenarios, along with associated latencies under conditions with 5G terrestrial infrastructure and out-of-coverage situations.

The devices were connected via cables, allowing for precise and static measurement of output. The tests were conducted over several minutes, and the results were collected using iPerf. Below are our observations and discussion of the results.

- In the roaming scenarios in Wi-Fi-mesh network, we observed significant drop in throughput from 40 Mbit/s to 15 Mbit/s in the frequency band 11 (2.4 GHz) upon switching from a direct link to a one-hop connection over a drone- This indicates that the network performance can be adversely affected by the added complexity of routing. This performance degradation, alongside an increase in latency of nearly 80%, emphasizes the challenges of maintaining high performance during roaming in mesh networks.
- When we switched from Wi-Fi-mesh network to the Terrestrial Infrastructure (5G gNB), the measurements revealed that a single drone connected to the 5G network can achieve impressive downlink throughputs of 246 to 255 Mbit/s, significantly surpassing the performance of Wi-Fi. However, when two drones communicate via the ground 5G network, the throughput drops significantly to 40 Mbit/s, highlighting the impact of switching to the technology with less possible throughput.
- In scenarios with repeater drones, the downlink throughputs are minimally affected at 5 GHz frequencies, but the round-trip-times do experience a notable increase. This suggests that while repeater drones can enhance coverage, they may introduce latency, which could be a consideration for time-sensitive applications.
- In scenario with range extension, we indicated that while Mesh Wi-Fi exceeds in data rates, the experimental 5G Sidelink implementation already supports comparable round-trip-times. This can be explained: commercial Wi-Fi offers better throughput with 2x2 MIMO, while we did not implement MIMO in 5G Sidelink yet. Despite, Wi-Fi has a higher bandwidth of 40 MHz in the 5 GHz frequency band, while a 5G Sidelink channel as specified for the ITS-frequency bands for road safety has a lower bandwidth of 10 MHz currently with an option to do carrier aggregation to 20 MHz total. 5G-Sidelink cannot be seen currently as a replacement of Wi-Fi-mesh technology, but has the potential in the 6G era to be extended for aerial applications. In

the D3.1 and D3.2, Fraunhofer IIS discusses the required functionality in the later releases of the 3GPP standard, which might be implemented for the multi-hop scenarios.

4.3 Experiment 2: NTN-link evaluation

4.3.1 Experimental Setup

The experiment focuses on an NTN link as an enabler of the communication where the terrestrial infrastructure fails and to enhance the reliability of interconnections among the drones where the terrestrial infrastructure is not dense enough to serve flying users.

Only one drone in the swarm is carrying the NTN UE or SAT-Modem responsible for the communication between itself and NTN Server (Amarisoft NTN gNB). In our experiment, with the help of the channel emulator, we establish the link between the NTN UE terminal and the NTN Server. An IP tunnel interface running on the control element on the drone (Raspberry Pi) directs the data flow to the NTN link.

A visualization of the network topology is provided in Figure 5.

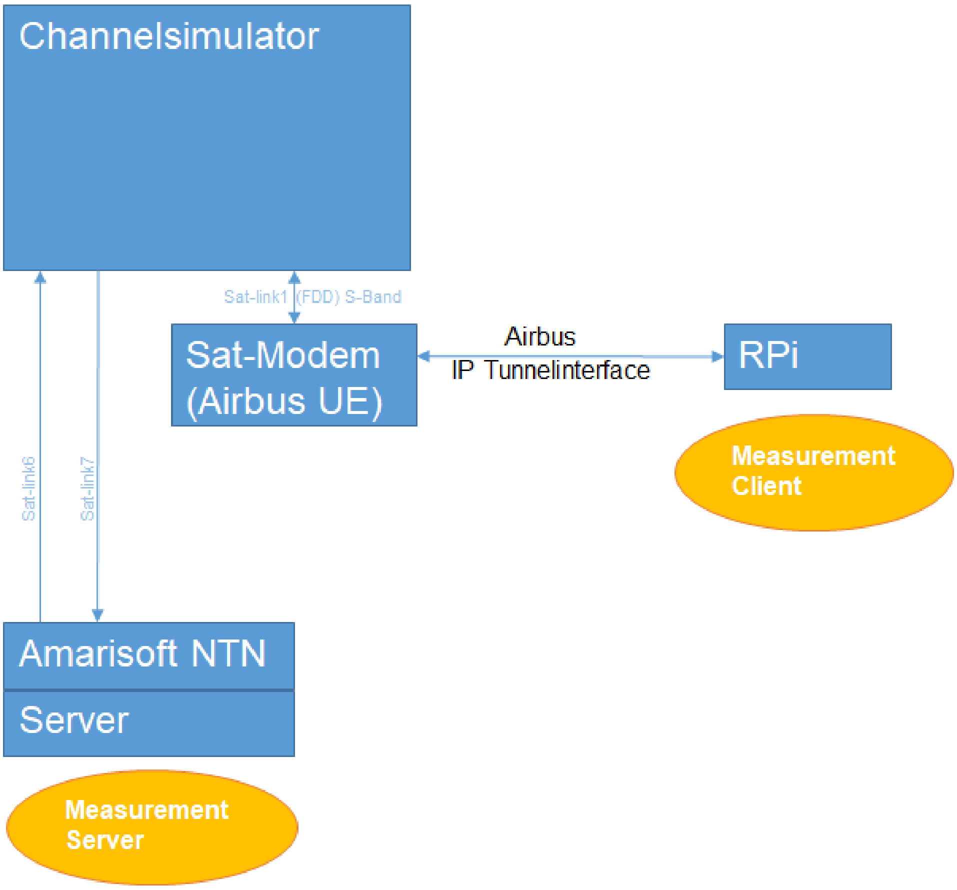


Figure 5. NTN-link evaluation setup

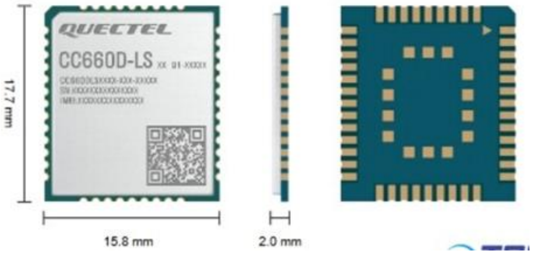
The following hardware and software are used in the test:

- Raspberry Pi 4
    - Airbus Tunnel interface to the NTN modem
- NTN 5G gNB (imitation of the base station on the satellite) (Amarisoft SDR Callbox mini)
- NTN UE single chip modem CC660D-LS
- Channel emulator

Table 1 contains parameters, configuration details and pictures of the hardware.

Table 1. List of hardware and configuration details for the NTN link evaluation experiment

Raspberry Pi 4 (Lakeside Labs) **with Tunnel IP interface to the NTN modem (Airbus)** (software implementation)	
NTN gNB – Callbox Mini configuration • Band 256 • UL: 1980–2010 MHz • DL: 2170–2200 MHz • dl_earfcn: 229300 • bandwidth: 200kHz • ul_max_harq_tx: 5, • dl_max_harq_tx: 5, • ul_max_consecutive_retx: 15, /* disconnect UE if reached */ • dl_max_consecutive_retx: 15, /* disconnect UE if reached */ • msg3_max_harq_tx: 1, • intra_freq_reselection: true, /* SIB1.intraFreqReselection-r13 */ • q_rx_lev_min: -70, /* SIB1.q-RxLevMin */ • q_qual_min: -34, /* SIB1.q-QualMin */ • si_window_length: 160, /* ms */ • si_radio_frame_offset: 0, /* in frames */ • si_periodicity: 128, • si_repetition_pattern: 2, • ul_sync_validity: 240, • GEO-stationary simulation mode • ground_position: latitude: 48.853, longitude: 2.348, altitude: 140	

NTN UE single chip modem CC660D-LS • 3GPP NTN Rel-17 L- (B255) S- (B256/23) • **IoT-NTN:** L- Band (B255): UL: 1626.5–1660.5 MHz; DL: 1525–1559 MHz • S-Band (B256): UL: 1980–2010 MHz; DL: 2170–2200 MHz • S-Band (B23): UL: 2000–2020 MHz; DL: 2180–2200 MHz **Bandwidth:** • UL single-tone and multi-tone • DL 200 kHz bandwidth • Data Transmission • Single-tone with 15/ 3.75 kHz subcarrier: UL • 5.8 kbit/s (15 kHz)/ 2.7kHz • Small size 17.7 mm × 15.8 mm × 2.0 mm • Supply Voltage Range: 2.2–3.6 V • Power Consumption: 3.4 µA @ Deep Sleep 278 mA @ TX, 23 dBm (B256)	
Channel emulator Keysight Propsim F64 (Fraunhofer IIS) • Frequency 450MHz-6GHz • Delay maximum 1.6sec (**Aerospace Option)** • Insertion delay 2.5 µs • Dopplershift maximum +/- 750kHz • Bandwidth maximum 450/600MHz / 1 to 4 MIMO • SISO, 2x2, 4x2, 2x4, 4x4, 4x8, 8x4 • Up to 1536 internal MIMO channels • Fast fading profiles/channel models • Up to 48 multipath • Interference generation AWGN, CW Integrated uplink and downlink separation	

4.3.2 Evaluation results

Our experiment centers on a Non-Terrestrial Network (NTN) link as a facilitator of communication when terrestrial infrastructure is unavailable. To improve the reliability of interconnections among drones, only one drone in the swarm is equipped with the NTN User Equipment (UE).

The registration process of the 5G NTN UE (SIM card of the CC660D NTN modem) in the network took 15 seconds for the initial setup and functioned reliably throughout the duration of the experiment. After the registration was complete, an IP address was allocated to the CC660D NTN modem by the network. The 5G NTN UE could access the NTN link via the tunnel interface (uplink) and could also be reached from the satellite network (downlink). Measurement results indicated a bit rate of 960 bits/sec for both uplink and downlink between the UE (iperf3 client) and the core network (iperf3 server) (see figure 4-3).

The experiment emulates data transmission via a geostationary satellite, which is approximately 36,000 km away from Earth, within the 5G Narrowband IoT framework that is suitable for transmitting small amounts of data.

4.3.3 Discussion of the Results

In the experiment, we utilized a single drone equipped with the NTN UE, responsible for communication with the NTN network (Amarisoft NTN gNB). With the assistance of a channel emulator, we successfully established the connection between the NTN UE terminal and the NTN Server. The link operates via a geostationary satellite, located about 36,000 km from Earth, with the IoT-NTN communication. The IP tunnel interface running on the control element of the drone (Raspberry Pi PC) directs the data flow to the NTN link.

Compared to the access time to terrestrial networks, which typically takes only a few seconds, the connection to the NTN Server, due to the distance to the satellite and increased delay, took 15 seconds in our experiment. The NTN link was stable and reliable. The measured bit rate of 960 bits/sec provided by the available NB-IoT modem is only suitable for transmitting very small amounts of data, such as GPS coordinates or drone telemetry and mission coordinates.

When utilizing geostationary satellites, a significant delay of at least 240 ms must be considered. This latency is attributable to the distance to the satellite and cannot be minimized. In contrast, if Low Earth Orbit (LEO) satellites were positioned at an altitude of 400 km from Earth, the round-trip latency could be reduced to 40 ms. Therefore, in the current scenario, the 3GPP NTN NB-IoT link can be regarded as a very low data rate backup system in comparison to 5G-TN (we measured 250 Mbit/s in downlink) and meshed Wi-Fi technologies (we measured 40 Mbit/s between two drones on 2,4 GHz frequency). A medium data rate can be achieved with 5G NR NTN communication links, which are broader in bandwidth (>= 5 MHz instead of 180 kHz). These systems are currently considered to be deployed by multiple satellite operators, who are active in 3GPP standardization.

4.4 Experiment of Combined TN and NTN Network with HAP

Lead: Fraunhofer IIS, **Contributors**: Deutsche Telekom, Airbus

4.4.1 Scenario Description

This scenario tested in the scope of the Task T5.2 Emulation is a combined terrestrial and non-terrestrial network based on Task T5.6. Here, a plane acting as a so-called high-altitude-platform (HAP) is providing a 5G non-terrestrial network (NTN) for User Equipment (UE) within an area on the ground. The connection to the 5G core network is provided by a backhaul connection via satellite. Besides the NTN, there is also a terrestrial network (TN) provided by base-stations (gNBs) on ground. The UE is served either by the terrestrial or the non-terrestrial network, depending on the connection conditions. This scenario is shown in Figure 6.

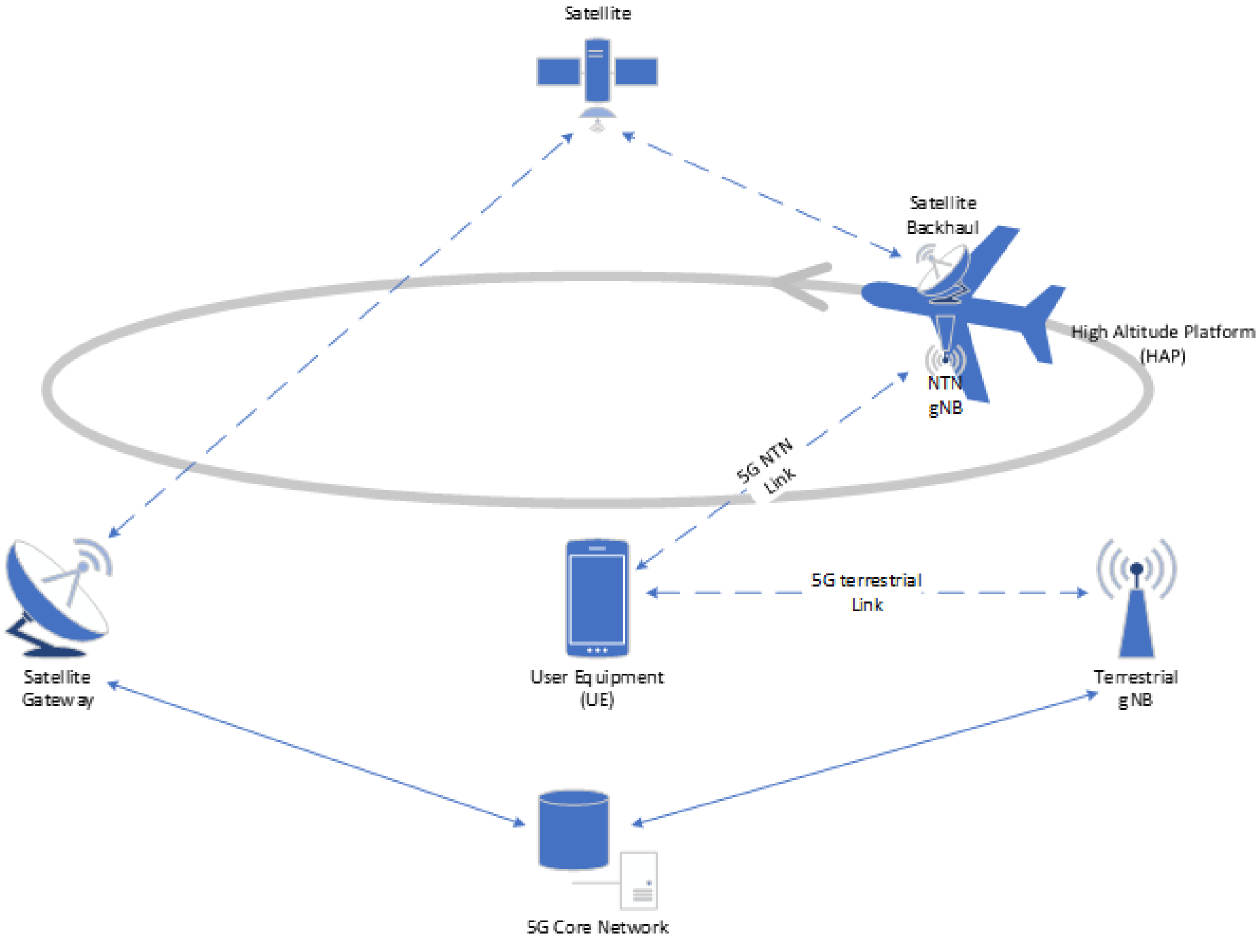


Figure 6. Combined TN and NTN Network with HAP emulation scenario

4.4.2 Hardware Setup

The hardware setup to re-create the described scenario is shown in Figure 7. The components shown in the Scenario Description are present here too, i.e.

- Standard 5G UEs /Modems
- NTN 5G gNB on HAP
- Terrestrial 5G gNB on ground
- Terrestrial 5G Core Network
- Satellite Backhaul equipment (HAP)
- Satellite Backhaul equipment (Ground)

Besides these, we use the Keysight PROPSIM F64 channel emulator to emulate the different radio links:

- Bidirectional Satellite Backhaul link (HAP <-> Ground)
- Bidirectional terrestrial 5G NR link (UE <-> Terrestrial 5G gNB)
- Bidirectional non-terrestrial 5G NR link (UE <-> NTN gNB on HAP)

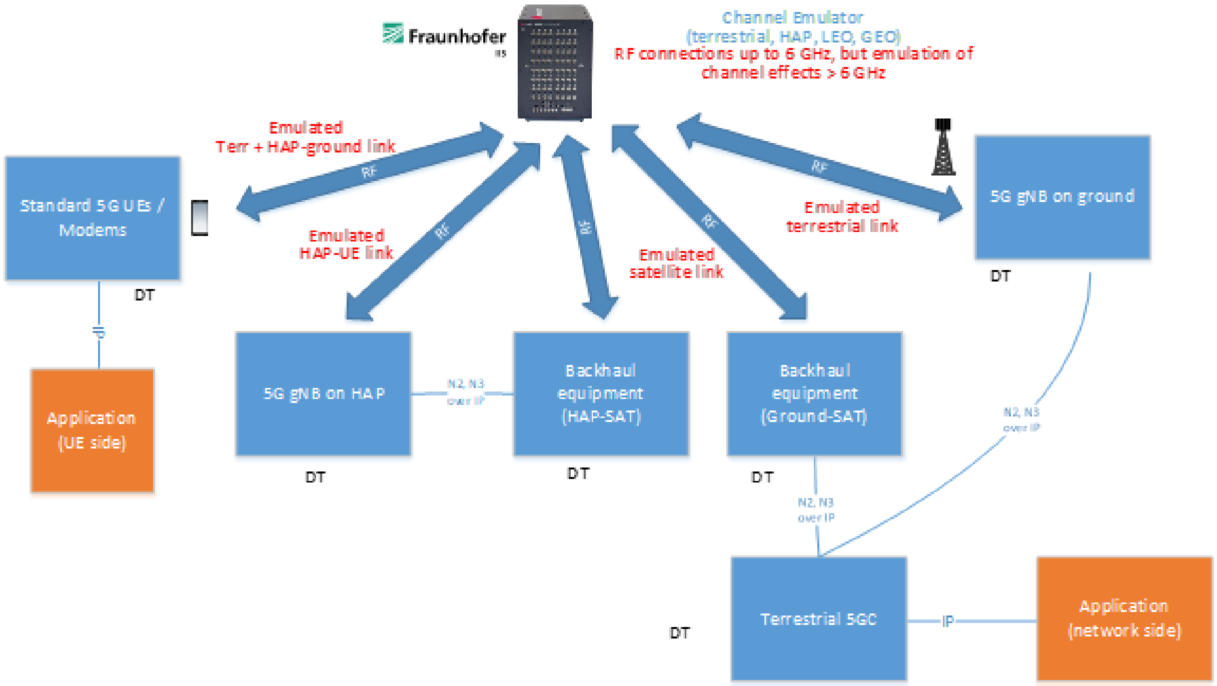


Figure 7. Hardware Setup: 6G-Architecture: HAP with backhaul over satellite, HAP-TN handover

4.4.3 Key Parameters and Configuration

In this section, we give an overview of the key parameters for the different emulated radio links: the 5G NR terrestrial and non-terrestrial network parameters, and the satellite backhaul parameters.

5G NR TN / NTN Parameters

- Band n1
- Full Duplex FDD
- 2x2 MIMO
- Downlink Bandwidth: 10 MHz
- Uplink Bandwidth: 10 MHz
- Downlink resource blocks: 52
- Uplink resource blocks: 52
- Subcarrier spacing: 15 kHz
- Slot duration: 1 ms
- Slots per frame: 10 slots per frame
- DL Frequency: 2160.150 MHz
- UL Frequency: 1970.150 MHz

Channel model used for TN:
According to TS 38.901:TDL-A, Delay spread 30 ns, Doppler 5 Hz, 2x2 MIMO

Channel model used for NTN:
Similar to TS 38.811: TDL-C, Line-of-sight, Speed 5 km/h
Notes:

- We assume a HAP flying in a height of 14 km.
- The HAP is assumed to fly in circles with a diameter of 10 km.
- The 5G NR cell covered by the HAP is assumed to have a diameter of 50 km.
- Therefore, we assume a propagation delay of 140 µs between HAP and UE.
- According to TS 38.901 we use a Delay spread scaling factor of 37 for a rural normal delay profile.
- As the UE performance was unexpectedly bad with Ricean Tap for LOS, this was removed and the attenuation of the Rayleigh Tap adjusted instead.

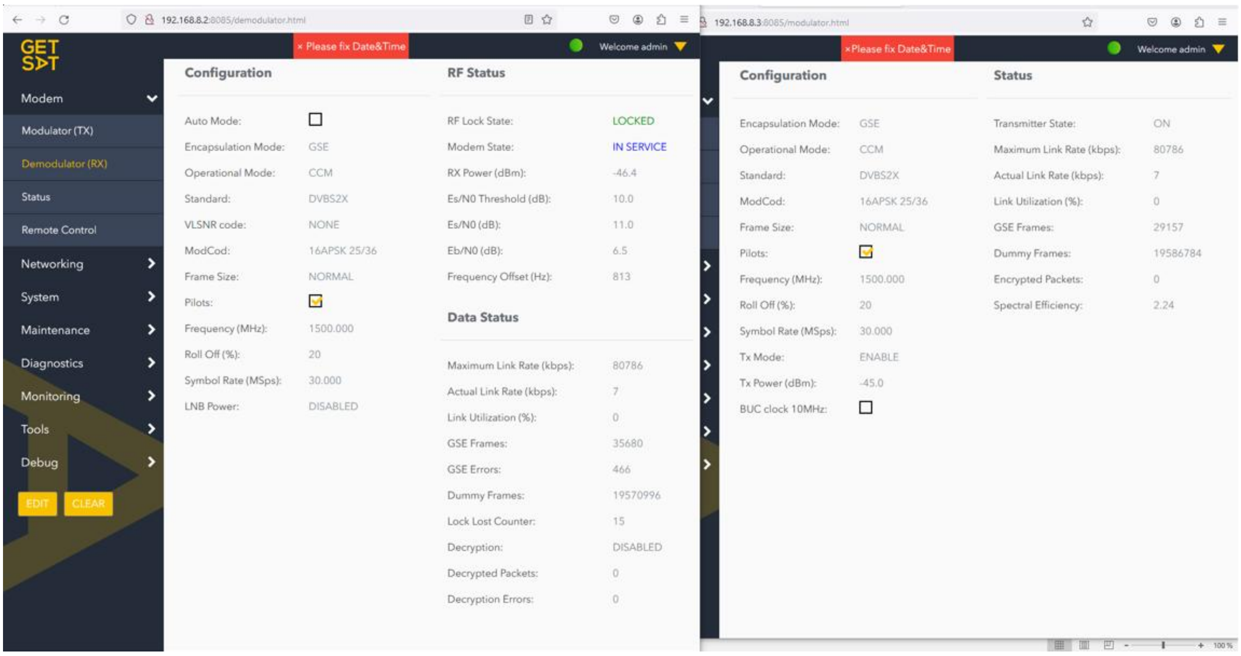
The following tap-delay-line (TDL) is used:

Tap No.	Delay	Gain (dB)	Fading
1	140 000 ns	0	Rayleigh
2	140 548 ns	-23.4	Rayleigh

Satellite Backhaul parameters

- Waveform: DVB-S2X
- Full Duplex FDD
- Downlink Bandwidth: 36 MHz
- Downlink Es/N0: 11 dB
- Downlink ModCod: 16 APSK 25/36
- Maximum Downlink Link Rate: 80786 kbit/s
- Downlink IF Frequency: 1500 MHz
- Uplink Bandwidth: 36 MHz
- Uplink Es/N0: 6 dB
- Uplink ModCod: 8 BPSK 5/9
- Maximum Uplink Link Rate: 48379 kbit/s
- Uplink IF Frequency: 1700 MHz
- One-way Satellite propagation delay Gateway à Satellite à UE: 250 ms
- One-way Satellite propagation delay UE à Satellite à Gateway: 250 ms

The status of the satellite modem downlink and uplink status is shown in Figure 8 and Figure 9.



Configuration

Auto Mode:	☐
Encapsulation Mode:	GSE
Operational Mode:	CCM
Standard:	DVBS2X
VLSNR code:	NONE
ModCod:	16APSK 25/36
Frame Size:	NORMAL
Pilots:	☑
Frequency (MHz):	1500.000
Roll Off (%):	20
Symbol Rate (MSps):	30.000
LNB Power:	DISABLED

RF Status

RF Lock State:	LOCKED
Modem State:	IN SERVICE
RX Power (dBm):	-46.4
Es/N0 Threshold (dB):	10.0
Es/N0 (dB):	11.0
Eb/N0 (dB):	6.5
Frequency Offset (Hz):	813

Data Status

Maximum Link Rate (kbps):	80786
Actual Link Rate (kbps):	7
Link Utilization (%):	0
GSE Frames:	35680
GSE Errors:	466
Dummy Frames:	19570996
Lock Lost Counter:	15
Decryption:	DISABLED
Decrypted Packets:	0
Decryption Errors:	0

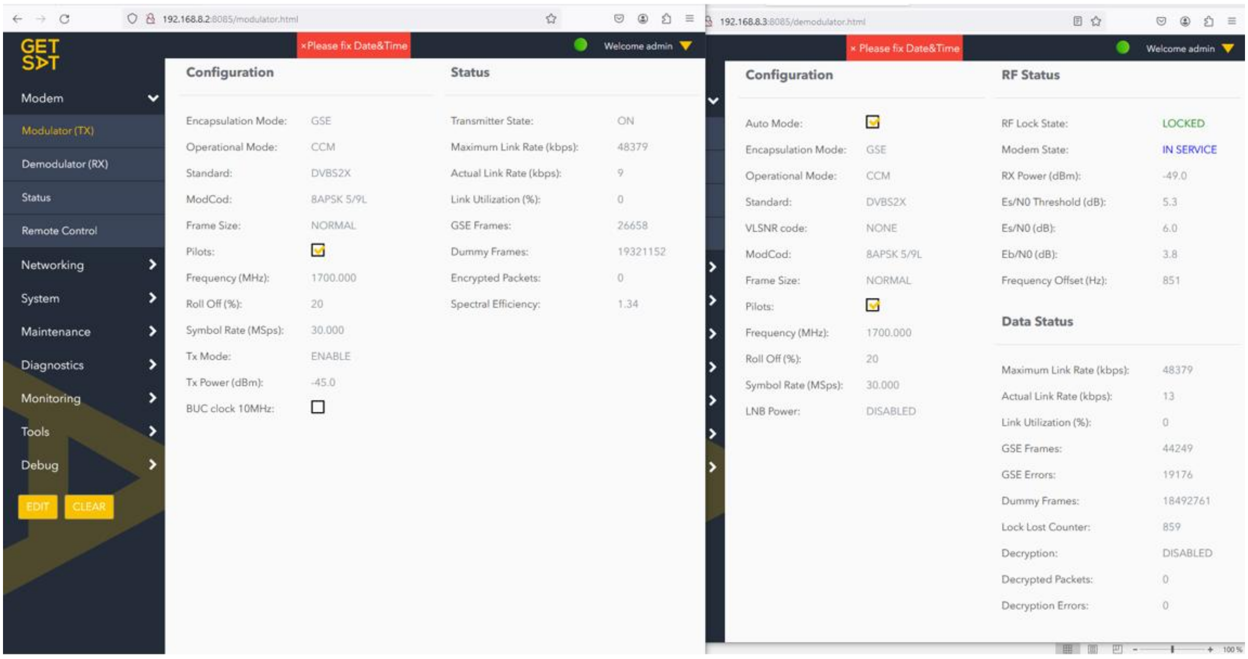
Configuration

Encapsulation Mode:	GSE
Operational Mode:	CCM
Standard:	DVBS2X
ModCod:	16APSK 25/36
Frame Size:	NORMAL
Pilots:	☑
Frequency (MHz):	1500.000
Roll Off (%):	20
Symbol Rate (MSps):	30.000
Tx Mode:	ENABLE
Tx Power (dBm):	-45.0
BUC clock 10MHz:	☐

Status

Transmitter State:	ON
Maximum Link Rate (kbps):	80786
Actual Link Rate (kbps):	7
Link Utilization (%):	0
GSE Frames:	29157
Dummy Frames:	19586784
Encrypted Packets:	0
Spectral Efficiency:	2.24

Figure 8. Satellite Modem Downlink Settings / Status



Configuration

Encapsulation Mode:	GSE
Operational Mode:	CCM
Standard:	DVBS2X
ModCod:	8APSK 5/9L
Frame Size:	NORMAL
Pilots:	☑
Frequency (MHz):	1700.000
Roll Off (%):	20
Symbol Rate (MSps):	30.000
Tx Mode:	ENABLE
Tx Power (dBm):	-45.0
BUC clock 10MHz:	☐

Status

Transmitter State:	ON
Maximum Link Rate (kbps):	48379
Actual Link Rate (kbps):	9
Link Utilization (%):	0
GSE Frames:	26658
Dummy Frames:	19321152
Encrypted Packets:	0
Spectral Efficiency:	1.34

Configuration

Auto Mode:	☑
Encapsulation Mode:	GSE
Operational Mode:	CCM
Standard:	DVBS2X
VLSNR code:	NONE
ModCod:	8APSK 5/9L
Frame Size:	NORMAL
Pilots:	☑
Frequency (MHz):	1700.000
Roll Off (%):	20
Symbol Rate (MSps):	30.000
LNB Power:	DISABLED

RF Status

RF Lock State:	LOCKED
Modem State:	IN SERVICE
RX Power (dBm):	-49.0
Es/N0 Threshold (dB):	5.3
Es/N0 (dB):	6.0
Eb/N0 (dB):	3.8
Frequency Offset (Hz):	851

Data Status

Maximum Link Rate (kbps):	48379
Actual Link Rate (kbps):	13
Link Utilization (%):	0
GSE Frames:	44249
GSE Errors:	19176
Dummy Frames:	18492761
Lock Lost Counter:	859
Decryption:	DISABLED
Decrypted Packets:	0
Decryption Errors:	0

Figure 9. Satellite Modem Uplink Settings / Status

4.4.4 Shadowing profiles

While performing the Latency (ping) and Throughput tests reported in the following sections, we apply a shadowing profile (see Figure 10) that starts with 0 dB initial attenuation and goes down to 80 dB attenuation after 120 seconds. This is done to show the influence of the channel attenuation on the Latency and Throughput.

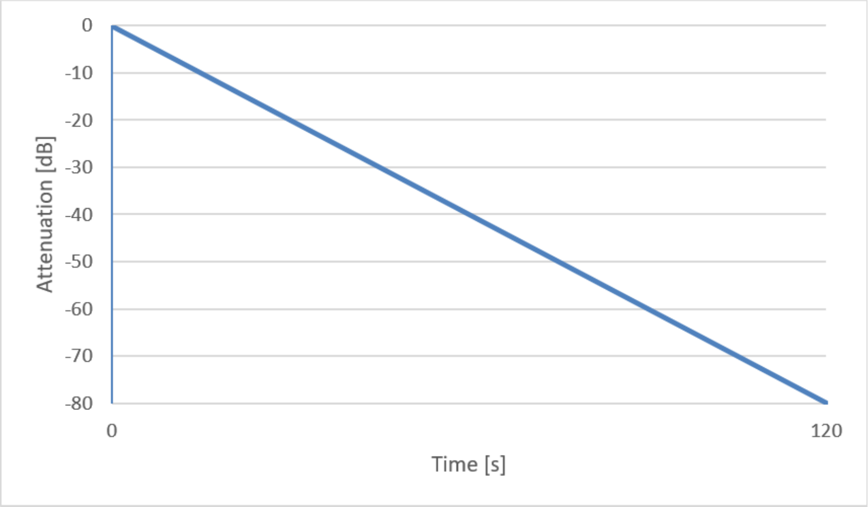


Figure 10. Shadowing profile to show effects of attenuation on Throughput and Latency

To enforce handovers during the combined TN / NTN tests, we use a periodic shadowing profile that goes from 0 dB attenuation to 60 dB attenuation for one path, while the other path goes from 60 dB attenuation to 0 dB attenuation within 30 seconds, and vice-versa in the next 30 seconds (see Figure 11). So, the optimal time for handover is at an attenuation of 30 dB every 30 seconds.

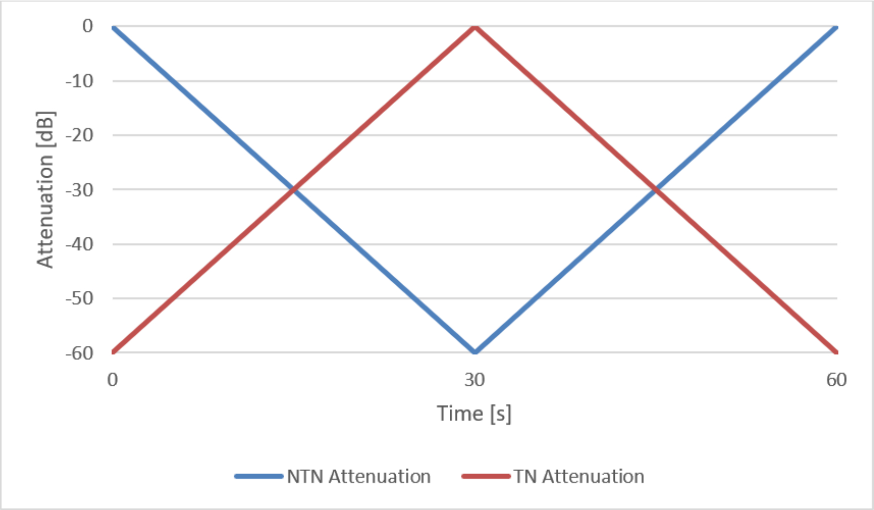


Figure 11. Shadowing profile to enforce handovers

4.4.5 Test Results for Combined TN / NTN Scenario

This section presents the test results for the combined scenario including both, terrestrial network (TN) and non-terrestrial network (NTN). Two tests are performed, for both the shadowing profiles described in Sec 4.4.4, enforcing handover between TN and NTN every 30 seconds, is used.

Voice call with mobility / repeated handover between TN and NTN

In the first test, a Voice over NR (VoNR) call between two UEs is established with a session duration of 300 seconds. During this VoNR session, we measure how many handovers are successful and how many fail. The results are given in Table 2.

Table 2. Overall 5G-NR SA handover statistics for VoNR tests

Overall Intra-NR Handover Summary		
Statistics	**Value**	
No. Of HO Attempt	11	
No. of HO Success	11	
No. of HO Failure	0	
Avg. HO Duration (sec)	0,019	
Overall Intra-NR Handover Success & Failure Rate		
	Success (%)	**Failure (%)**
Overall Results	100,00%	0,00%

As can be seen from test results in Table 2, during the test 11 handover attempts have been recorded and all these handovers were successful.

Ping / Latency Test with Mobility / repeated handover between TN and NTN

In the second test, instead of a voice call between two UEs, a ping command is used to measure the round-trip-latency between the UE and the core network. The ping command uses "wait for response" i.e. it waits for response before subsequent ping is sent (5000ms timeout). The test results are given in Table 3.

Table 3. Overall 5G-NR SA handover statistics for ping/latency tests

Overall Intra-NR Handover Summary	
Statistics	**Value**
No. Of HO Attempt	6
No. of HO Success	6

No. of HO Failure	0	
Avg. HO Duration (sec)	0,019	
Overall Intra-NR Handover Success & Failure Rate		
	Success (%)	**Failure (%)**
Overall Results	100,00%	0,00%

As can be seen from test results in Table 3, during the test 6 handover attempts have been recorded and all these handovers were successful.

Figure 12 shows the Synchronization Sequency Reference Signal Received Power (SS-RSRP) in dBm * 100. For instance, a value of -9000 corresponds to -90.00 dBm. The effects of the applied shadowing profile can be seen, as well as the 6 handover occasions at the minimum values (approx. -115 dBm).

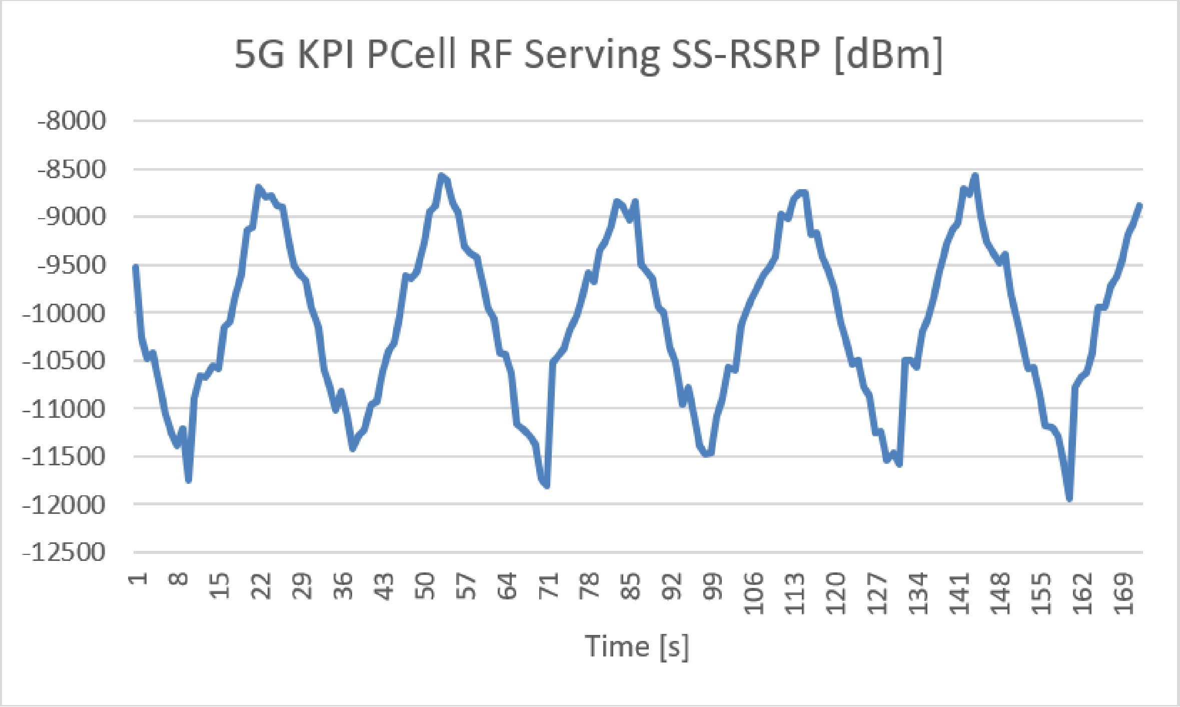


Figure 12. Synchronization Sequency Reference Signal Received Power

The next graph in Figure 13 shows the Synchronization Sequency Signal to Interference and Noise Ratio (SS-SINR) in dB * 100. For instance, a value of 3000 corresponds to 30.00 dB. Just like in the graph above, the effects of the applied shadowing profile can be seen, as well as the six handover occasions at the minimum values (approx. 0 dB SNR).

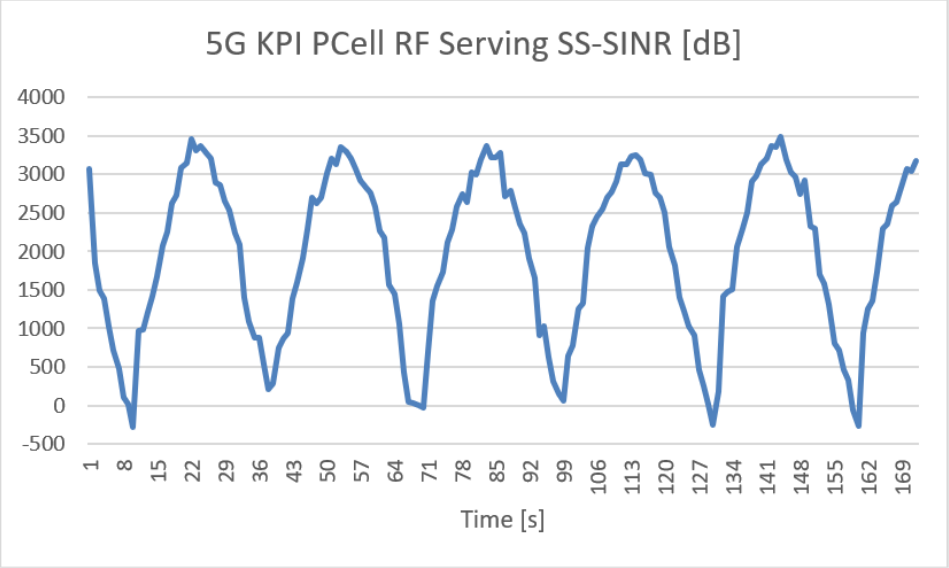


Figure 13. Synchronization Sequency Signal to Interference and Noise Ratio

The last graph in Figure 14 shows the round-trip-time results. It can be clearly seen when the UE is connected to the terrestrial network and when it is connected to the NTN, based on the different RTT values. The terrestrial RTT is almost always below the 200 ms line, while the NTN RTT consistently remains above 500 ms. Interestingly, during handover events, the RTT increases significantly (RTT values of 1000 ms and more) even though the handovers themselves were successful, as shown in Table 3.

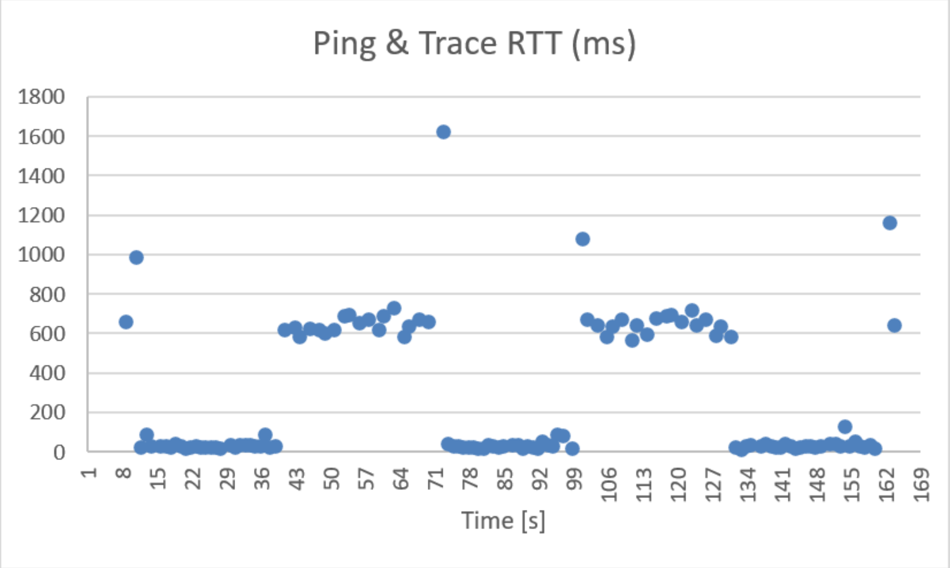


Figure 14. Measured Round-trip-time

Overall, these hybrid TN / NTN tests showed good and consistent results without any obvious issues.

4.4.6 Discussion of the Test Results

Voice calls in the hybrid TN / NTN network could be initiated, received, and established without any issue, demonstrating consistent and reliable performance. The NTN round-trip-time was in the expected range (above 500 ms) because of the large propagation delay alone due to the distance to the geostationary satellite.

5 Task 5.3: Multi-technology network integration

5.1 Overview

Lead: Ericsson Germany (until mid-2024), Fraunhofer IIS (from mid-2024 until end of the project), **Contributors:** Meshmerize, Airbus, Motius, Lakeside Labs, Deutsche Telekom, KTH, Ericsson Sweden, Skysense

5.1.1 Motivation

The scope of Task 5.3 was originally outlined in D5.1, aiming to integrate the components developed in WPs 1-4 into a functional demonstration of wireless communication links over-the-air. This demonstration was to be tested with dedicated hardware under various real-world environmental conditions, including drones, antenna modules for ground stations, antennas for the flying vehicle, and dedicated radios or Software-Defined Radios (SDRs) to meet specific requirements. Our intention was to evaluate the technologies we investigated, particularly focusing on the new 7 GHz to 15 GHz frequency band for which the new RF front-end component was essential. However, we faced challenges in utilizing the new 3GPP-compliant 6G frequencies. The antenna system DA2G prototype from EAG (7GHz) did not reach the necessary maturity for testing by the scheduled time and is shown separately. Furthermore, the GEO satellite Heinrich Hertz (H2SAT) (Ka-band) was not available for our planned demonstrations with drones in April 2025, as reported by the DLR.

In response to these setbacks, the demonstration partners found a fallback solution. A new demonstration setup is proposed as a performance baseline for future combined NTN/TN 6G systems. We are demonstrating a multi-connectivity in 3D network that provide simultaneous end-to-end connectivity for the flying users (drones). Additionally, we are showcasing the best possible data rates, latency, and other performance metrics achievable by our setup.

5.1.2 Setup

The field-demonstration took place on the site of Airbus in Ottobrunn/Munich. There is a helipad with sufficient free space for drone flights. We set up our special ground base stations there and test our setup with several drone platforms that are equipped with dedicated communications and antenna modules flying around the terrain. Figure 15 shows the demonstration area and Figure 16 and Figure 17 impressions from the demonstration preparations.

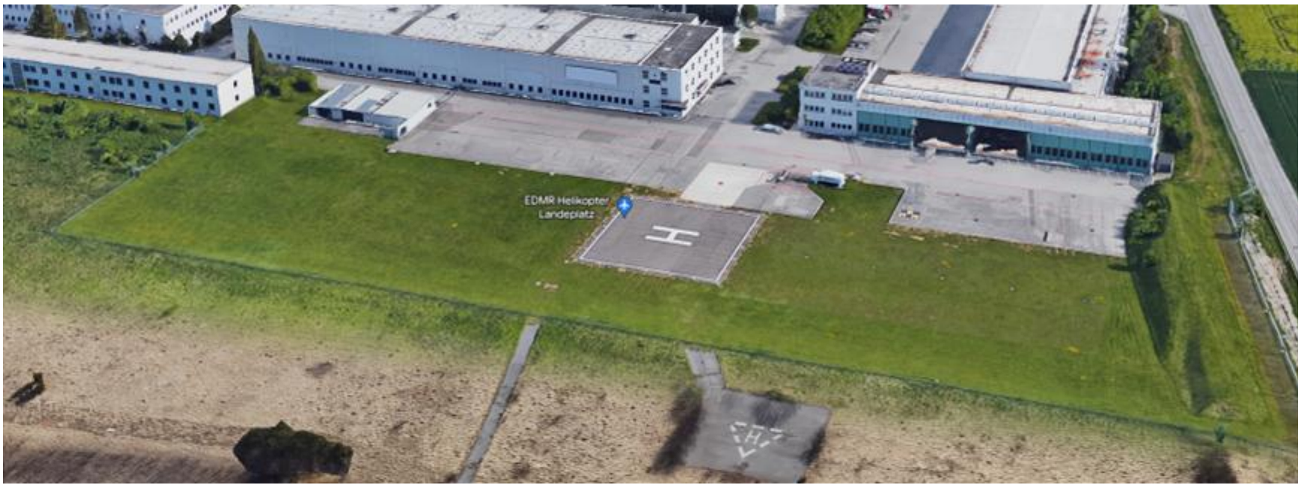


Figure 15. Airbus Ottobrunn Helipad, Munich, Germany

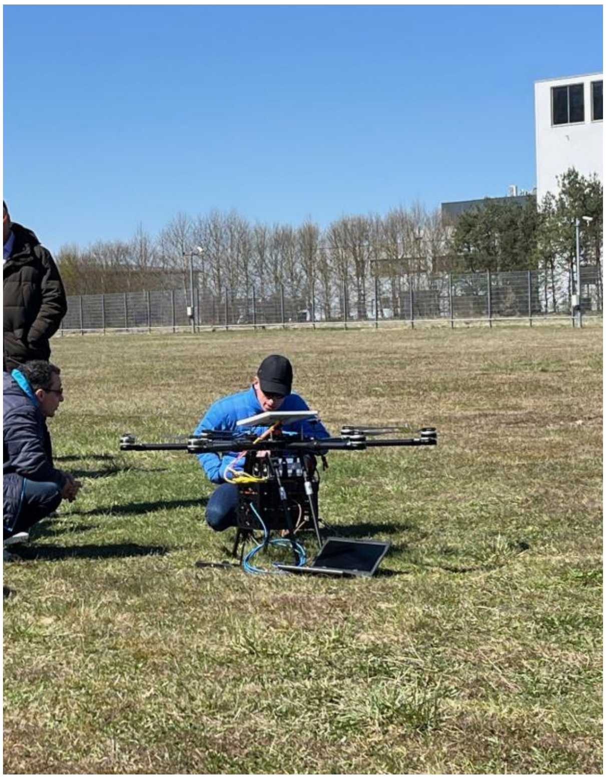


Figure 16. Preparation of the drone with multi-connectivity solutions (Wi-Fi-mesh, 5G and NTN terminals)

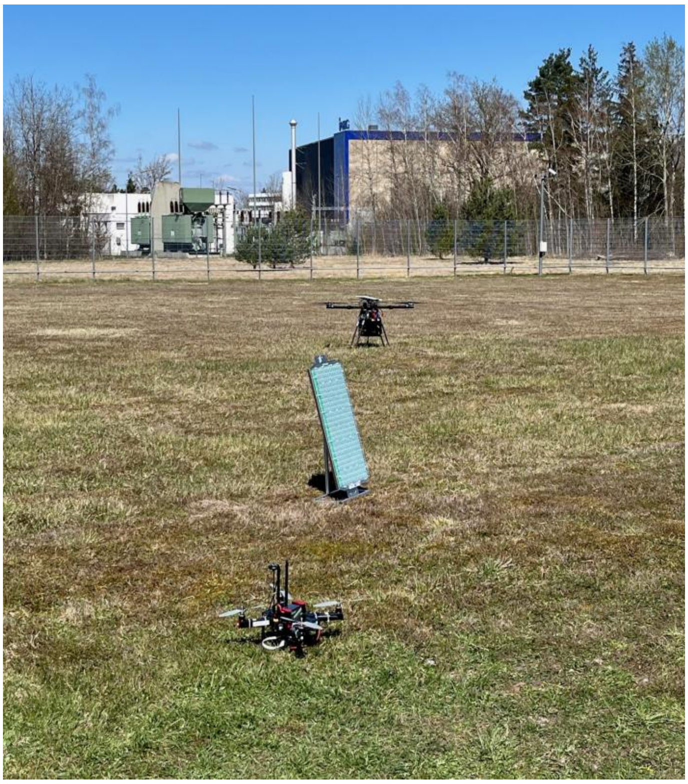


Figure 17. EAG antenna for the 6G-candidate 7GHz and the drones for the demonstrations

In absence of the H2SAT demonstration, partners were searching for the alternative European satellite providers. The LEO satellites providers which were considered, were ruled out after a thorough analysis for various reasons, such as the unavailability of testing licenses from BNetzA in Germany and the limited constellation sizes that could not adequately support real-time data transfer, critical for the demonstration. Consequently, the partners resorted to commercially available solutions from LEO satellite provider, with a terminal provided by KTH.

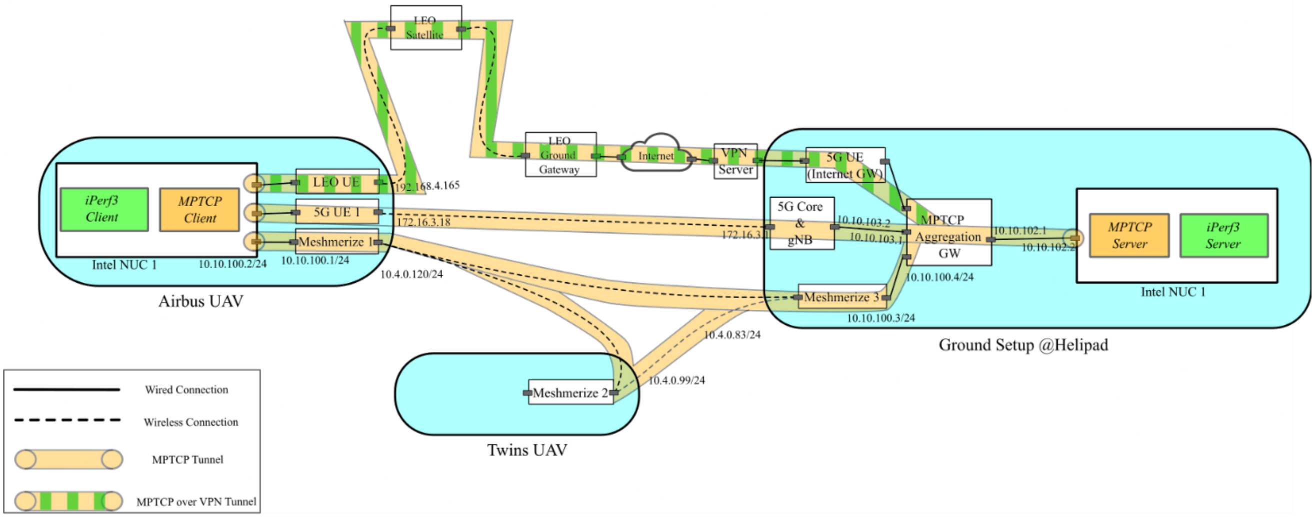


Figure 18. Seamless multi-technology link handover demo setup

Figure 18 shows the end-to-end setup of the flight demo at Airbus site in Ottobrunn, Germany. The drone (“Airbus UAV” in Figure 18) was equipped with three user equipment (UE): 1. a Meshmerize UE for Direct Air-to-Air (DA2A) connectivity; 2. a 5G UE for terrestrial connectivity, and 3. a LEO satellite UE for space-based connectivity. Hence, the drone had the capability of connecting to the ground via three different wireless links. On the ground, an aggregation gateway was set up to collect all the data traffic over three links. The 5G link was established with a private 5G network, using a femto cell base station on the ground at 3.7 GHz. The femto cell indicates that the 5G coverage is limited to approximately <100 m, due to the limited transmit power of the base station. The 5G core network was connected to the aggregation gateway. The Meshmerize UEs use a proprietary mesh protocol on top of Wi-Fi technology (details about the mesh hardware is given in Section 3.2) to establish the DA2A connectivity and the 2nd Meshmerize UE was set up at the gateway for data reception from the Airbus UAV. The 3rd Meshmerize UE on the Twins UAV was also available to relay the data to the ground if the Airbus UAV flies outside of the coverage zone of the Meshmerize device on the ground. Lastly, another 5G UE was connected to the gateway to receive the data traffic via the LEO satellite link. This 5G UE had a SIM card from one of the public mobile network operators and had an access to the public Internet. As the data traffic is sent to the public Internet via the LEO satellite link, we used this 5G UE as an Internet access point for the aggregation gateway at the field. We set up a virtual private network (VPN) over the LEO satellite link to be able to route the data traffic via Internet back to the aggregation gateway at the field.

The goal of this flight measurement is to evaluate how this multilink configuration can help provide seamless and reliable drone connectivity for safety-critical applications such as remote piloting and monitoring. To achieve seamless connectivity, the underlying links have to complement one another if one of the links become unavailable due to various reasons, such as out-of-coverage zones or network congestion. The joint use of the 5G, Meshmerize and LEO satellite links is very suitable for drones as they have different link characteristics. Therefore, we aim to manage these links in an efficient way to achieve seamless connectivity during the drone flight. While the links can be managed on different layers of the network stack, we decided to use the multipath transport layer (layer 4). This layer is responsible for efficiently managing the network resources of the underlying links and we employed the multipath transmission control protocol (MPTCP) in the transport layer.

MPTCP receives data packets from the data traffic application and it allocates a different portion of the data traffic over the underlying wireless links to provide loss-less and in-order packet delivery to the receiver. The receiver re-assembles the data packets as received from different links. MPTCP senses the links' capabilities in terms of capacity, latency and packet drops to determine the amount of data traffic to allocate to a certain link. Packet scheduling and congestion control algorithms are located at the core of MPTCP to achieve this goal. Congestion control algorithms are responsible for detecting link congestions so that MPTCP can avoid network bottlenecks and use the other links when a link congestion is detected. The congestion window (CWND) size determines the allowed number of packets that can be sent without receiving acknowledgement from the receiver. Hence, CWND size influences the amount of data that can be sent via an underlying link. We used the Reno congestion control algorithm in our setup. Reno halves the CWND size of a link if it detects a packet loss. Then, it gradually increases the CWND again if it does not detect packet losses anymore. As for the scheduler, we employed the shortest round-trip-time (sRTT) algorithm, which aims to utilize the links that have the lowest round-trip latency.

Regarding the data traffic, we generated a constant bit rate traffic at 10 Mbit/s using the iPerf3 tool. The traffic was generated at the Airbus UAV and sent to the ground, representing the video stream for remote piloting operations. We selected 10 Mbit/s since it allows for high-resolution video streaming (e.g., High-Definition or even more) and we wanted to observe whether this rate can be consistently maintained during the flight. Video streaming for remote piloting operations is one of the most capacity demanding applications for drone connectivity.

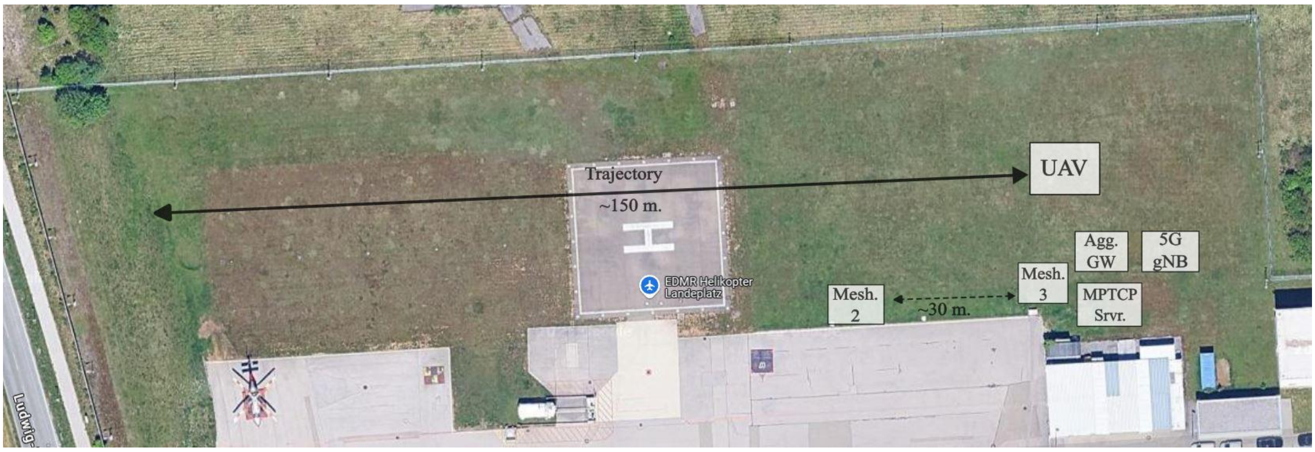


Figure 19. Flight Demo Setup at Airbus Ottobrunn Helipad

Figure 19 shows the bird view of the flight demo setup at the Airbus helipad in Ottobrunn, Germany. The Airbus UAV was flown horizontally at altitudes up to 30 m to fly away from the 5G base station and the Meshmerize UEs. The goal was to fly out of the coverage zones of the 5G and Meshmerize to see how dynamically MPTCP can adapt to the changing link conditions and whether at least one of the links can be utilized when other links are out of coverage. Therefore, the Airbus UAV flew up to 150 m away horizontally from the 5G base station as well as the Meshmerize UEs. During the flight, packet captures, 5G base station statistics (SNR, modulation and coding), and Airbus UAV logs were collected for evaluation, which we elaborate further in the next section.

Table 4. Individual Link Performance during static measurements on the ground at Airbus Ottobrunn Helipad

	Meshmerize	**5G**	**LEO**
Data Rate	>30 Mbit/s	5 Mbit/s	~5 Mbit/s
Latency	~5 ms	~30 ms	150-200 ms (ca. 20 ms only LEO)
Initial Connection Time	A few seconds	A few seconds	A few minutes
Reconnection Time	A few seconds	A few seconds	A few minutes

Table 4 shows the individual link performances in ideal conditions in a lab environment. The Meshmerize link with a channel bandwidth of 40 MHz can provide more than 30 Mbit/s with link latencies below 10 ms. As it has the lowest link latency and abundant link capacity compared to the other links, MPTCP prioritizes the Meshmerize link. The capacity of the 5G link is limited to 5 Mbit/s since we configured it using a 10 MHz channel. Employing wider channel bandwidth can improve the capacity of the 5G link but we preferred this setup since 10 MHz channel provided a stable connection with our private femto cell base station. The latency over the LEO satellite link is very high, up to 200 ms. The main contributor to this is the public 5G connection that we use on the aggregation gateway to receive the data packets from the internet. Otherwise, the LEO connection has a link latency around 20 ms. Nevertheless, the high end-to-end latency influences MPTCP's link selections and it does not utilize the LEO satellite link unless the Meshmerize and 5G links are congested. This also aligns with our intended measurement scenario as we aim to utilize the LEO satellite link as the back-up connectivity. During measurements, the LEO link had capacity bottlenecks, hence, only up to 5 Mbit/s was achievable on the uplink channel. The reduced uplink performance is likely attributable to dynamic resource allocation, spectrum sharing among multiple users within the satellite beam, and adaptive modulation responding to varying channel and network conditions. Lastly, the initial link establishment time is only a few seconds with both the Meshmerize and 5G networks while it takes at least a few minutes with the LEO satellite network.

This demo setup used in the demonstration serves as a performance baseline for future combined NTN/TN 6G systems.

5.1.3 Measurement Results

This section delves into the evaluation of the collected data captures from the flight. We first investigate the time-varying data rate results from the individual links to understand how MPTCP changed the amount of data traffic allocated to the individual links. We also correlate it with the communication distance of the Meshmerize and the 5G links. Later, we also compare the 5G link data rate with its uplink signal-to-noise ratio (SNR). Then, we compare the total amount of data traffic sent via each link as well as the round-trip-time (RTT) distributions over individual links.

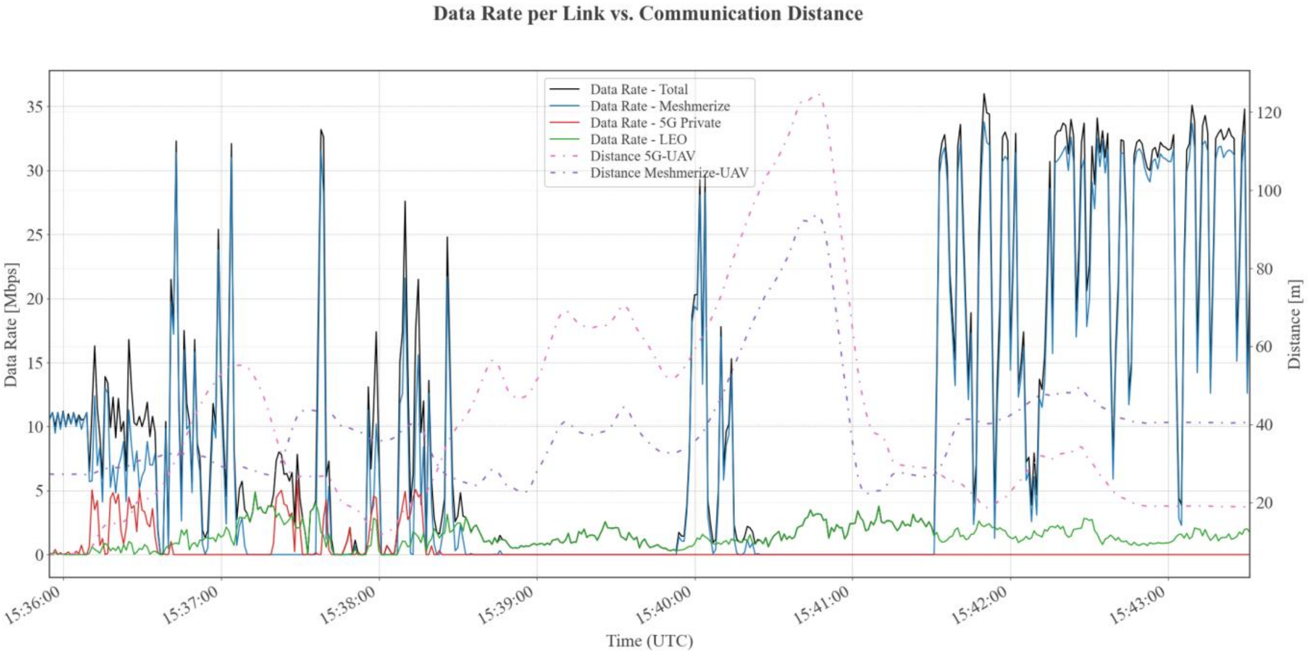


Figure 20. Achieved data rate per link versus communication distance

Figure 20 demonstrates the time-varying data rate distribution per individual links as well as the communication distance of the 5G and the Meshmerize links. Firstly, a roughly stable 10 Mbit/s data stream is present in the first ~20 seconds of the graph before the Airbus UAV takes-off. MPTCP allocates almost all the data to the Meshmerize link in this time range due to its favorable link latency and capacity. However, as soon as the Airbus UAV starts flying, the overall data rate (black line) starts to oscillate and the amount of data rate assigned to the Meshmerize link reduces. The doppler effect might have caused RF channel instability with the Meshmerize link. Hence, MPTCP begins to allocate ca. 5 Mbit/s data rate to the 5G link and <1 Mbit/s data rate to the LEO link (between ~15:16:20 and 15:16:40 in the graph). Between ~15:16:40 and 15:17:30, the 5G connectivity is broken; hence, MPTCP allocates data traffic also to the LEO satellite link. The data rate on the Meshmerize link fluctuates and observes data rate spikes up to ~35 Mbit/s. As MPTCP cannot send all the data packets, it stores them in the transport layer buffer and pushes them in a burst once the Meshmerize link has enough link capacity. Around 15:17:10, the Meshmerize link is also broken, then MPTCP allocates all the data traffic as much as possible to the LEO satellite link only. Therefore, the data rate on the LEO satellite link increases up to 5 Mbit/s between 15:17:10 and 15:17:30.

Between 15:17:30 and 15:18:30, both the 5G and the Meshmerize links are unstable. MPTCP tries to push as much data as possible from its buffer over both links, causing large data rate spikes. In the meantime, the communication distance of the Meshmerize and the 5G link increases and at ~15:18:30, both links are broken. Then, MPTCP still utilizes the LEO satellite link and can send a few Mbit/s data rate stably. After this time, if the Meshmerize link is established again, MPTCP sends all the buffered traffic over Meshmerize, causing data rate oscillations between 5 Mbit/s and 35 Mbit/s. However, the

5G link is never established again even if the communication distance reduces after ~15:41:00. This is potentially due to antenna misalignment on the Airbus UAV.

Overall, a total of ~1000 MB, 120 MB, and 60 MB of the data traffic was sent during the flight over the Meshmerize, 5G and LEO satellite links, respectively.

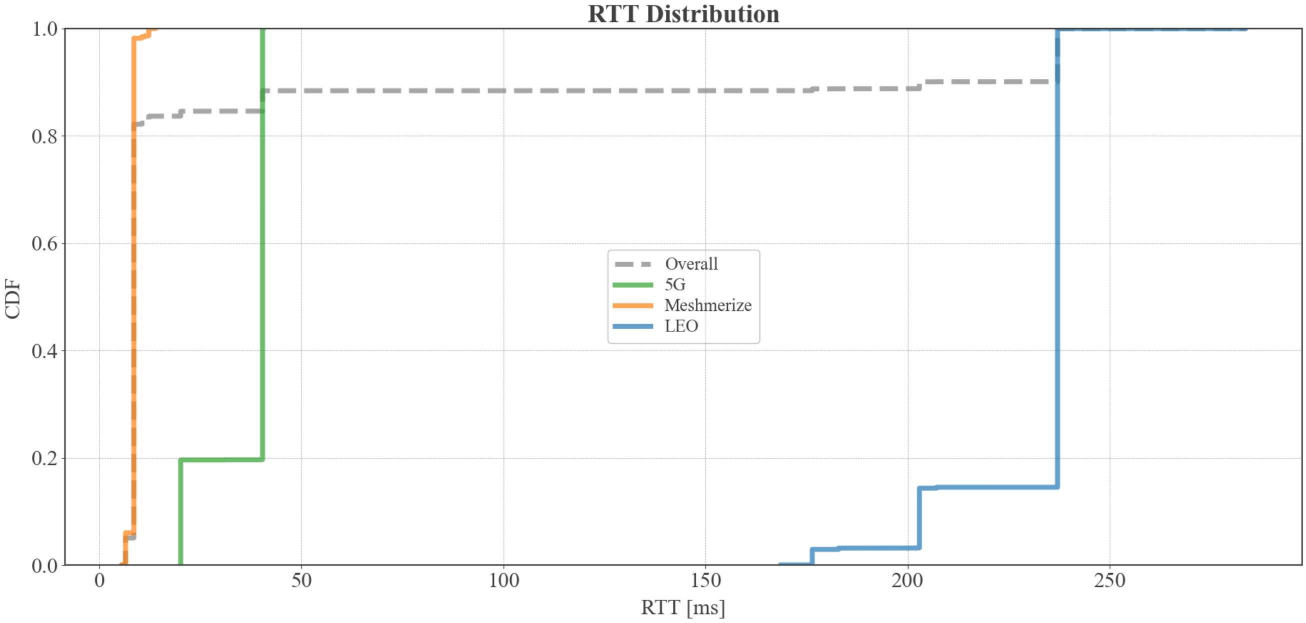


Figure. 21 Round-Trip-Time Distribution

Figure. 21 shows the achieved RTT distribution during the flight. This result is only based on MPTCP's link RTT estimations that are encoded in MPTCP packet headers. Hence, RTT outliers that occur during link outages are not taken into account in this graph. Overall, the graph shows that, based on MPTCP's RTT estimations, the RTT could stay below 10 ms for ~80% of the time as those packets were sent via the Meshmerize link. The max. RTT reaches as high as 280 ms, due to high end-to-end latency over the LEO satellite connection, majorly caused by the 5G public network connectivity for internet connection at the aggregation gateway.

5.1.4 Discussion of the Results

The goal of this flight campaign was to evaluate how we can utilize multilink connectivity in an efficient way to ensure seamless and reliable connectivity for safety-critical drone applications, such as remote piloting. The multilink configuration can be handled in different layers of the network stack, and in this test, we evaluate the coordination of the links using a multipath protocol on the transport layer, namely MPTCP.

The measurement results show that MPTCP dynamically adapts the data traffic allocation for each link depending on the changing link conditions. In our triple link configuration, even if two of the links are broken, at least one of the links were available and MPTCP utilized it to ensure continuous connectivity. In this regard, the measurement results demonstrate that seamless and continuous connectivity was achieved using Meshmerize, 5G and LEO satellite links during the flight even during coverage holes and link congestion.

In Figure 20 we observe that the data rate largely fluctuates between 5 and 35 Mbit/s, mostly depending on the availability of the Meshmerize link. Such data rate fluctuations are not favorable for real-time video streaming, since the video packets are delayed for a while and then sent to the receiver altogether as a burst. In practice, the video player would observe video stalls during buffering periods at the sender. When all the buffered packets arrive at the received, they would be useless for the remote pilot due to the expiration of the video information. One of the main reasons behind this behavior is that the data rate of the application, 10 Mbit/s, could only be handled by the Meshmerize link on its own. Because the maximum capacity of the 5G and LEO satellite links were limited to 5 Mbit/s. Hence, during the times when the Meshmerize link was broken, MPTCP always had to buffer data as it could not push all the application data traffic via 5G and LEO satellite links. In addition, due to the large link latency of the LEO connectivity and 5G link quality fluctuations, MPTCP could not utilize the full capacity of these links. Therefore, this three-link configuration with MPTCP is not suitable to support continuous and low-latency 10 Mbit/s streaming. Nevertheless, if the data rate of the application traffic is configured below the minimum capacity of individual links (e.g., <5 Mbit/s in this setup), then we can observe stable data rate and low-latency data delivery to the receiver. Because, even if two out of the three links become unavailable, the remaining link itself can still provide sufficient capacity for continuous streaming without buffering.

In real-life, public 5G networks can provide larger coverage than the ~100 m range of our femto cell private 5G base station. Nevertheless, we intended to limit the 5G coverage to evaluate how well MPTCP could handle out-of-coverage scenarios. Also, 5G links can have more capacity in real-life (e.g., around 40 Mbit/s or even more), but we decided to limit the link capacity to 5 Mbit/s to achieve stable connectivity. Hence, in real-life with public cellular networks, it is likely that MPTCP would more evenly distribute the data traffic between Meshmerize and 5G links, rather than allocating more data to the Meshmerize link.

We used the MPTCP protocol in this setup due to its software maturity however; the TCP protocol is intrinsically not designed for real-time connectivity. TCP is a loss-less and in-order data delivery protocol; therefore, it relies on transport-layer retransmissions to avoid packet losses and buffers to resolve out-of-order packet arrivals. These features are not suitable for low-latency data transmission. Hence, other multipath transport protocols such multipath datagram congestion control protocol (MP-DCCP) or multipath QUIC (MP-QUIC) should also be evaluated in this setup. As they are based on the user datagram protocol (UDP), these multipath protocols can provide data transmission in a best-effort manner without retransmissions, which can alleviate the data rate fluctuations, hence more stable data streaming.

5.1.5 Conclusions

Overall, the results show that in our multilink configuration setup, at least of the links are always available, hence continuous connectivity for drones can be maintained even if 1 or 2 links experience coverage gaps. MPTCP can dynamically handle the data traffic allocation, changing the data rate to each link over time by sensing the link capabilities. Hence, the results show that this multilink scenario with different links and different link properties can be also managed on the transport layer. Also, even if the links are broken, once they are restored, the multipath transport layer takes the links into account again and try to utilize them efficiently. However, MPTCP is not suitable for real-time data streaming since it aims to deliver all the packets, which is unsuitable for the real-time connectivity demands of drones.

5.2 Explainable AI for UAV Handover Management

As part of Task 5.3, we developed a novel framework for explainable handover decision-making in drone (aka UAV) communication scenarios. This framework integrates deep reinforcement learning, post-hoc feature attribution methods, and Large Language Models (LLMs) to enable interpretable decision support for any deep learning-based model in the wireless communications domain.

Our primary contribution is an explainable, learning-based approach to mobility management that can generalize beyond UAV use cases to other domains in radio access network (RAN) control and intelligent network automation.

5.2.1 Explainability Framework

We trained a handover decision policy using Deep Q-Networks (DQN) on a randomized, parameterized simulation environment. The training objective was to minimize handover frequency while maintaining connection reliability. The environment models a multi-cell wireless network with dynamic UAV trajectories and signal conditions.

The trained policy outputs discrete actions corresponding to either maintaining the current connection or initiating a handover to one of the neighboring base stations.

To interpret the behavior of the RL policy, we implemented a post-hoc explainability pipeline using DeepSHAP, an extension of SHAP (SHapley Additive exPlanations) tailored for deep neural networks. This allows for local feature attribution that quantifies the contribution of each input feature to a specific decision.

Explanations are generated for all possible handover actions at a given decision point (e.g., "stay connected", "handover to cell 217", etc.), enabling counterfactual reasoning - not only why an action was chosen, but why alternative actions were not.

5.2.2 Inference on Real Data

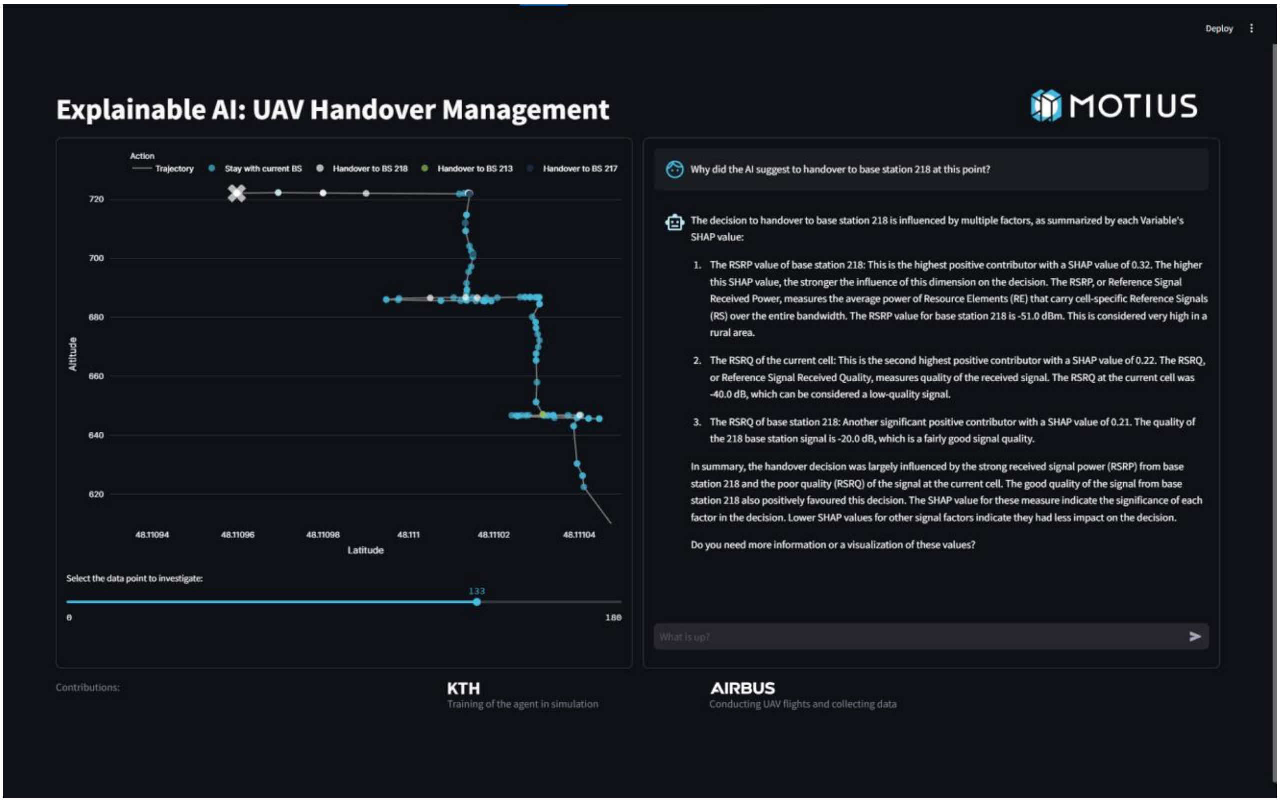
To evaluate the proposed framework, we applied it to real-world LTE data collected from UAV flight trials. These tests were conducted using a commercial DJI M600 drone outfitted with dual LTE modems. The dataset consists of timestamped GPS coordinates, network-side LTE metrics such as Reference Signal Received Power (RSRP) and Reference Signal Received Quality (RSRQ), and system parameters reflecting live operational conditions. Six distinct flight trajectories were recorded in a rural setting, providing a diverse set of mobility patterns and signal variations.

To ensure consistency with the simulation environment used during training, GPS data from the flight logs was normalized to match the unit-less coordinate system of the reinforcement learning (RL) model. We focused on the four base stations that exhibited the strongest average RSRP in the data, extracting and aligning their signal measurements with the UAV’s flight data based on the synchronized timestamps. This enables inference of the RL policy on real flight sequences.

SHAP values are computed for each possible action at every decision point. These attributions allow understanding not just why the model recommends a particular handover, but also why it rejects alternatives - enabling counterfactual reasoning across the action space.

5.2.3 Demonstrator Interface

To make the system accessible to a broader audience, we developed an interactive, web-based demonstrator. This tool visualizes UAV trajectories step by step, highlighting the policy's handover decisions along the path. The user can select a data point on the left and inspect the measurements and which action the policy chooses in that situation. For each decision, SHAP-based explanations are computed. Then, questions about the decision can be asked in natural language in a chat window (e.g., “Why did the UAV hand over to base station 218?”).



Why did the AI suggest to handover to base station 218 at this point?

The decision to handover to base station 218 is influenced by multiple factors, as summarized by each Variable's SHAP value:

1. The RSRP value of base station 218: This is the highest positive contributor with a SHAP value of 0.32. The higher this SHAP value, the stronger the influence of this dimension on the decision. The RSRP, or Reference Signal Received Power, measures the average power of Resource Elements (RE) that carry cell-specific Reference Signals (RS) over the entire bandwidth. The RSRP value for base station 218 is -51.0 dBm. This is considered very high in a rural area.
2. The RSRQ of the current cell: This is the second highest positive contributor with a SHAP value of 0.22. The RSRQ, or Reference Signal Received Quality, measures quality of the received signal. The RSRQ at the current cell was -40.0 dB, which can be considered a low-quality signal.
3. The RSRQ of base station 218: Another significant positive contributor with a SHAP value of 0.21. The quality of the 218 base station signal is -20.0 dB, which is a fairly good signal quality.

In summary, the handover decision was largely influenced by the strong received signal power (RSRP) from base station 218 and the poor quality (RSRQ) of the signal at the current cell. The good quality of the signal from base station 218 also positively favoured this decision. The SHAP value for these measure indicate the significance of each factor in the decision. Lower SHAP values for other signal factors indicate they had less impact on the decision.

Do you need more information or a visualization of these values?

Figure 22. User interface of the Explainable AI: UAV Handover Management Demo

To answer such questions the LLM receives a structured input containing the Policy's input values, the selected action, and the SHAP values for all available choices as context to the user's query. The LLM understands the fundamentals of the RL and SHAP methods and can interpret the data correctly so users do not need data science knowledge to use the tool to understand the model's behavior.

To assess the utility of our system's explanation, we conducted a preliminary qualitative analysis in our project. A small group of domain experts used our system's graphical user interface and provided their opinion on the usefulness of the explanation. The small-scale informal analysis was helpful in providing observations on the intuitiveness of the interface and the usefulness of the resulting explanation. The overall perception was that the explanation was useful. A controlled user study is a point of future work.

6 Task 5.4: 3D network demonstration with multiple drones flying as a swarm in coordination

Lead: Lakeside Labs, **Contributors:** Twins, RED Bernard, Logistik Center Austria Süd, Meshmerize.

In Task 5.4 a swarm of drones (aka UAVs) is used to enable novel use cases. On the one hand, they use existing network infrastructure as mobile aerial users and on the other hand, they enhance the network capabilities for ground users. Combining the existing technologies 5G and Wi-Fi mesh, the drones enable local coordination and network services for edge and cloud processing.

Both use cases were demonstrated at the premises of the Logistik Center Austria Süd in Fürnitz, Austria shown in Figure 23.

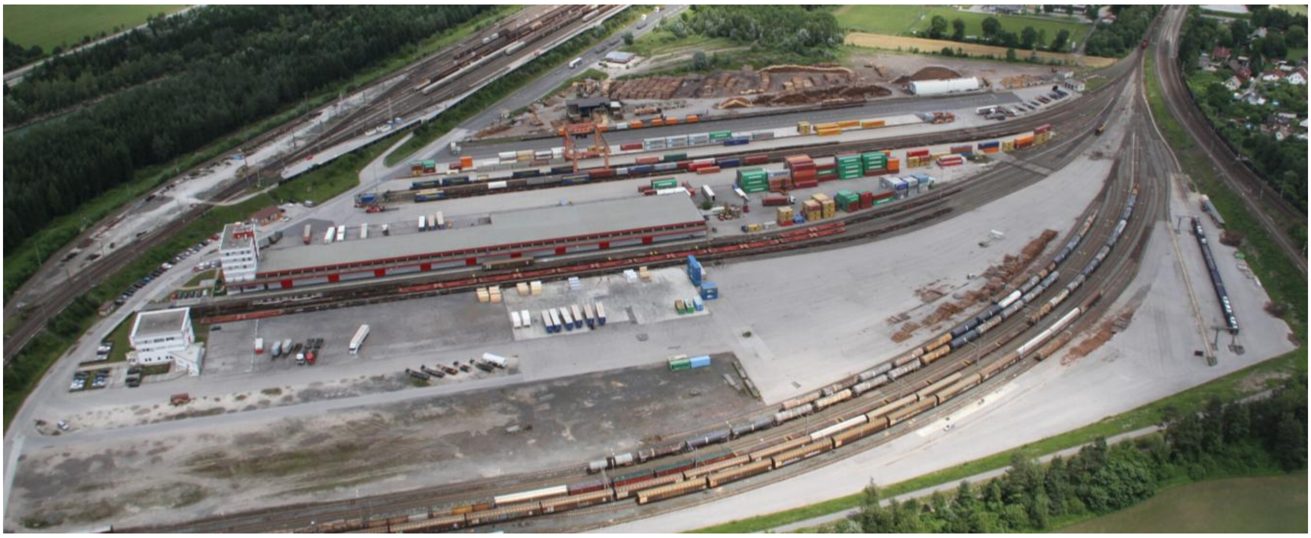


Figure 23. Logistik Center Austria Süd in Fürnitz, Austria

Since the current legal framework does not support the use case of autonomous drone swarms, for every drone there was a safety pilot to intervene in case of emergencies. Nonetheless, we conducted research in the legal obligations that come with the operation of autonomous drone swarms.

One measure to reduce the ground risk for non-involved persons is to keep a safety margin around the mission area visualized in Figure 24 and to automatically engage a kill switch in case a drone leaves the pre-defined mission area. An enhancement would be the integration of parachutes to reduce the ground risk even further.

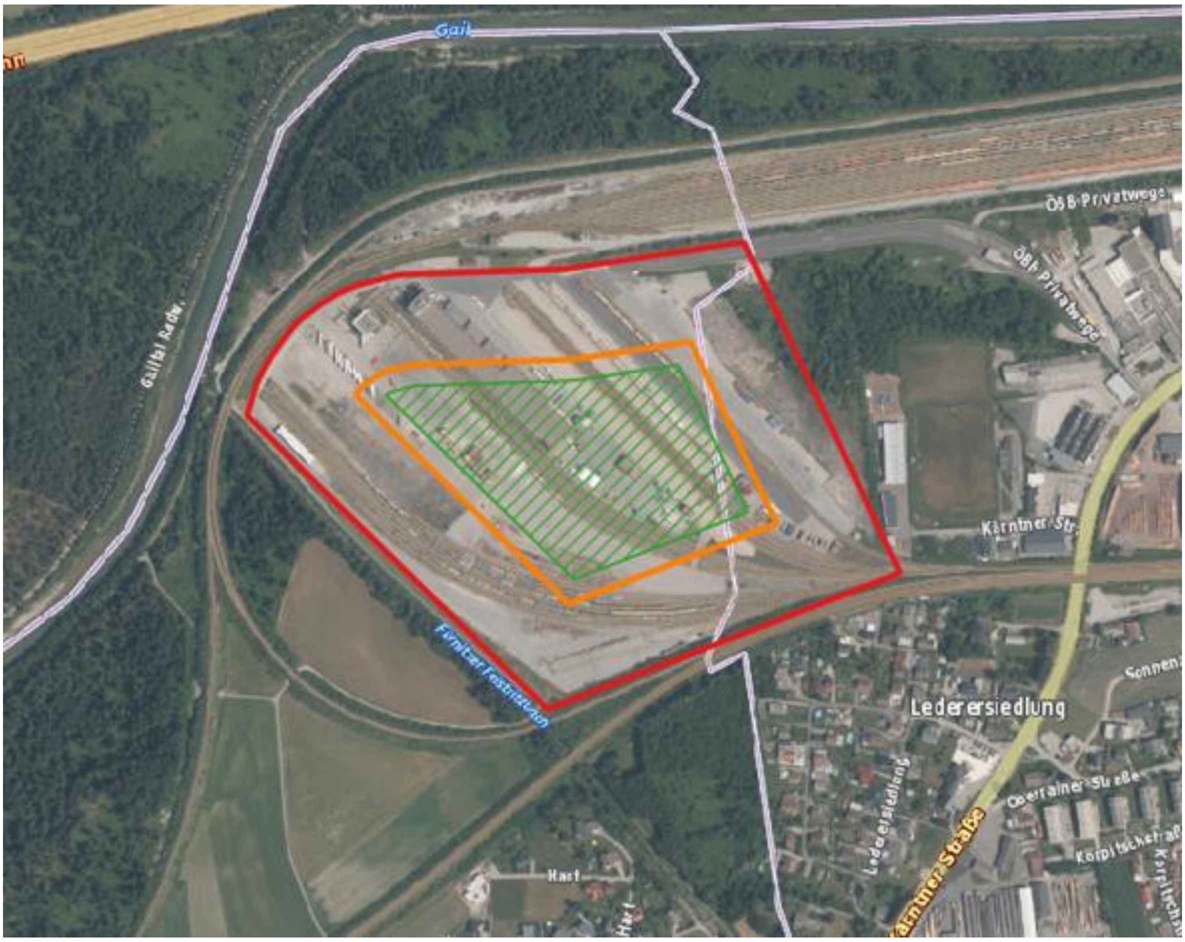


Figure 24. Reduced mission area with safety margin to reduce the ground risk

6.1 Use case 1: Logistics centers supported by swarms of drones

In this use case a swarm of drones operates in a logistics center to support the localization of containers. It is a two-phase mission where first a drone creates an aerial overview image from a higher flight height (70 m – 100 m) (see example image in Figure 25) to identify the location of container stacks. These locations then serve as input for a swarm of drones. The drone swarm autonomously allocates the stacks to the individual drones for a close-up inspection at a lower flight height.

The optimal flight height has been determined in a series of test flights conducted at a flight height between 5m and 80m, by collecting and analyzing aerial images of container stacks. The best results in terms of image quality and object detection were achieved for lower altitudes, particularly up to 25m. The flight height of the swarm must also take operational safety into account. Therefore, a minimum flight height of 18m has been defined, based on the maximum height of container stacks and the operational space required by reach stackers used for container transport. Balancing these factors results in an optimal flight height range of 18m to 25m.

The goal of the inspection is to identify the containers using their IDs and calculate their GPS location. The focus of this use case is on the low-flying drone swarm which is discussed in the following.

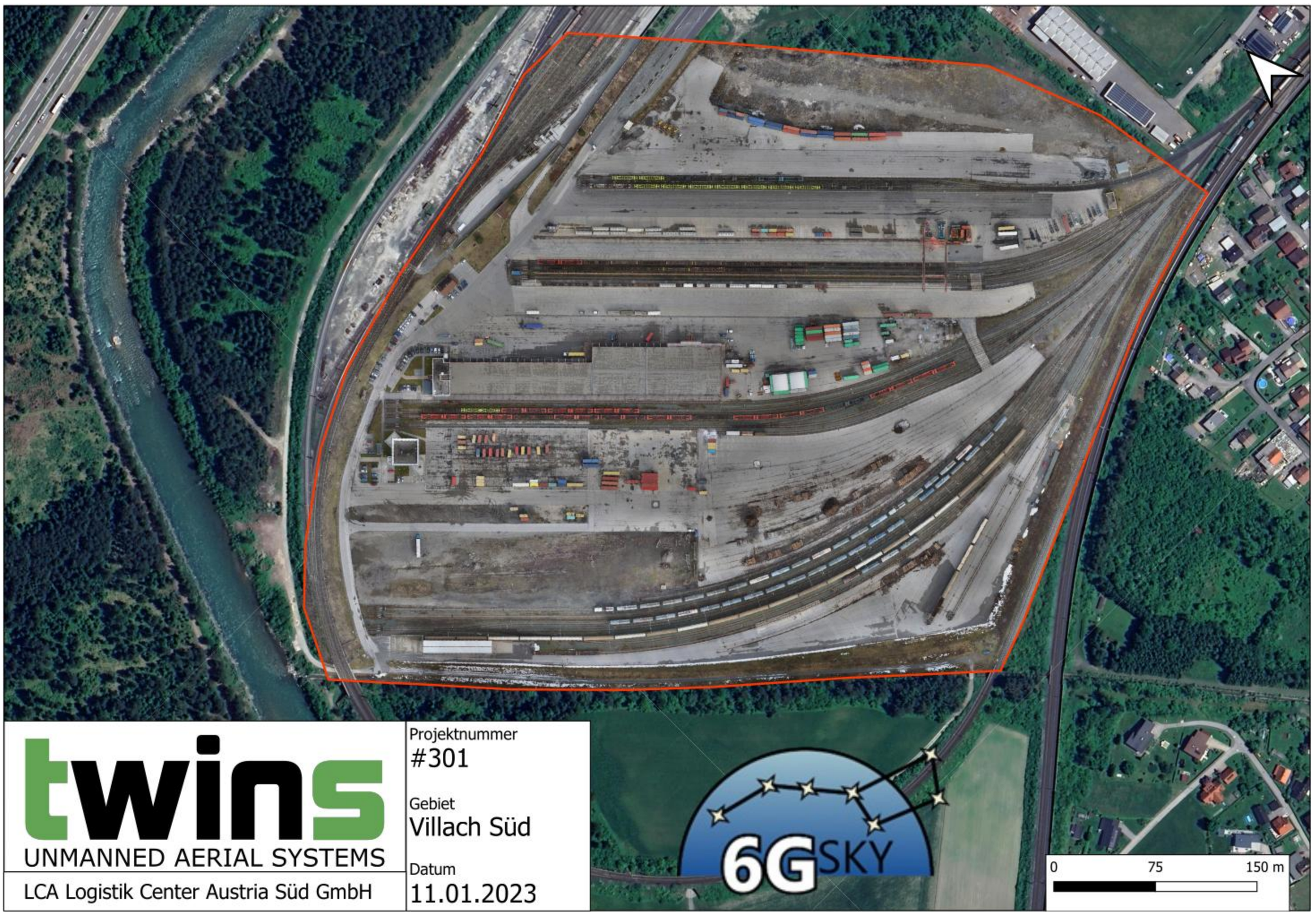


Figure 25. Overview image of LCA created using photos collected by twinFold GEO drone in Jan 2023

6.1.1 Network Technologies

This use case uses both the 5G cellular network and local Wi-Fi mesh networking. The mesh network provides local communication regardless of available infrastructure. This enables local coordination of the drones in the swarm which is important for reliable operation of the swarm, e.g., for task allocation or collision avoidance. The 5G network provides internet connectivity to enable transmitting the payload data to cloud services for further processing and remote operation.

In the experiments, four Wallys DR4029 wireless access points were used as mesh devices for the drones and the ground control station (GCS). The drones were connected to the 5G cellular network using Quectel RM500Q UE. The setup of the drones has been evaluated in the lab emulations beforehand and hence further details about the configuration can be found in Chapter 4.2.1.

6.1.2 System Components and Architecture

The use case is implemented by three main components shown in Figure 26:

Figure 26. High-level architecture of the logistics use case

- Drone swarm
- Ground control station
- Cloud server

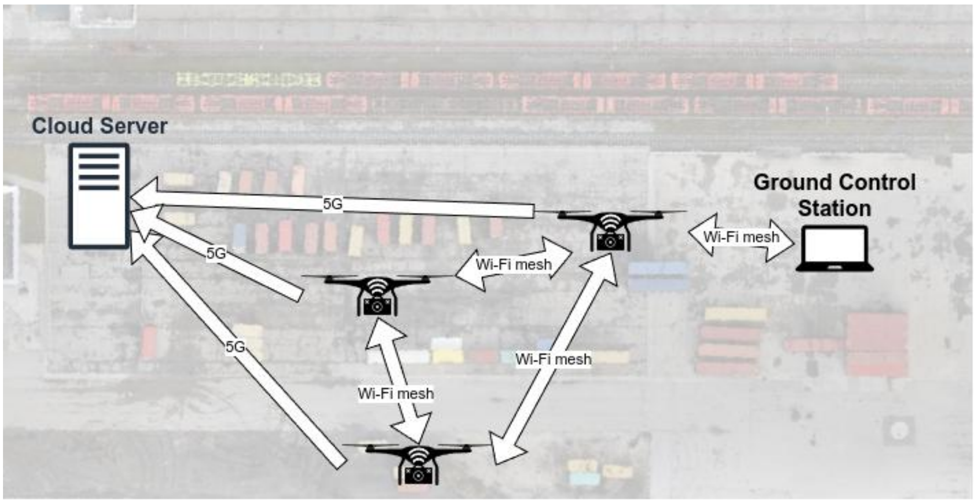


Figure 26. High-level architecture of the logistics use case

Drone swarm

The drone swarm consists of three twinFOLD GEO drones described in Section 3.1. They communicate locally between each other and the GCS using a mesh Wi-Fi described in Section 3.2. It enables them to receive mission commands from the GCS, stream telemetry to the GCS, and perform robust coordination in absence of infrastructure. The communication to the cloud server is achieved through the 5G network. The drones offload the payload data, i.e., the captured images together with telemetry data, to the cloud server for post-processing and data fusion.

Each drone is equipped with a flight controller for low-level control, an on-board companion computer for communication, data processing, and mission control, and a mesh node and 5G modem for communication. The detailed setup of the drones' communication and computation hardware is visualized in Figure 27.

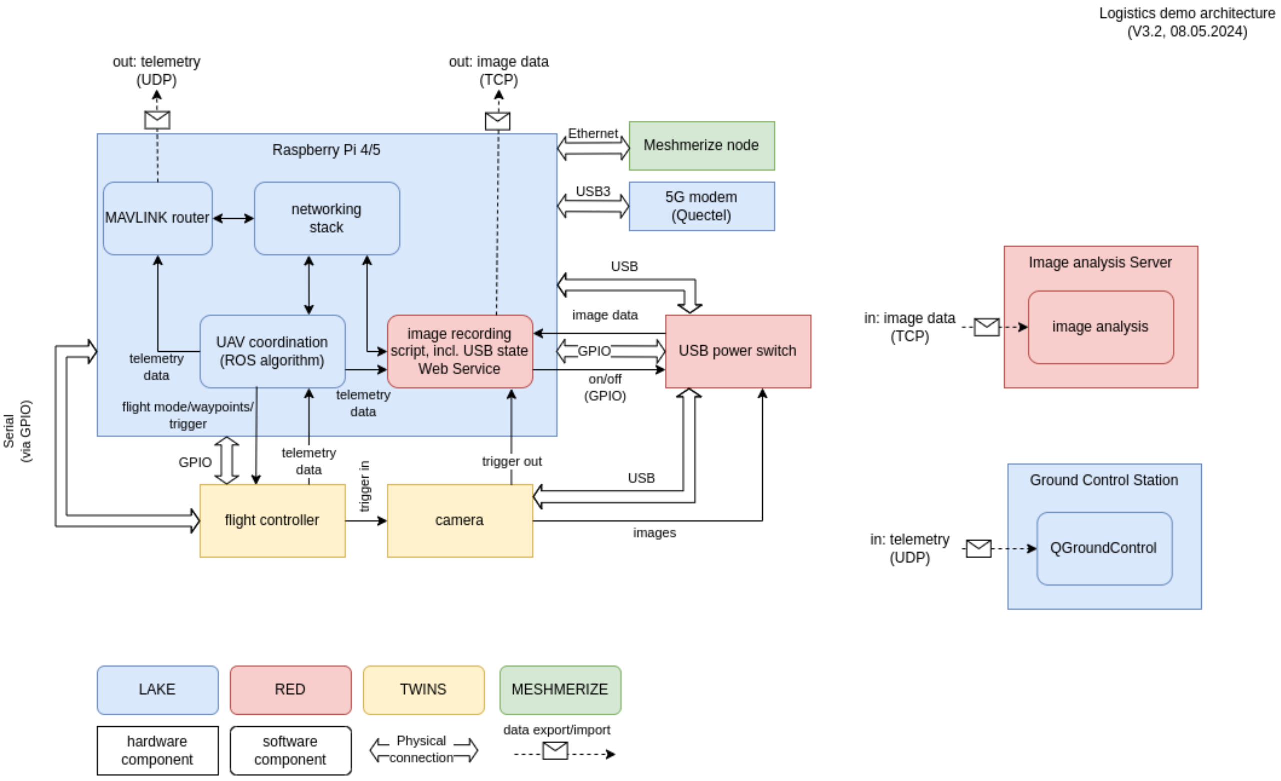


Figure 27. Communication and computation hardware on board the drones

Ground control station (GCS)

The GCS is responsible for mission execution and monitoring. It is implemented on a laptop running Ubuntu Linux with ROS2[1] (see Figure 28). It is connected to the drones over mesh Wi-Fi. It is used to start the mission and track its progress (left side of Figure 29) and visualize the drone telemetry (right side of Figure 29).

[1] ROS2 documentation: https://docs.ros.org/en/humble/index.html, accessed 2025-05-26

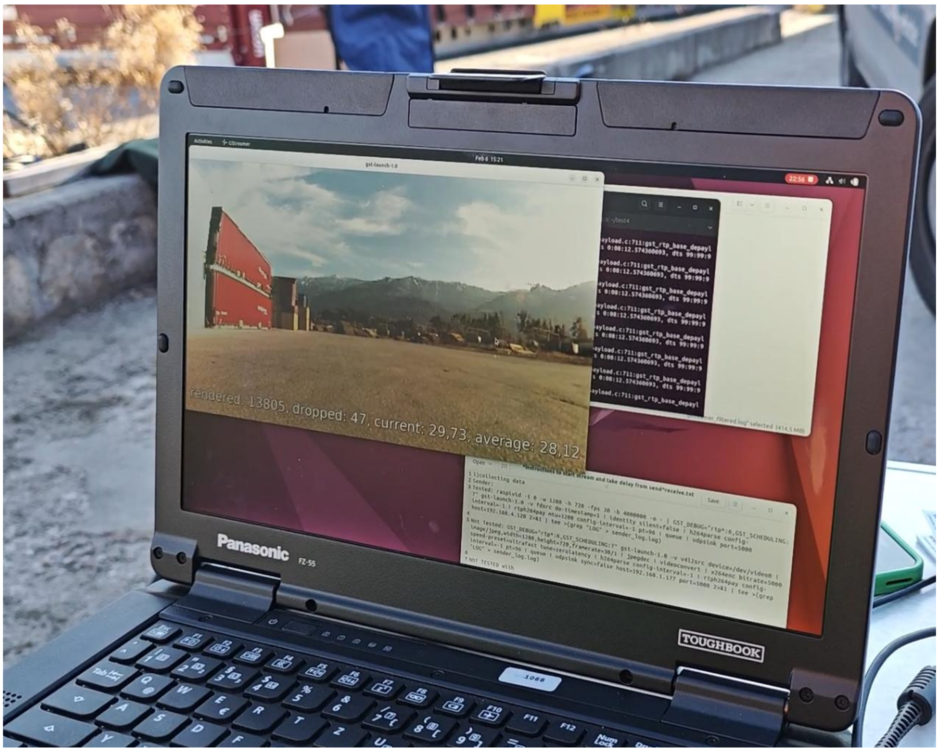
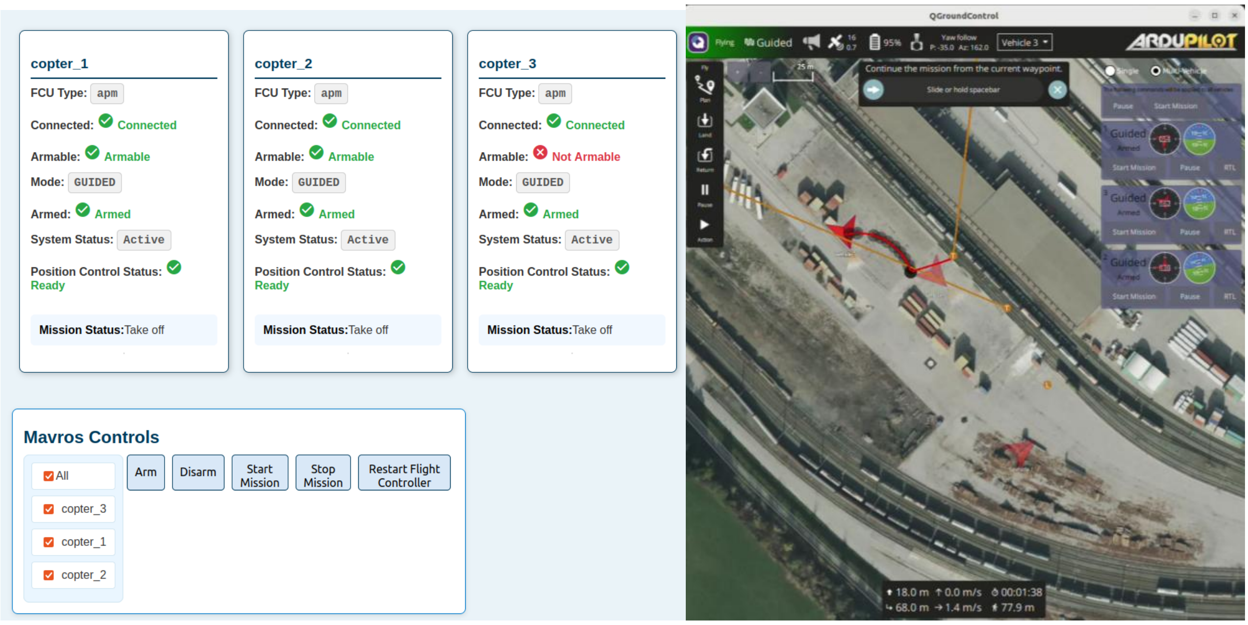

rendered: 13805, dropped: 47, current: 29,73, average: 28,12
Panasonic
TOUGHBOOK

Figure 28. Ground control station laptop.



copter_1

FCU Type: `apm`

Connected: ✓ Connected

Armable: ✓ Armable

Mode: `GUIDED`

Armed: ✓ Armed

System Status: `Active`

Position Control Status: ✓ Ready

Mission Status: Take off

copter_2

FCU Type: `apm`

Connected: ✓ Connected

Armable: ✓ Armable

Mode: `GUIDED`

Armed: ✓ Armed

System Status: `Active`

Position Control Status: ✓ Ready

Mission Status: Take off

copter_3

FCU Type: `apm`

Connected: ✓ Connected

Armable: ✗ Not Armable

Mode: `GUIDED`

Armed: ✓ Armed

System Status: `Active`

Position Control Status: ✓ Ready

Mission Status: Take off

Mavros Controls

☑ All
☑ copter_3
☑ copter_1
☑ copter_2

Figure 29. Ground control station with vehicle control (left) and telemetry view (right)

Cloud server

The cloud server for image processing is located off-site and connected to the drones via the Internet. It collects the telemetry and payload data of all drones which use 5G to send the data during flight, every time a drone finishes circling a container stack. The server then performs optical character recognition (OCR) and fuses the resulting data with the telemetry data of the drone to map the container IDs to GPS locations.

6.1.3 Methodology and Implementation

Coordination of drone swarm

The drone swarm operates completely autonomous and coordinates its actions in a distributed manner without central control. While the GCS acts as central access for the operator of the swarm, it is only needed to send the mission start command. All computations for coordination, mission execution, and navigation are performed locally on board the drones.

The implementation is homogeneous among all drones and consists of three levels:

1. Mission level
2. Common functionalities
3. Hardware abstraction

Mission Level

The mission is implemented as behavior tree which allows modular and flexible design of the mission. A strongly simplified version is shown in Figure 30. In this example, all branch nodes (i.e., nodes with children) sequentially execute all child nodes and then return control to their parent node. All leaf nodes execute the stated behavior and then return control to their parent node.

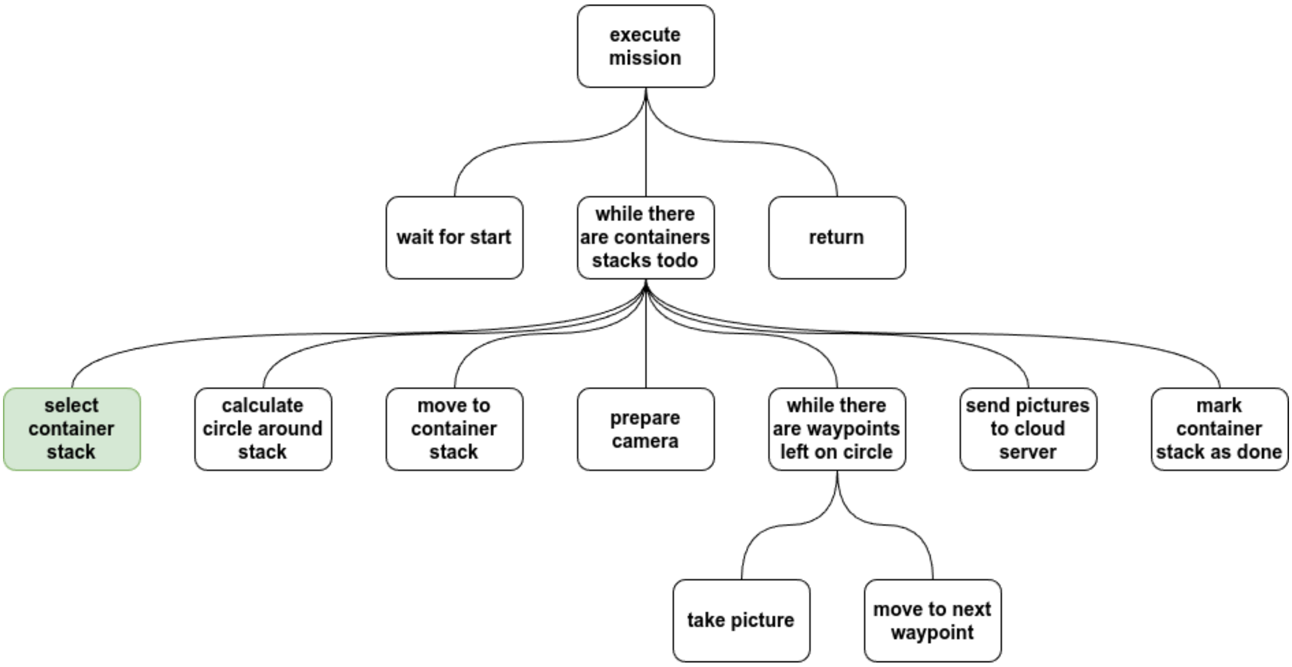

execute mission
wait for start
while there are containers stacks todo
return
select container stack
calculate circle around stack
move to container stack
prepare camera
while there are waypoints left on circle
send pictures to cloud server
mark container stack as done
take picture
move to next waypoint

Figure 30. Simplified behavior tree model of the mission with the core task allocation algorithm highlighted in green

Common functionality level

The common functionality level implements a variety of behaviors that are independent of the specific mission and independent of the underlying drone hardware. Among others, this includes primitive behaviors such as waypoint navigation or camera triggering, communication for synchronizing the origin of the common coordinate system, and coordination for task allocation, i.e., selecting a container stack.

The common functionality level implements the core functionality of the use case: The distributed task allocation to assign container stacks to drones. A detailed description of this algorithm can be found in deliverable D3.3. In the ROS implementation of the algorithm, the container stacks are given to the drones as a list of tasks that have a defined GNSS coordinate, an altitude at which the containers shall be inspected, and a radius at which the drones circle the containers.

The task allocation algorithm is constantly running in the background as soon as the ROS algorithms are started. At the beginning of the mission, before taking off, the drones already select the most appropriate task using the cost function in Equation 1.

$$\mathbf{t} = \underset{\mathbf{t}_i \in \mathbf{T}}{\arg\min} \left\{ \lambda \cdot \|\mathbf{v} - \mathbf{t}_i\|^2 + \frac{1-\lambda}{|\mathbf{T}_\mathrm{a}|} \cdot \sum_{\mathbf{t}_j \in \mathbf{T}_\mathrm{a}} \|\mathbf{v} - \mathbf{t}_j\|^2 \right\}$$

Equation 1: Task allocation cost function.

It minimizes both the squared distance of the agent location (**v**) to the task location (**t**) and the average squared distance to all tasks assigned to other vehicles ($\mathbf{T_a} \subseteq \mathbf{T}$). The parameter λ adjusts the weight of the two distance metrics. When λ = 1, drones prioritize coverage over connectivity, compared to λ = 0, where drones prioritize connectivity. The side effect of a strong focus on connectivity is a reduced coverage, as drones fly in close formation to stay connected.

During the mission, the drones constantly inform the other drones about their currently selected task and the progress they made in scanning the corresponding containers. They use the Wi-Fi mesh network, possibly through multi-hop to disseminate this information in the swarm.

Nevertheless, there are situations, where more than one drone selects the same task. Once the drones detect such a conflict, the drone that has made less progress on that task yields the task to the other drone. In case both drones made the same progress, the drone with the lower task cost is given priority.

To work on a task, i.e., scan the containers in a selected stack, the drones fly straight to the container stack, circle the container stack while repeatedly taking pictures at a pre-defined distance interval, and update the task progress, i.e., percent of the circle completed. Once the circle (and the task) is completed, the drones upload the pictures to a cloud server for processing. Then, they select the next uncompleted task that is not assigned to another drone. If there are no more tasks left, they return home.

Hardware abstraction level

The hardware abstraction level implements the communication with the ArduPilot flight controller using the MAVLink protocol. It uses the ROS topics and services provided by MAVROS to retrieve status information from the drones (e.g. status, position) and send navigation setpoints (e.g. waypoints) and camera controls (e.g. triggers). This level provides a hardware independent interface to the upper layers to allow changing the underlying hardware (e.g., using PX4 flight controllers) or communication interfaces (e.g. using ROS Distributed Data Service instead of MAVLink).

Image processing and analysis

For image processing and analysis, the drones are equipped with high-resolution cameras. On the cloud server an optical character recognition (OCR) software is deployed to detect and extract shipping container IDs from the aerial images. Using the drone's GPS coordinates, the geolocation of each container was also recorded. The drones fly autonomously over the container yard, capturing images of container sidewalls and doors where ISO 6346[2]-compliant IDs are typically displayed. During flight, image data was uploaded to the cloud, where a server-based OCR system processed the images in near real-time. The system successfully identified container IDs, demonstrating the feasibility and accuracy of combining aerial imaging, GPS tracking, and OCR technology for automated container identification and localization in logistics and port operations.

Data Flow Overview

During the field demonstration, the data collection and processing workflow followed a structured pipeline:

1. **Autonomous Image Capture:** Each drone performed an automatic circular flight around a designated container stack, capturing high-resolution images (6000x4000 Pixel) at regular intervals (every ~2m for the first flight and ~5m for the second and third flight) to ensure complete coverage of container sidewalls and doors. For each image taken, the current GPS coordinates and the heading of the drone were also recorded, enabling precise geolocation of the container positions within the yard.
2. **Local Image Transfer:** Once the circular flight path was completed, the captured images were transferred from the onboard camera to the Raspberry Pi system on the drone.
3. **Wireless Image Upload:** The Raspberry Pi uploaded the images to a remote SFTP server using a 5G LTE connection, enabling near real-time offloading of data.

[2] ISO 6346: 2022:Freight containers - Coding, identification and marking, https://www.iso.org/standard/83558.html, accessed 2025-06-10

4. **OCR Processing:** A server running the OCR detection software periodically accessed the SFTP server, downloaded the new images, and performed image evaluation and container ID extraction.

This flow enabled efficient data acquisition, wireless transmission, and centralized processing, demonstrating a scalable approach for automated container identification using aerial platforms.

OCR Pipeline Description

The container ID recognition was performed using a three-stage OCR pipeline specifically designed to handle ISO 6346-compliant IDs from drone imagery. In the first stage, a YOLOv8[3] Nano object detection model (input size:2048×2048 pixels) was used to detect container ID regions, handling both vertical and horizontal text orientations. The detected ID regions were cropped and passed to the second stage, where a second YOLOv8 Nano model (input size: 480×480 pixels) was applied to detect and segment individual characters within each cropped region. In the final stage, each character crop was classified using a MobileNetV3[4] Small model (input size: 32×32 pixels).

All models were trained using a combination of previously recorded drone imagery and publicly available container ID image data, including resources from websites[5]. This diverse dataset helped the models generalize across different container styles, lighting conditions, and ID placements.

To ensure output validity and improve recognition accuracy, post-processing rules were applied: the first four characters must be letters (owner code and equipment identifier), and the final seven must be digits (serial number and check digit). This modular approach provided a lightweight yet effective solution for robust container ID reconstruction in challenging aerial imaging conditions.

Data Fusion and Post-Processing

For each container stack, approximately 100 images were captured from different angles and positions to ensure full coverage. While individual OCR results were generally reliable, certain challenges –

such as partial occlusions, variable lighting conditions, dirt, or physically damaged container ID markings – occasionally led to recognition errors or incomplete outputs. To compensate for these

[3] https://roboflow.com/model/yolov8, accessed 2025-05-22

[4] Howard et al., „Searching for MobileNetV3", Proc. ICCV 2019, https://doi.org/10.1109/ICCV.2019.00140 accessed 2025-05-22

[5] such as www.prefixlist.com, accessed 2025-05-22

issues and produce a coherent interpretation of the scene, a post-processing and data fusion step was applied.

The post-processing proceeded under the assumption that the full list of container IDs present in the yard (ground truth) was known, but their exact locations (i.e., which container stack each container belonged to) were not. All detected OCR results from the image set were compared against the ground truth list using the Levenshtein distance[6] – a string similarity metric that accounts for insertions, deletions, and substitutions. This produced a distance matrix of size *[#detections]* × *[#ground truth IDs]*.

For each ground truth container ID, the detection with the lowest Levenshtein distance was selected as its best match, provided the score was ≤ 3. If the same best match was detected across multiple images, the mean GPS coordinates of all matching detections were calculated to estimate the container's final location. This process resulted in a robust and georeferenced mapping of ground truth container IDs to container stacks based on the drone's recorded positions at the time of image capture.

Remark on the calculation of the GPS coordinate: The GPS location of each container was estimated using the drone's onboard GPS coordinates in combination with its heading information. Specifically, as the drone flew in a circular pattern around the container stack, each image was associated with a corresponding heading and radius from the stack center. By projecting the detected container ID positions relative to the drone's orientation and distance, the system was able to approximate the container's actual ground location.

6.1.4 Experiment Setup

The demonstration was performed on the container terminal of the Logistik Center Austria Süd in Fürnitz, Austria. Figure 31 shows a satellite image of the environment and placement of three drones, three container stacks:

- **A**: Observation altitude 18 m, circle radius 35 m
- **B**: Observation altitude 25 m, circle radius 40 m (higher altitude due to nearby buildings)
- **C**: Observation altitude 18 m, circle radius 30 m

and the ground control station. Please note that the satellite image does not reflect the container stack position at the day of the experiment. The size of the container circle was chosen according to the size of the container stacks. The spacing between picture locations was 2 m and 5 m, depending on the flight. The parameter λ of Equation 1 was set to 1 in order for drones to select the closest container stack.

[6] Dictionary of Algorithms and data structures, https://xlinux.nist.gov/dads/HTML/Levenshtein.html, accessed 2025-06-10

In total, three flights were performed:

- **Flight 1**
  17.03.2025 | 17:14 – 17:20
  Image data collected for one container stack.
  111 images recorded.
- **Flight 2**
  18.03.2025 | 11:32 – 11:35
  Image data collected for two container stacks.
  102 images recorded.
- **Flight 3**
  18.03.2025 | 12:33 – 12:36
  Image data collected for two container stacks.
  99 images recorded.

Remark: One drone experienced hardware failure due to rotor contact with the FlexPCB USB cable connecting the camera to the Raspberry Pi. As a result, it was not possible to collect image data from this third drone and container stack C.

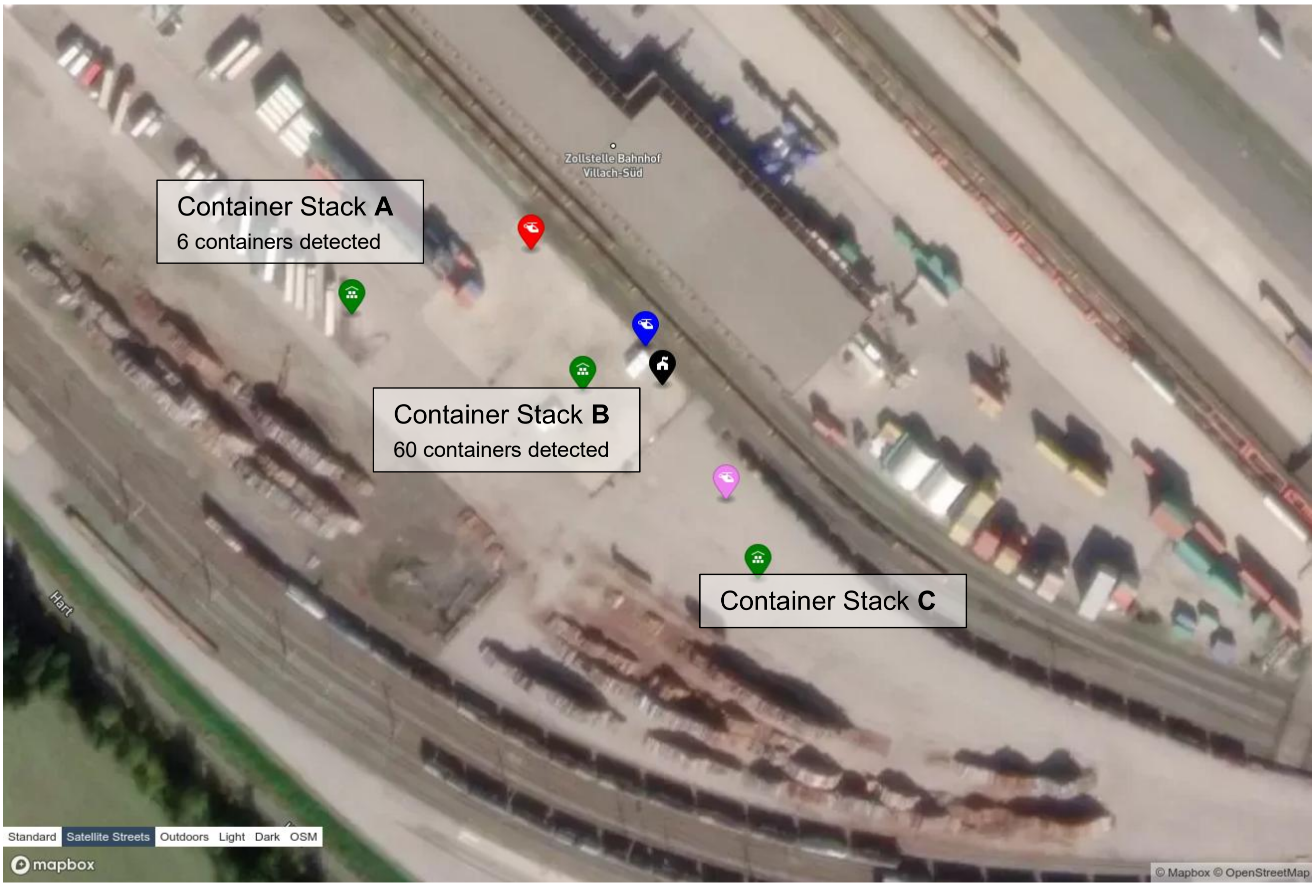


Figure 31. Setup of the demonstration at the logistics center with container stacks (green), drones (red, blue, violet), and ground control station (black)

6.1.5 Results and Analysis

ROS Network

The network of ROS nodes uses a data distribution service (DDS) as middleware for discovery, serialization and transportation. It enables distributed discovery and control over different quality of service (QoS) options for the transportation. However, the number of data packets exchanged during service discovery grows quadratically with the number of nodes. In our experiments we realized that the mesh Wi-Fi could not support more than two drones. Hence, we analyzed the network usage in more detail and compared it to other middlewares and protocols. In detail, we recorded the number of data packets exchanged during the discovery handshake of two drones and one ground control station (GCS). The result can be seen in Figure 32.

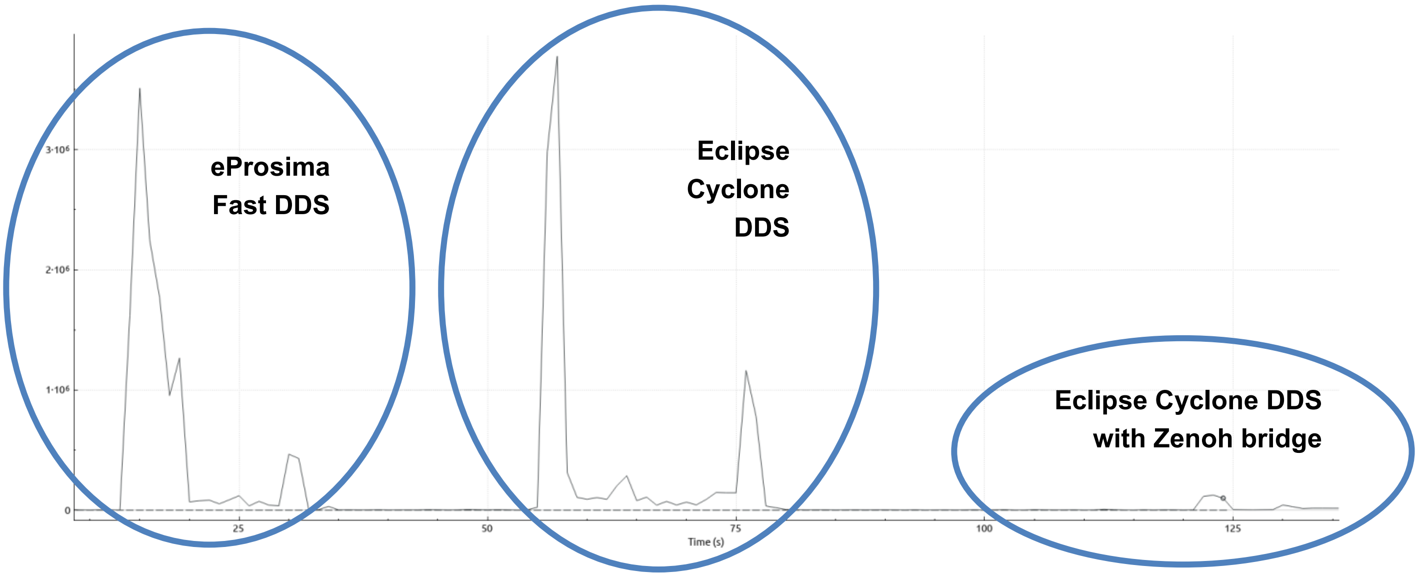


Figure 32. Number of data packets per second during handshake of ROS nodes

In the default setup, ROS uses the eProsima Fast DDS middleware implementation[7] which produced a peak of more than **3.5 million packets per second**. After exchanging the middleware for the Eclipse Cyclone DDS[8], the result was similar, reaching a peak of almost **3.8 million packets per second**. We then disconnected the different ROS instances by restricting the discovery to the local host only and bridged the DDS instances using the Zero Overhead Network Protocol (Zenoh)[9]. The resulting discovery "peak" reduced to approximately **125 thousand packets per second** which is a significant relief for the network. This is achieved through a different discovery mechanism and more efficient

[7] https://fast-dds.docs.eprosima.com/en/stable/fastdds/ros2/ros2.html, accessed 2025-05-14

[8] https://cyclonedds.io/, accessed 2025-05-14

[9] https://zenoh.io/, accessed 2025-05-14

messages: Instead of advertising all publishers and subscribers, only the desired subscriptions are advertised and grouped by drones (as opposed to the individual several dozen ROS nodes per drone) for compression.

Mission control and drone swarm coordination

The mission execution with the ROS framework successfully allocated every drone to one container stack. Each drone then flew an autonomous mission by calculating a circle around the container stack, flying to the closest point on that circle, taking pictures at a fixed distance of 5 m (2 m distance for the first flight proved to create too much redundant data), and then returning to the takeoff location. The circles of photo waypoints are visualized in Figure 33.

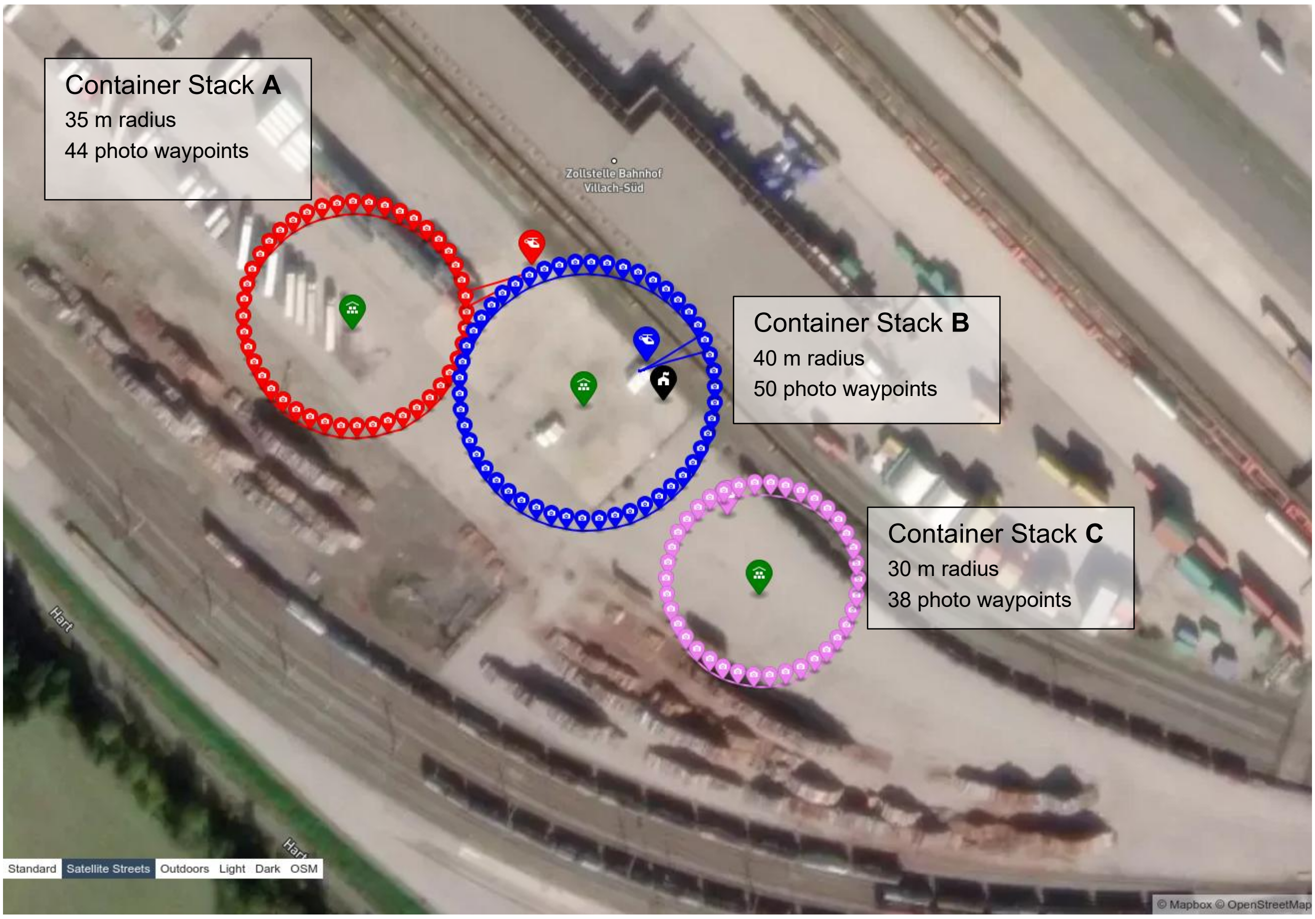


Figure 33. Autonomous waypoint mission using colors to show the drone to container allocation (Note: The background is a generic satellite image and does not reflect the actual state of the logistics center during the experiment)

The coordination algorithm successfully resolved conflicting task allocation of the central container stack. Both the red and the blue drone selected it as the closest container stack. However, the blue drone was given priority because it was closer and thus had lower cost in task allocation.

Image processing and analysis

Shipping container IDs follow the ISO 6346 standard and consist of four key elements: a three-letter owner code, a one-letter equipment category identifier (usually “U” for freight containers), a six-digit serial number, and a check digit used for validation. These IDs are critical for tracking containers throughout global logistics networks. In this demonstration, the goal was to detect these IDs accurately from drone imagery using OCR technology.

To visually illustrate the performance of the detection pipeline, Figure 34 and Figure 35 show aerial images of container stacks A and B, respectively. Overlaid on these images are the detection boxes highlighting the identified container IDs as extracted by the OCR system.

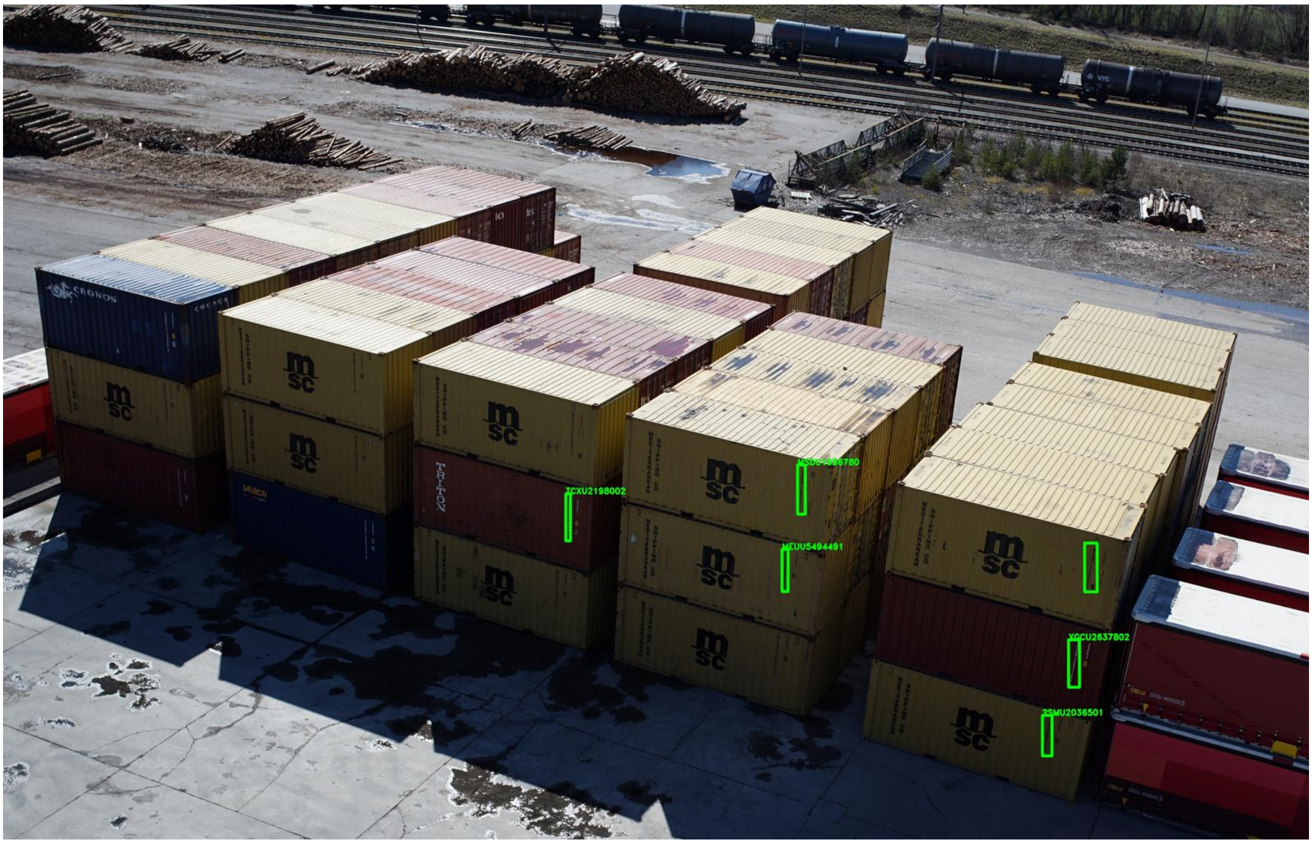


Figure 34. Aerial image of container stack A with container ID detection boxes

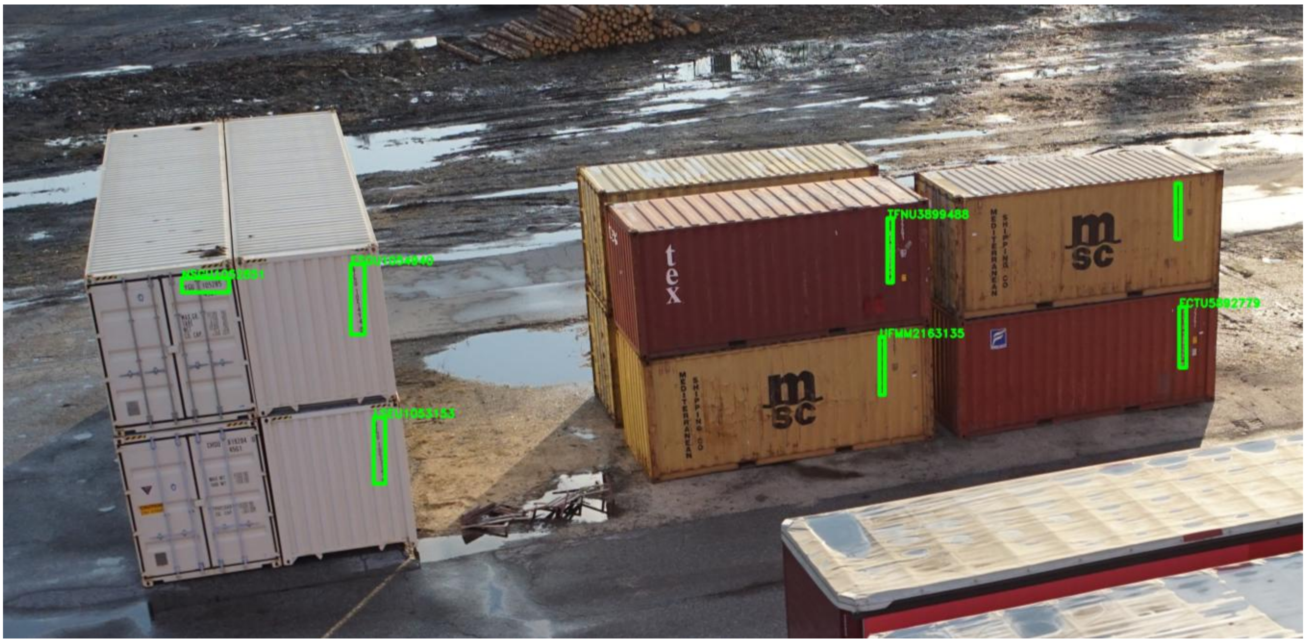


Figure 35. Aerial image of container stack B with container ID detection boxes

After processing the imagery, the data fusion and post-processing step aggregated the OCR outputs and assigned geolocations to the detected container IDs. Table 5 summarizes the outcome of the container ID matching and localization process for the first flight.

Table 5. Matching of container IDs and localization of first flight

Best Match	Ground Truth ID	Levenshtein Score	Longitude	Latitude	Ground Truth Stack	Occurrences
CGSU8192040	CHSU8192040	1	13.793430	46.564319	A	1
ESGU1052850	LYGU1052850	2	13.793434	46.564314	A	2
FCIU5892229	FCIU5892229	0	13.793447	46.564320	A	2
LSGU1053158	LYGU1053158	1	13.793437	46.564312	A	3
LSGU1054940	LYGU1054940	1	13.793438	46.564313	A	1
MEDU2163186	MEDU2163186	0	13.793443	46.564308	A	1
MEDU2701147	MEDU2701147	0	13.793446	46.564311	A	1
MSCU6469619	MSCU6469619	0	13.793431	46.564320	A	4
TFNU3899488	TEMU3899488	2	13.793439	46.564312	A	1

Table 5 illustrates the mapping achieved through our data fusion process. For each ground truth container ID, the best-matching OCR result was selected based on the Levenshtein distance, with a maximum allowed threshold of three. When the same best match was detected in multiple images, the corresponding GPS coordinates were averaged to yield a final, reliable location estimate. Most matches achieved a Levenshtein score of 0–2, indicating high OCR accuracy even under suboptimal imaging conditions. For example, container ID MSCU6469619 was consistently detected across four images, thereby increasing the confidence in both the recognition and the assigned location. Even minor deviations in characters, such as the change from CHSU to CGSU, were successfully reconciled through fuzzy matching.

However, one container, MEDU2930879, located on container stack A, was not detected properly during the experiment. This was likely due to a challenging imaging condition such as occlusion or severe damage to the container ID. Despite this exception, the overall accuracy of the OCR and geospatial fusion process remains high. Furthermore, no containers from stack B or artificial container IDs were incorrectly matched to stack A, ensuring that the system reliably differentiated between the two stacks. It is also worth noting that an additional 60 randomly generated container IDs were included in the ground truth dataset to test the system's ability to avoid false positives, and none of these randomly generated IDs were mistakenly matched to containers on stack A.

For this flight, the system achieved an overall detection accuracy of 90%, based on the successful matching of all containers except for the one failed detection of MEDU2930879. Figure 36 shows on overview of the container yard, with the container IDs successfully detected and matched to their respective locations, during the first drone flight. Each container’s position is marked.

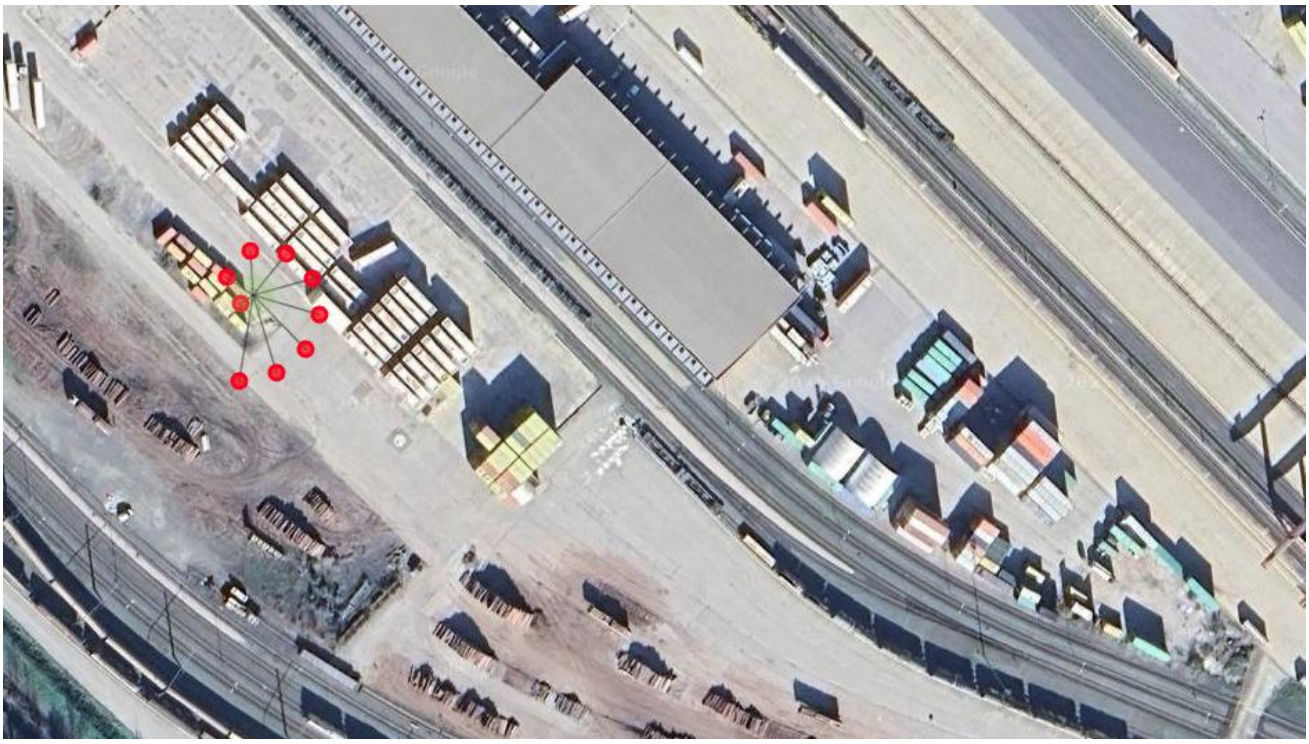


Figure 36. Overview image of the container yard with matched container IDs from the first flight. The red dots, representing GPS locations, are fanned out due to overlapping coordinates where the matching lines intersect. Note: The first flight was a test flight and only stack A was monitored

For the second and third flights, we conducted a similar analysis, yielding the following results visualized in Figure 37. The results of the three flights are summarized in Table 6.

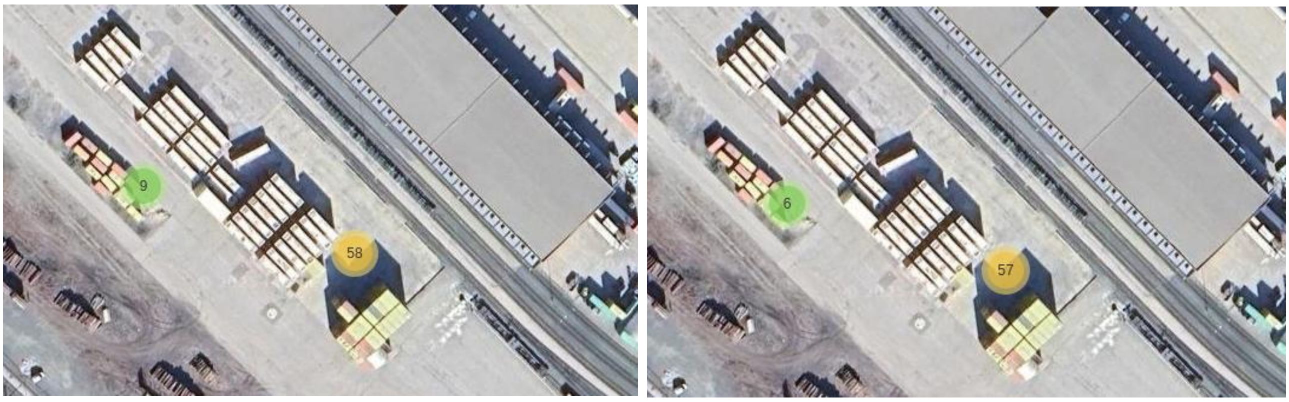


Figure 37. Container ID detection and geolocation results from the second (right) and third (left) flights for stack A (green) and stack B (yellow)

Table 6. Summary of detection results for the three flights

Flight number	**Stack A correct/total**	**Stack A accuracy**	**Stack B correct/total**	**Stack B accuracy**	**Overall correct/total**	**Overall accuracy**
#1	9/10	90%	-	-	9/10	90%
#2	9/10	90%	58/73	79%	67/83	81%
#3	6/10	60%	57/73	78%	63/83	76%

Upload and Processing Latency:

The total latency from image capture to the availability of processed results typically ranged between 6 to 7 minutes. This delay was the result of several sequential steps in the data pipeline:

- Completion of drone flight pattern: The image transfer process from the camera to the onboard Raspberry Pi only began after the drone had completed its predefined circular flight path around the container stack. Since this maneuver itself took several minutes, it represented the most significant contributor to the overall latency. However, this guaranteed that the drones did not have to wait for the image transfer during flight. It has to be noted that this is a limitation of the camera of the demo system that does not support parallel image capture and transmission to the Raspberry Pi.
- Image transfer to onboard System: Once the flight path was completed, images were copied from the drone's camera to the onboard Raspberry Pi. This introduced minor latency, primarily due to limited read/write speeds and the processing capabilities of the Raspberry Pi.

- Upload to remote server: After the images were stored on the Raspberry Pi, they were uploaded to an external SFTP server via a 5G mobile network. Upload duration was influenced by network quality, and given that each image was approximately 10 MB, a flight could generate more than 1 GB of data, contributing a considerable portion to the overall delay. See below for a more in-depth analysis.
- Download for processing: On the server side, the uploaded images were downloaded to the image processing system. File transfer times varied depending on network conditions and storage performance.
- Image processing: Once downloaded, the images were processed using the OCR and data fusion pipeline. This included container ID detection, string matching, GPS localization, and aggregation. Although this phase was relatively fast compared to the previous steps, it still added a small amount of latency before results could be finalized.

Potential improvements: A major reduction in latency could be achieved by initiating the image transfer from the camera to the Raspberry Pi incrementally starting after just a few images have been captured rather than waiting until the entire flight path is completed. This would enable parallelization of data transfer and image capture, significantly shortening the overall processing time.

Limitations

Several limitations affected the accuracy and performance of the container ID detection system:

1. Limited and Homogeneous Training Data: The training dataset for both the object detection and OCR models was largely constrained to imagery collected from two prior drone flights, along with a small selection of publicly available images sourced from the internet. As a result, the dataset lacked sufficient diversity in key aspects such as container types, character variations (particularly alphabetic characters), and environmental conditions. This narrow scope likely limited the models’ ability to generalize across different scenarios, reducing detection accuracy in more complex or varied real-world settings. Expanding the dataset to include a broader range of containers and capture conditions would be essential to improve model robustness.
2. Lighting conditions: Variability in lighting conditions across the different flights significantly affected the clarity of the captured images and, consequently, the performance of the OCR model. The first flight took place under partly cloudy skies, while the second and third flights were conducted in sunny conditions. Changes in sunlight, shadows from nearby containers, and uneven illumination often obscured parts of the container IDs, leading to detection errors or incomplete matches. Container IDs that were evenly lit – without strong shadows or glare – were generally recognized more accurately by the OCR model. Ensuring more consistent and uniform lighting during image capture, either through optimal timing or adaptive camera settings, could help improve overall detection accuracy.
3. Occlusion and viewing angle limitations: The detection and recognition of container IDs were frequently impacted by partial occlusions caused by adjacent containers. In addition, the

viewing angle of the camera played a critical role: when images were captured at steep angles, the OCR model struggled to accurately interpret the alphanumeric characters, often leading to incorrect or incomplete readings. These factors – both occlusion and non-optimal perspective – contributed to misreads or failed detections, thereby reducing the overall accuracy of the container identification process.

Potential Improvements

To improve the detection rate and overall performance of the system in the future, the following steps can be considered:

1. Optimizing flight path: Refining the drone’s flight path can significantly improve both image quality and detection reliability. By adjusting parameters such as altitude, flight angle, and camera orientation, the system can reduce the likelihood of occlusions and minimize steep viewing angles that hinder OCR performance. An optimized path would ensure better coverage of the container area, capture more direct views of container IDs, and reduce overlap and blind spots – ultimately increasing the likelihood of detecting complete and legible IDs.
2. Expanding and diversifying training data: A more diverse and expansive dataset, incorporating images from different container yards, weather conditions, times of day, and varied container types, would significantly enhance the model's ability to generalize and perform well in different real-world scenarios. More training data would also allow the models to better handle edge cases such as dirt on container IDs, unusual lighting conditions, and extreme occlusions. Additionally, using synthetic data augmentation techniques could further diversify the dataset, improve robustness and reduce the likelihood of misdetections.

Network Bandwidth

The images captured by the cameras on the drones produced a significant amount of network load during the mission. In the presented setup, all files were copied over the cellular network once the drones finished the container inspection. This setup was a simplification for the demonstration and comes with a few limitations. In a real-word deployment, real-time image transfer would likely be essential for effective operation. If real-time processing was needed during the mission (e.g., to adapt drone behavior dynamically, or to trigger detailed inspection of specific scenes), a significantly higher network throughput would be required throughout the flight.

To better understand the implications for network performance in such a real-time scenario, we analyzed the data rates required by the imaging system. To quantify the data rate, we computed the amount of data generated by the camera sensors throughout the flights from the image sizes and the times between subsequent images. This means, if the available network bandwidth would drop below the calculated data rate of the sensors, the images would queue up and could not be processed in real time.

Figure 38 and Figure 39 show the measured camera sensor data rates based on the images collected during the demonstration flights at LCA for 5 m spacing and 2 m spacing between capture locations, respectively. The average data rate per drone varies roughly between 20 Mbit/s and 30 Mbit/s while images were captured approximately every 3-4 seconds. It can be seen, that more frequent image capturing (at a distance of 2 m) produces higher data rates compared to less frequent image capturing (at a distance of 5 m).

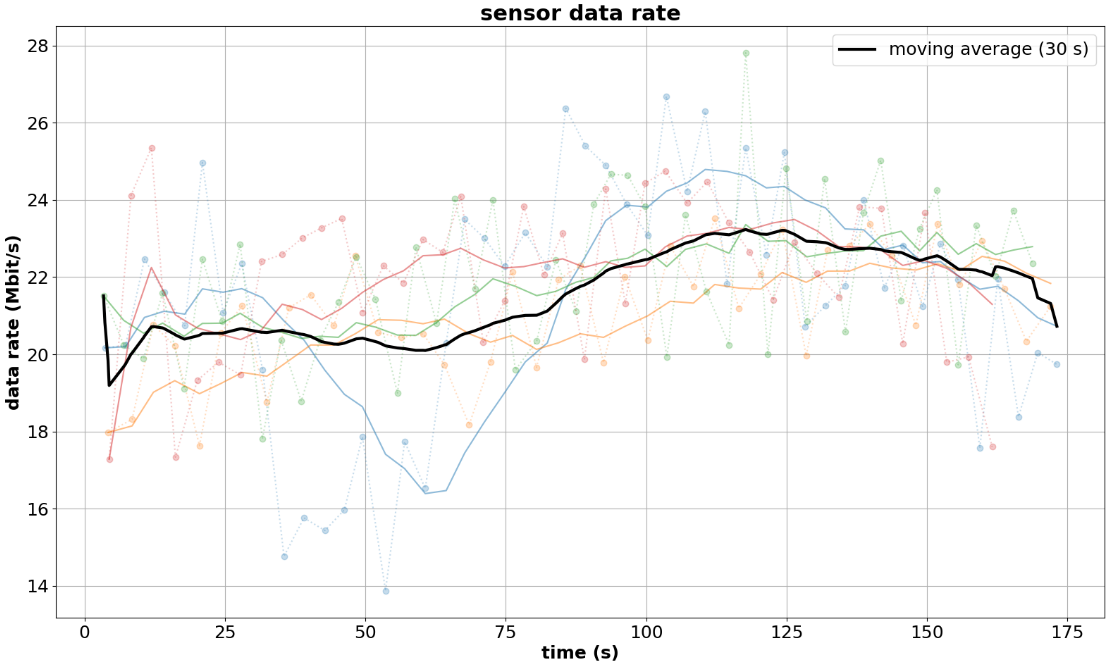


Figure 38. Measured data rate of the drone camera with 5 m spacing for different flights (colored) and averaged over all flights (black)

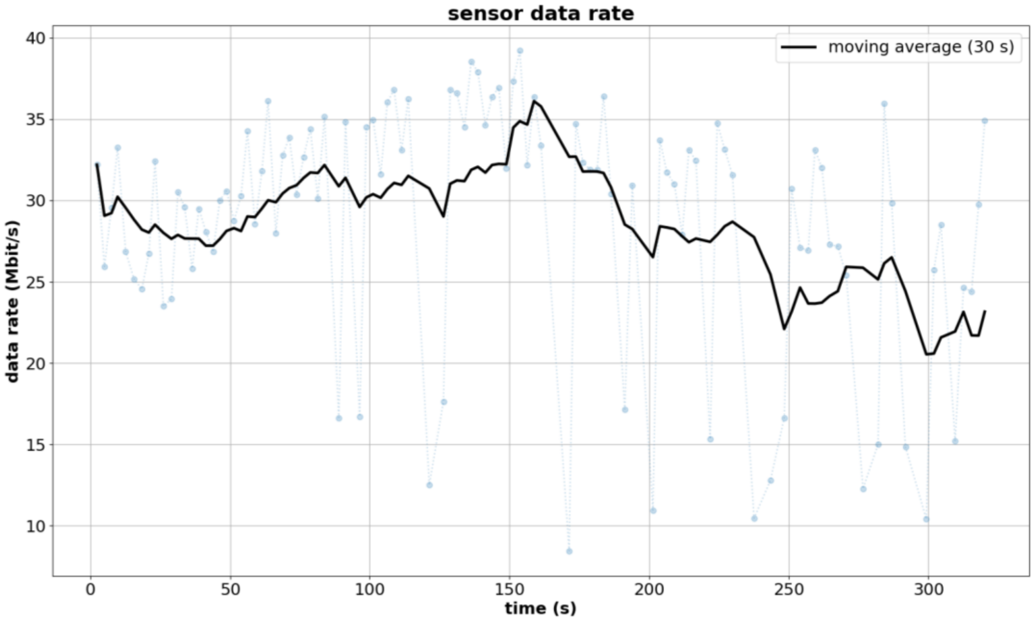


Figure 39. Measured data rate of the drone camera with 2 m spacing for different flights (colored) and averaged over all flights (black)

More generally, the required data rate for real-time transfer can be expressed as:

$$r = {}^{S}/_{\Delta t}$$

where S is the size of a single image and Δt is the time interval between consecutive captures by a single drone. For a swarm of drones D, the total network load can be expressed as

$$L = \sum_{i \in D} r_i$$

where r_i is the individual data rate of drone i. The resulting network load for different swarm sizes and capture intervals is visualized in Figure 40, illustrating the scaling behavior of bandwidth requirements under various operational conditions. Note that these values represent idealized data rates based on image size and timing alone, and do not account for additional overhead introduced by network protocols, encryption, retransmissions, or other system-level factors.

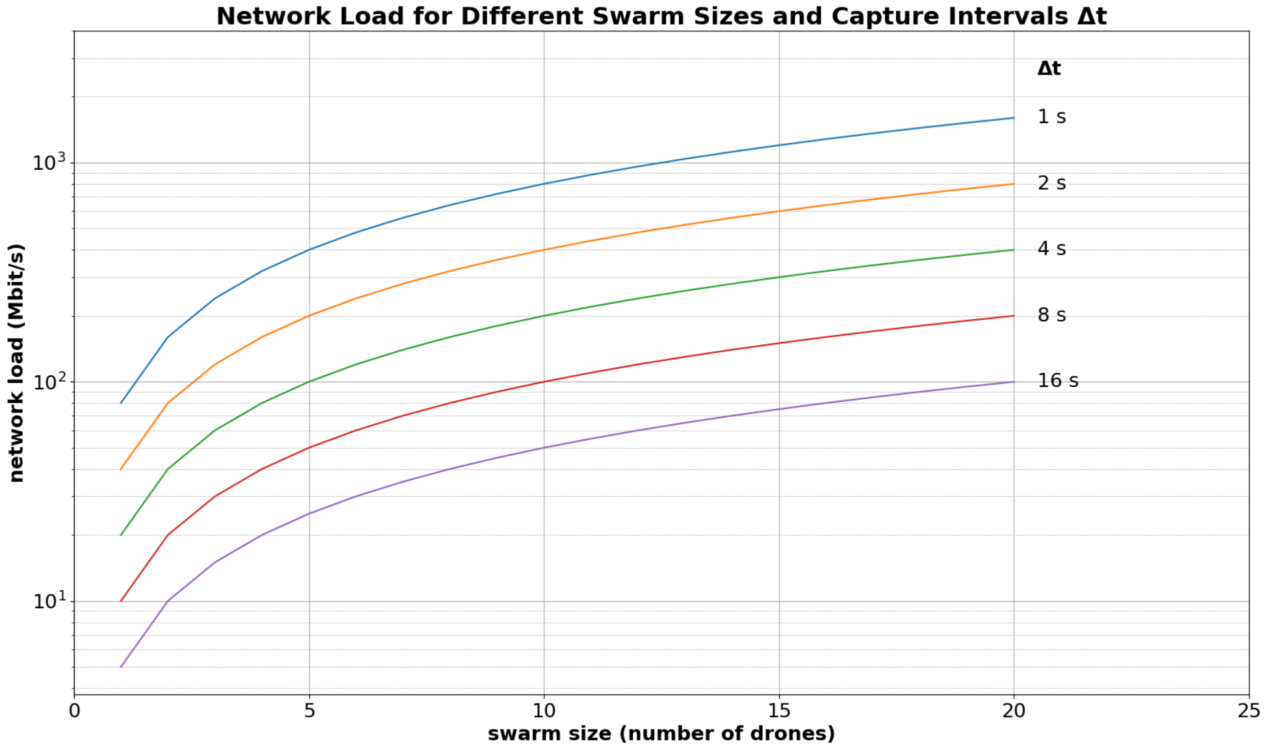


Figure 40. Network load for different swarm sizes and capture intervals

Conclusions and Discussion

This use case has successfully demonstrated the feasibility and effectiveness of using a swarm of autonomous drones to support container localization in a real-world logistics environment. By combining distributed drone coordination, high-resolution aerial imaging, cellular and mesh connectivity, and cloud-based OCR processing, the system achieved high accuracy of up to 90% in container identification and geolocation across multiple test flights. Key contributions include the reliable execution of autonomous missions, where the drone swarm effectively allocated tasks and navigated the environment with minimal operator intervention. The system featured a robust image processing pipeline, leveraging a three-stage OCR architecture and data fusion techniques to accurately extract ISO 6346-compliant container IDs under diverse real-world conditions. It also demonstrated efficient handling of large image datasets.

Despite the successful demonstration, several challenges were identified. Network bottlenecks during ROS node discovery limited the scalability of the mesh network. Additionally, environmental factors such as lighting conditions, occlusions, and camera viewing angles affected the accuracy of OCR results. Finally, real-time data transfer at high image capture rates placed substantial demands on available bandwidth, highlighting the need for more efficient data handling and communication strategies.

While demonstration was conducted using current-generation technologies such as Wi-Fi mesh and 5G cellular connectivity, it already incorporated several concepts central to future 6G systems. The real-time offloading of high-resolution image data via cellular networks, combined with local swarm coordination over a mesh network, offered an early showcase of network-assisted autonomy, a concept expected to be central in 6G use cases. Moreover, the limitations observed in bandwidth, network discovery overhead, and the scalability of ad-hoc mesh networks provide valuable insights into communication bottlenecks that 6G must address. In this sense, the field demonstration served not only as a validation of the current system design but also as an early exploratory step toward the types of networked robotic applications that 6G aims to support. The data traffic patterns, latency measurements, and control architecture explored here can inform practical requirements for future 6G protocols and infrastructure planning.

6.2 Use case 2: UAV Swarm to Support Autonomous Mobility and Infrastructure in Rural Areas

In this use case a mixed swarm of aerial drones (UAVs) and ground rovers (UGVs) operate in a logistics center. The rovers emulate the reach stackers that move the containers that have previously been identified and located the drone swarm of use case 1. The drones support the rovers by providing reliable network connectivity in the highly obstructed environment.

6.2.1 Network Technologies

This use case uses both the 5G cellular network and local Wi-Fi mesh networking. However, the focus of the use case is on the mesh Wi-Fi to highlighting its flexibility. In the experiments, three Wallys DR4029 wireless access points were used as mesh devices. They support Wi-Fi 5 (802.11ac) and were operated in the 5 GHz ISM band.

6.2.2 System Components and Architecture

The use case is implemented by three main components shown in Figure 41:

- Relay drone (UAV)
- Autonomous UGV
- Ground control station (GCS)

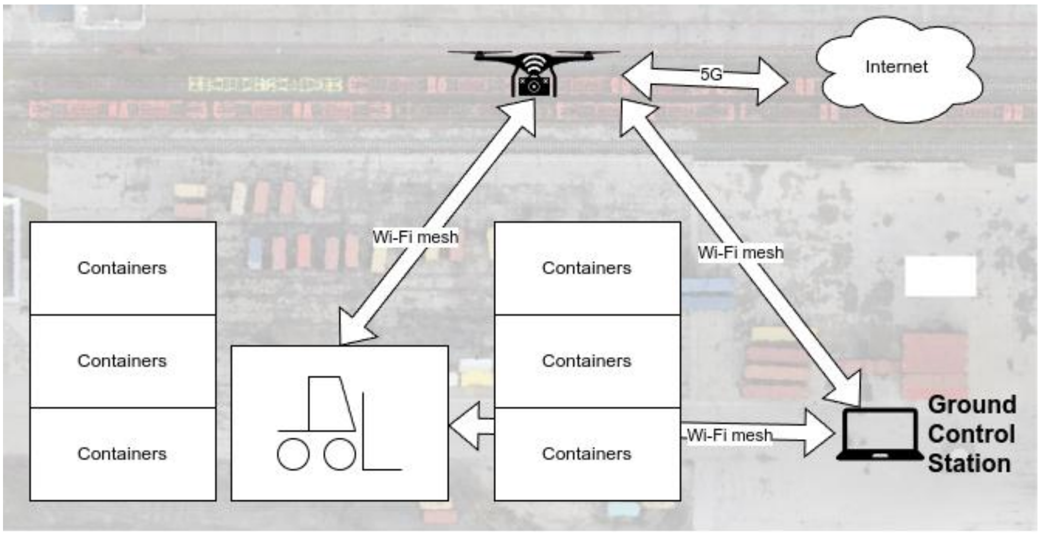


Figure 41. High-level architecture of the autonomous mobility use case

Relay drone

The relay drone is a twinFOLD SCIENCE drone, similar to the twinFOLD GEO described in Section 3.1 and shown in Figure 42. It is a quadcopter with a maximum takeoff weight of 1900 g and equipped with a Raspberry Pi for computation, a Quectel 5G modem, and a Wi-Fi mesh router for communication. The internal architecture is visualized in Figure 43.

The role of the relay drone is to provide enhanced network connectivity to the ground vehicle. For this purpose, it is connected to the ground vehicle and the ground control station via a mesh Wi-Fi and to the Internet via 5G. It relays payload data between the rover and the GCS (or the Internet) while receiving mission commands from the GCS.

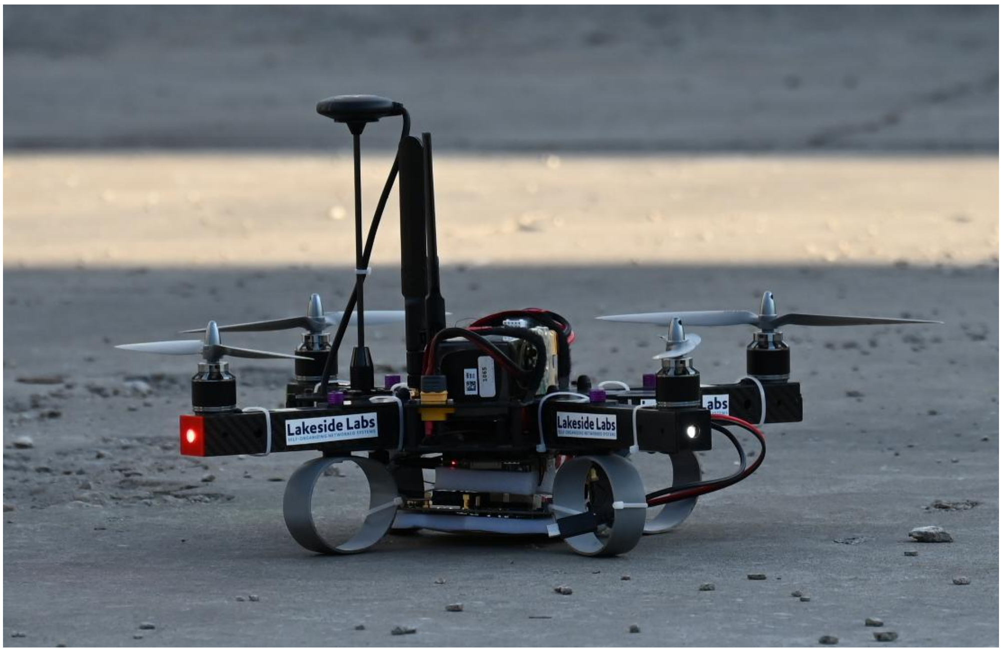


Figure 42. twinFOLD SCIENCE quadcopter

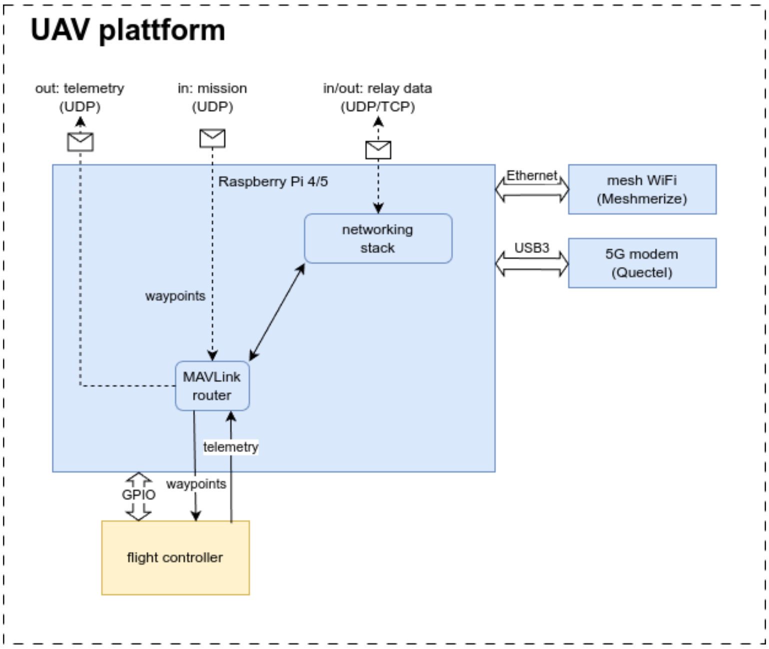


Figure 43. Architecture of the relay drone.

Unmanned ground vehicle

The ground vehicle is a custom, electronically driven rover, called SPIDER, shown in Figure 44. It carries computation and communication modules and can be controlled either through the on-board computer or remotely by an operator. For external control it sends telemetry data as well as a live video stream to the GCS. The internal architecture is visualized in Figure 45. The live video stream is captured onboard using a USB video source (via v4l2src). The video is encoded in real time using H.264 (30 frames per second, with a resolution of 1280x720 pixels and a target bit rate of 5 Mbit/s) and transmitted over UDP using RTP streaming. A GStreamer-based pipeline handles this process, including video capture, encoding, RTP packetization, and UDP transmission. On the GCS side, a corresponding GStreamer pipeline receives, depacketizes, decodes, and displays the live video, while optionally recording the stream. This architecture enables low-latency video transmission for applications such as to support operator situational awareness and remote control.

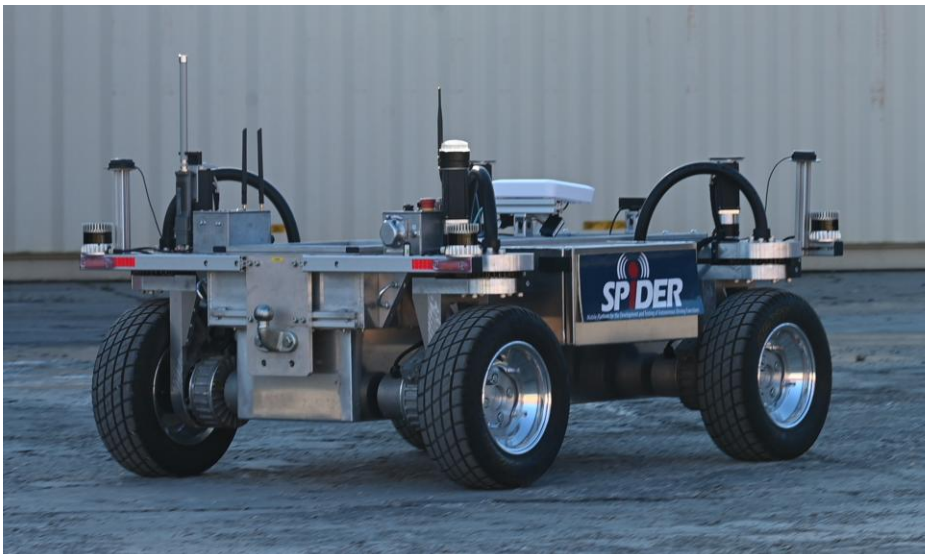


Figure 44. The SPIDER unmanned ground vehicle with the mounted mesh device (white box in the middle)

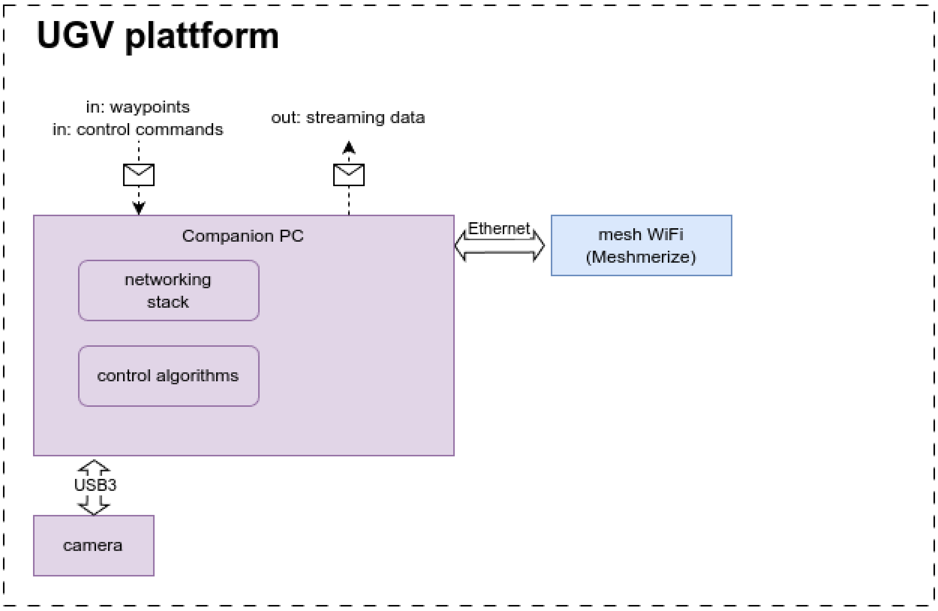


Figure 45. Architecture of the unmanned ground vehicle

Ground control station

The ground control station (GCS) is responsible for mission execution and monitoring.

The GCS is responsible for mission execution and monitoring. It is implemented on a laptop running Ubuntu Linux (see Figure 46). It is connected to the drone and the rover over mesh Wi-Fi. It is used to start the mission and receive telemetry (using the COTS software QGroundControl) and visualize the video stream received from the rover (using GStreamer).

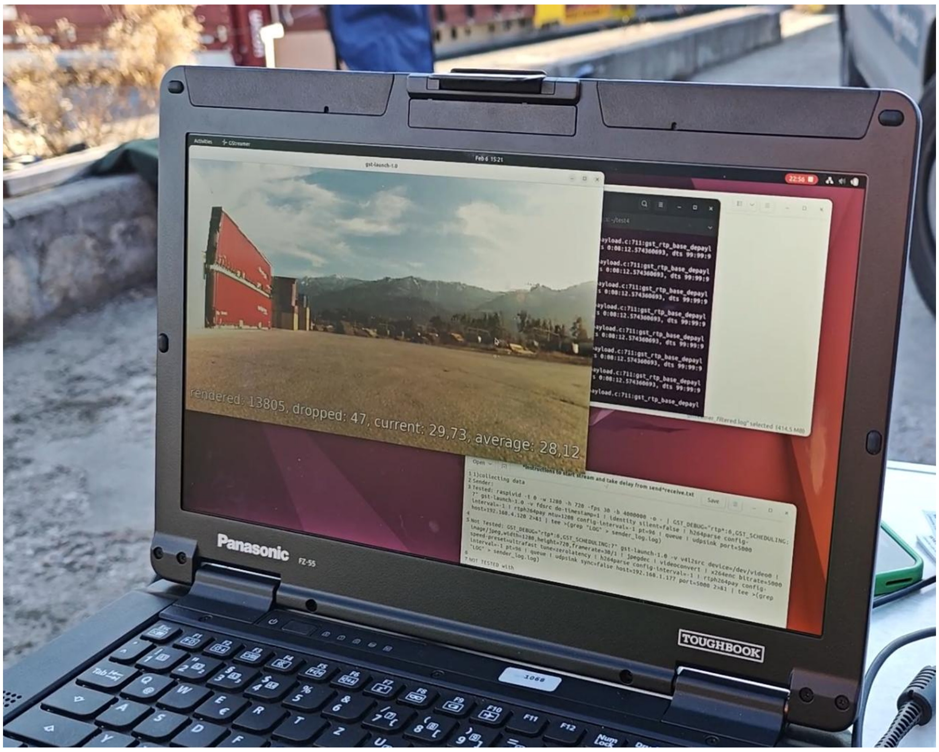

gst-launch-1.0
rendered: 13805, dropped: 47, current: 29,73, average: 28,12
Panasonic
FZ-55
TOUGHBOOK

Figure 46. Ground control station laptop showing the live video stream during the demo

6.2.3 Experiment Setup

The demonstration was performed on the container terminal of the Logistik Center Austria Süd in Fürnitz, Austria. The rover first driven manually driven on a path between multiple container stacks. This path was recorded and repeated multiple times (see blue line in Figure 47).

For the drone was flown at different altitudes to determine the best altitude to provide line of sight to both the rover and the GCS. The final selected altitude was 30 m. The drone was flying at a speed of 1.2 m/s to a list of waypoints extracted from the rover path (see green markers in Figure 47).

For the experiments with aerial relay, the drone was flying above the rover

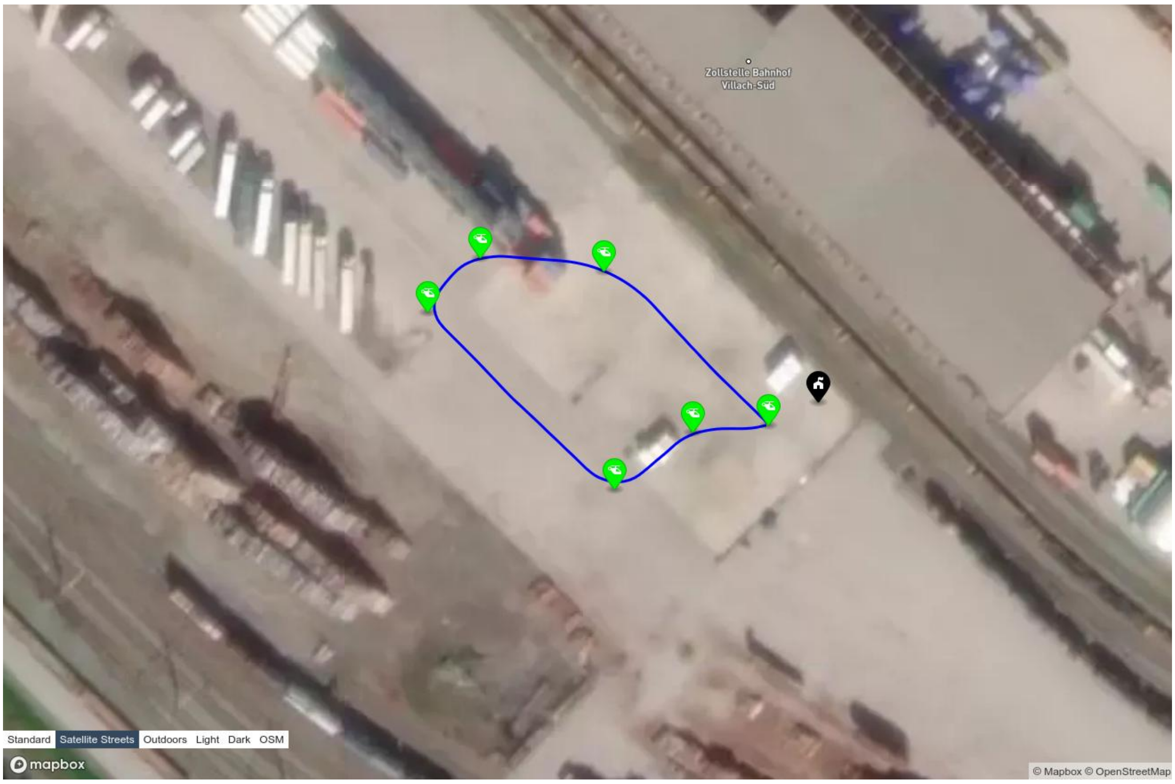


Figure 47. Setup of the use case 2 demonstration with rover path (blue), drone waypoints (green), and ground control station (black)

6.2.4 Evaluation Results

For evaluation, we performed several repetitions where the rover followed the path while streaming a live video to the GCS. On the GCS we recorded:

- **Latency**: The end-to-end delay of the video data in seconds. In case of communication disruptions, no data samples were recorded.
- **Link metric**: The quality of the communication link as distance estimation by the Meshmerize protocol.
- **Distance**: The distance between the GCS and the rover in meters based on its GNSS receivers.

We performed two types of experiment:

- **Ground communication only**: The rover directly sends the video stream to the GCS using the Wi-Fi mesh.
- **With aerial relay**: The rover sends the video stream to the GCS using the Wi-Fi mesh. The drone acts as aerial relay in case the direct link degrades. The drone was flying above the rover while it moved along the pre-defined path.

Figure 48 shows both rover and drone at the beginning of one experimental run.

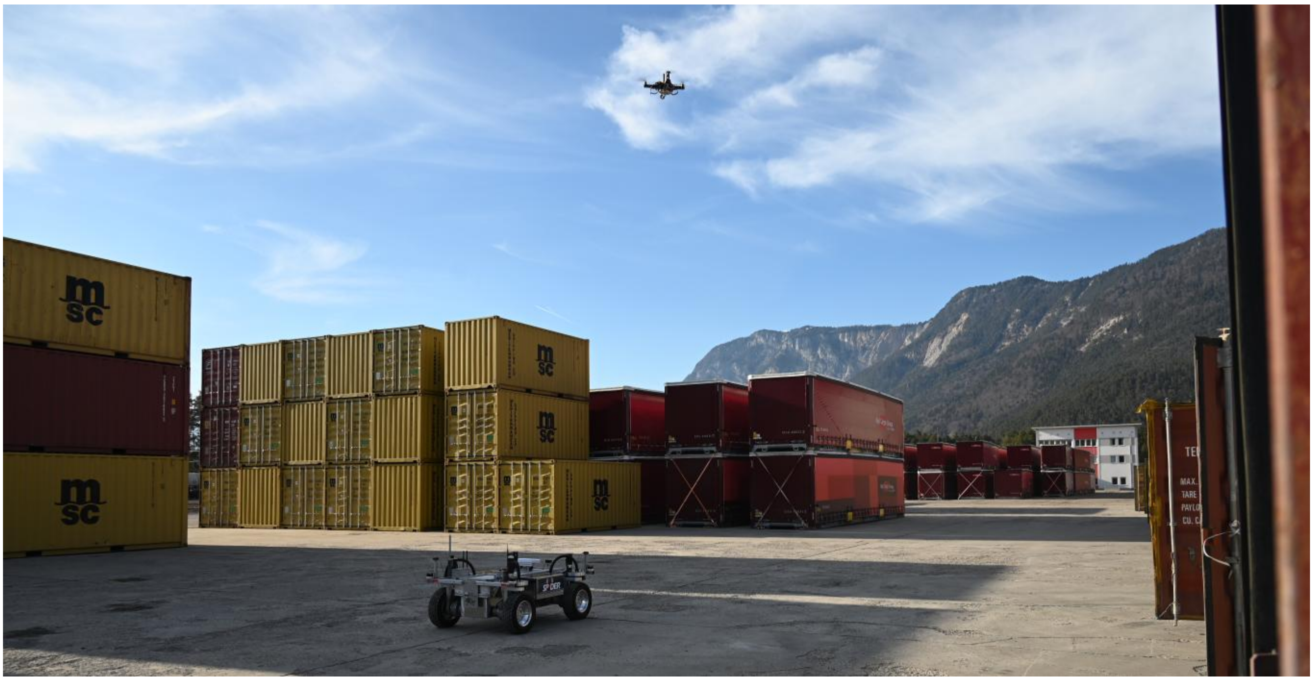


Figure 48. Rover and drone during the demonstration of the use case 2 demonstration

Using a local time server, we synchronized the different system components to match the measurements and aggregate the data. The results of the individual runs and their average are shown in Figure 49 and Figure 50. The two figures show end-to-end latency, estimated link performance, and distance between rover and GCS averaged over several runs. The results in Figure 49 show the setup were only direct communication between rover and GCS was available, while Figure 50 shows the results of the aerial relay supported as multi-hop link when required (green shading).

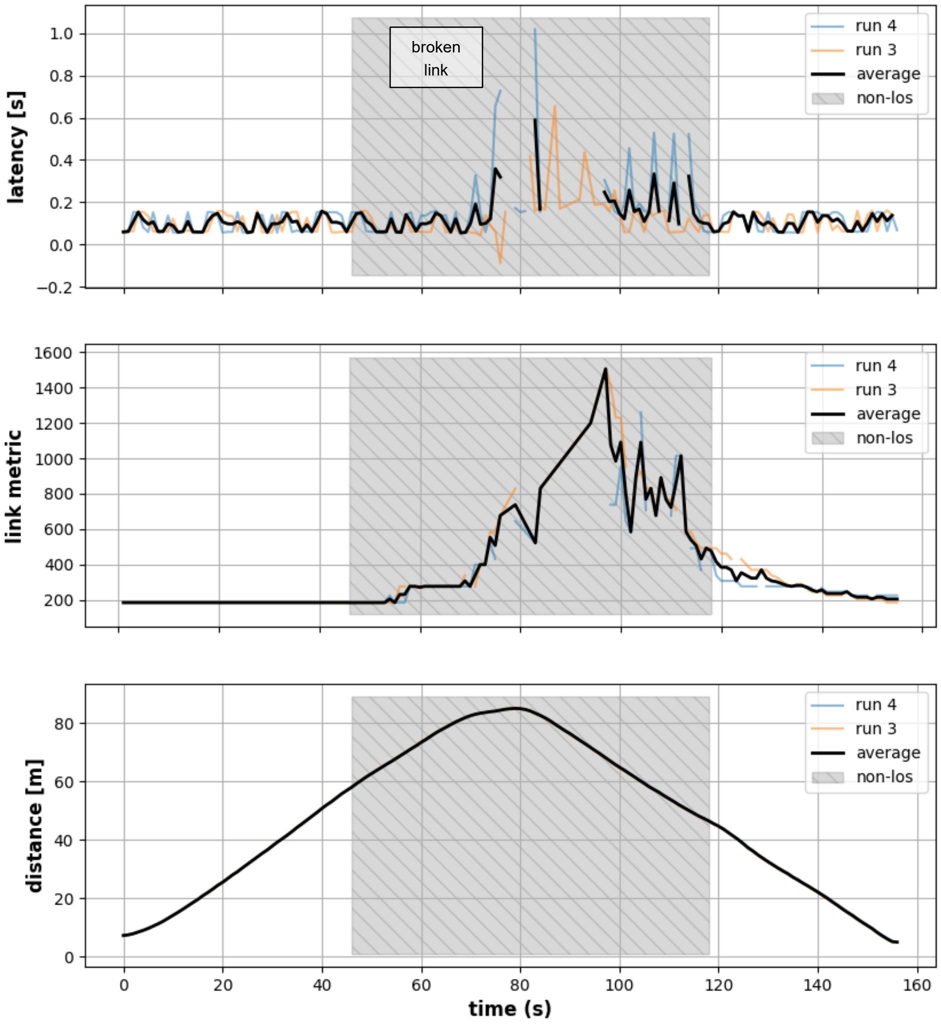


Figure 49. Network measurement results of the use case 2 demonstration with ground communication only

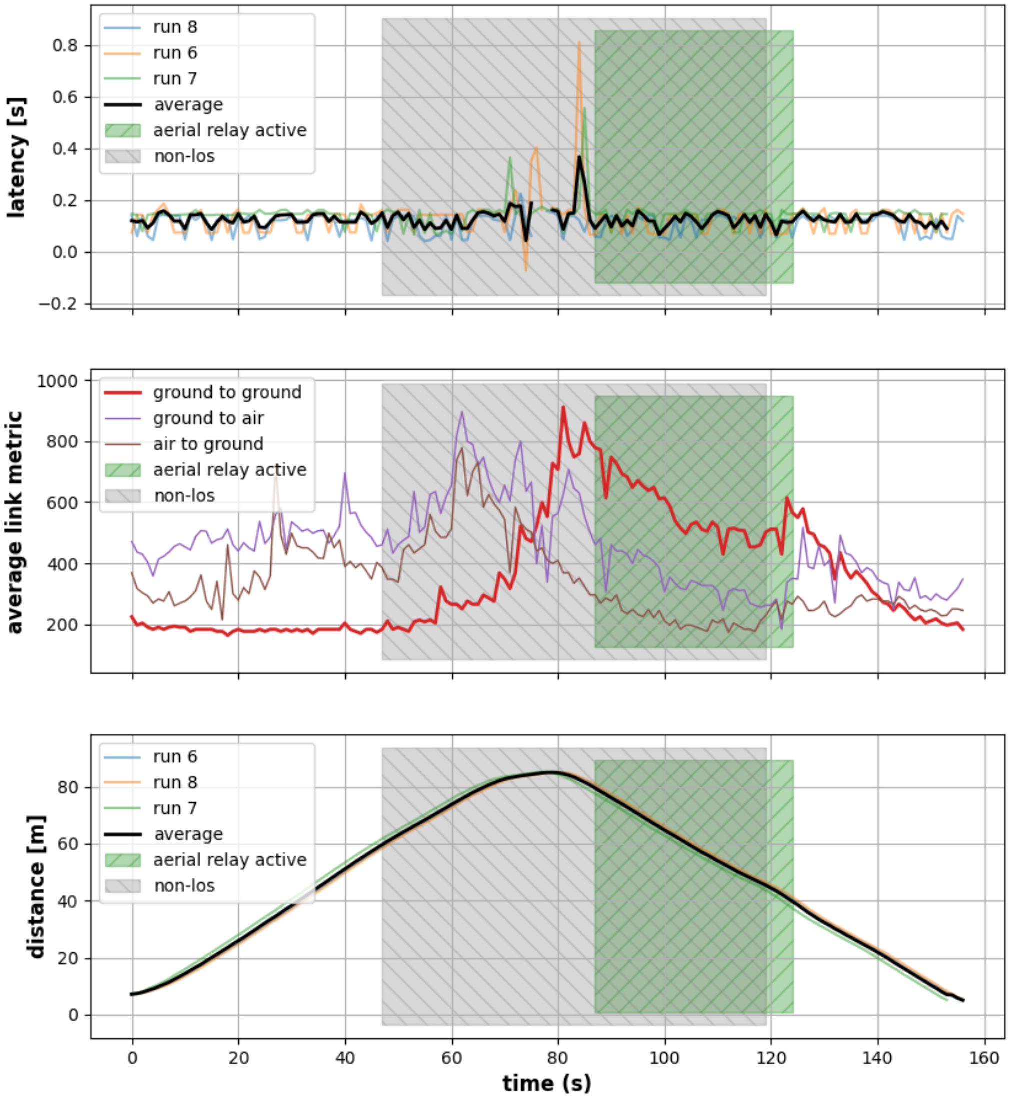


Figure 50. Network measurement results of the use case 2 demonstration with aerial relay

The part of the trajectory where containers occluded the line-of-sight between the rover and the GCS are shaded in gray. It can be seen that the link quality deteriorates when the distance increases and there is no line of sight. With ground communication only (Figure 49), the link breaks for several seconds until it reestablishes when the rover returns to the GCS. This happened in every experiment. With the drone as aerial relay, it can be seen that the link switches to multi-hop during that time (green shading in the right graphs in Figure 50). There is only a short time with increased latency which normalizes quickly when the new multi-hop link is established. Overall, the latency rarely surpasses 200 ms when the drone relay is used.

Another interesting observation from the middle graph in Figure 50 is the estimated link performance averaged over the runs that included the drone. While the ground-to-ground link performs better if there is line of sight, the link to and from the aerial relay performs much better in case of obstacles.

6.3 Discussion

While the systems described in these use cases rely on current-generation technologies such as Wi-Fi and 5G, they demonstrate functional aspects that align with anticipated features of future 6G networks. In Use Case 2, the deployment of a drone as a mobile relay node highlights the relevance of multi-hop communication to maintain network connectivity in obstructed environments – an approach that future 6G systems aim to support more seamlessly and autonomously. The manual adjustment of drone positioning in this context reflects an early form of topological adaptation, a capability expected to be automated in next-generation networks. In Use Case 1, the autonomous drone swarm demonstrates key principles of decentralized coordination, local task allocation, and low-latency mesh-based collaboration, which align with 6G's emphasis on self-organizing, edge-intelligent networks. Furthermore, the offloading of image data over a cellular connection to a remote processing server illustrates early instances of split computing and network-assisted sensing – both considered foundational components in the 6G vision. Taken together, these systems not only explore early implementations of 6G-relevant concepts but can also serve as practical input for defining technical requirements in future 6G system design, especially in terms of connectivity resilience, distributed intelligence, and integration of communication with sensing and control.

7 Task 5.5: Demonstration sense and avoid mechanisms

Lead: Skysense, **Contributors:** Lakeside Labs, Airbus, twins.

7.1 Motivation

As the use of drones becomes increasingly widespread across industries such as logistics, agriculture, surveillance, and emergency response, the airspace is becoming more crowded and complex. This congestion raises significant safety concerns, especially in scenarios involving autonomous or semi-autonomous drones operating in shared environments. One critical issue is the risk of mid-air collisions caused by uncoordinated drone activity, particularly from unidentified or intruding drones that may not follow standard flight protocols.

To address this, we demonstrate a system that enhances situational awareness and airspace safety by detecting and tracking intruding drones in real time and communicating their positions to a nearby friendly 6G-SKY drone. This allows the 6G-SKY drone to autonomously adjust their flight path to avoid potential collisions.

7.2 Technologies

To enable real-time tracking of intruding drones and proactive collision avoidance the Skysense drone sensor (as seen in Figure 51), leverages the following integrated technologies:

1. RF Signal Interception and Analysis:

   The core of the Skysense sensor system relies on intercepting the Direct Remote ID (DRI) sent from the drone. By passively scanning the relevant frequency bands and listening for radio communication belonging to the DRI protocol, the system is able to extract telemetry data, most importantly the GPS coordinates of the drone, without interfering with the operation of the drone.

2. RF Signal Conditioning

   Direct Remote ID is sent on 2.4 GHz and 5.8GHz band. Skysense Drone sensor utilizes custom hardware to filter and amplify RF signals on these frequency bands.

3. Data Parsing:

   Using protocol-specific parsers, the intercepted data stream is decoded to identify and isolate drone position information.

4. Real-Time Position Tracking:

Once the positional data is extracted, the system performs continuous tracking of the intruding drone's location. This information is updated in real time and formatted to JSON before being passed on.

5. Communication to Friendly Drone

The system then forwards the intruder's position over MQTT (Message Queuing Telemetry Transport) to the 6G-SKY drone controller system so that evasive actions can be taken.

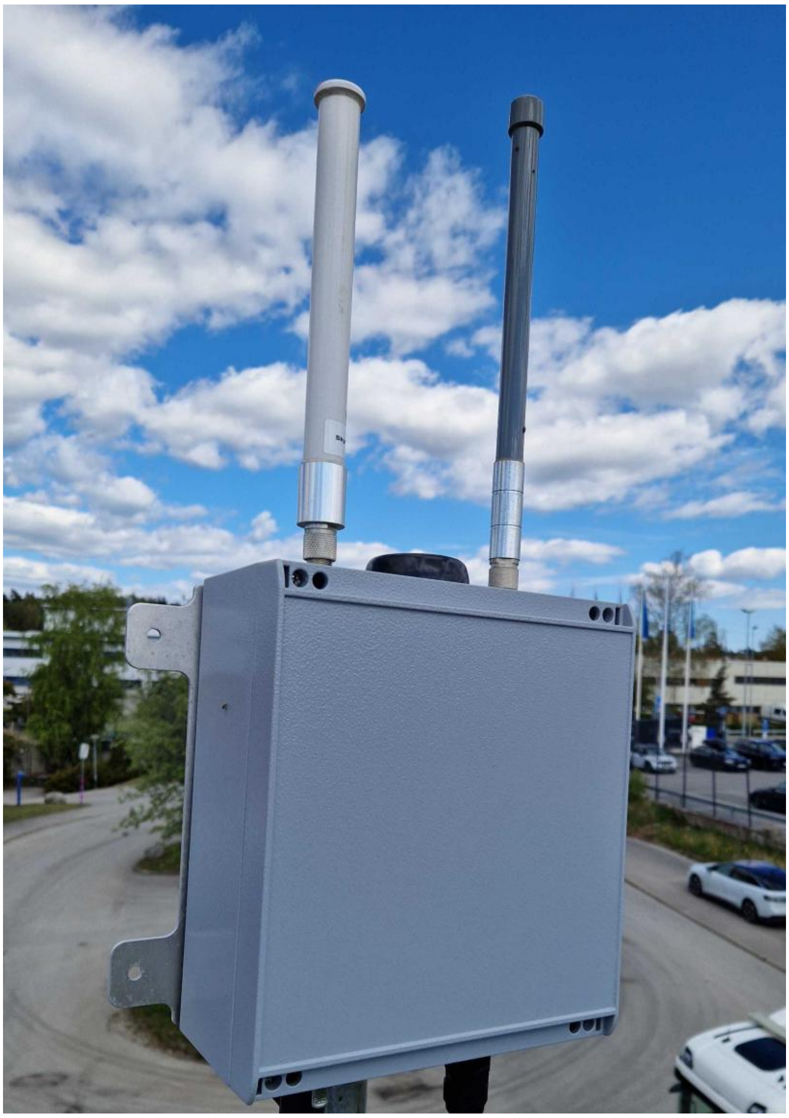


Figure 51. Skysense Drone Sensor (during preliminary tests)

7.3 Use case

A 6G-SKY drone is flying an autonomous mission when an intruding drone is approaching. Without any evasive actions being taken there is a risk of collision. The Skysense drone sensor detects the intruding drone and forwards its position to the 6G-SKY drone which first halts its mission when the intruding drone approaches and then proceeds to land when the intruding drone gets to close.

7.4 Setup

In the image below the setup for this demonstration is illustrated. The intruding drone, a Mavic Air 3 is approaching, it is detected by the Skysense sensor that extracts its position and forwards it to a remote MQTT server. The position is picked up by 6G-ky drone mission command and control which instructs the 6G-SKY drone to avoid the unidentified drone.

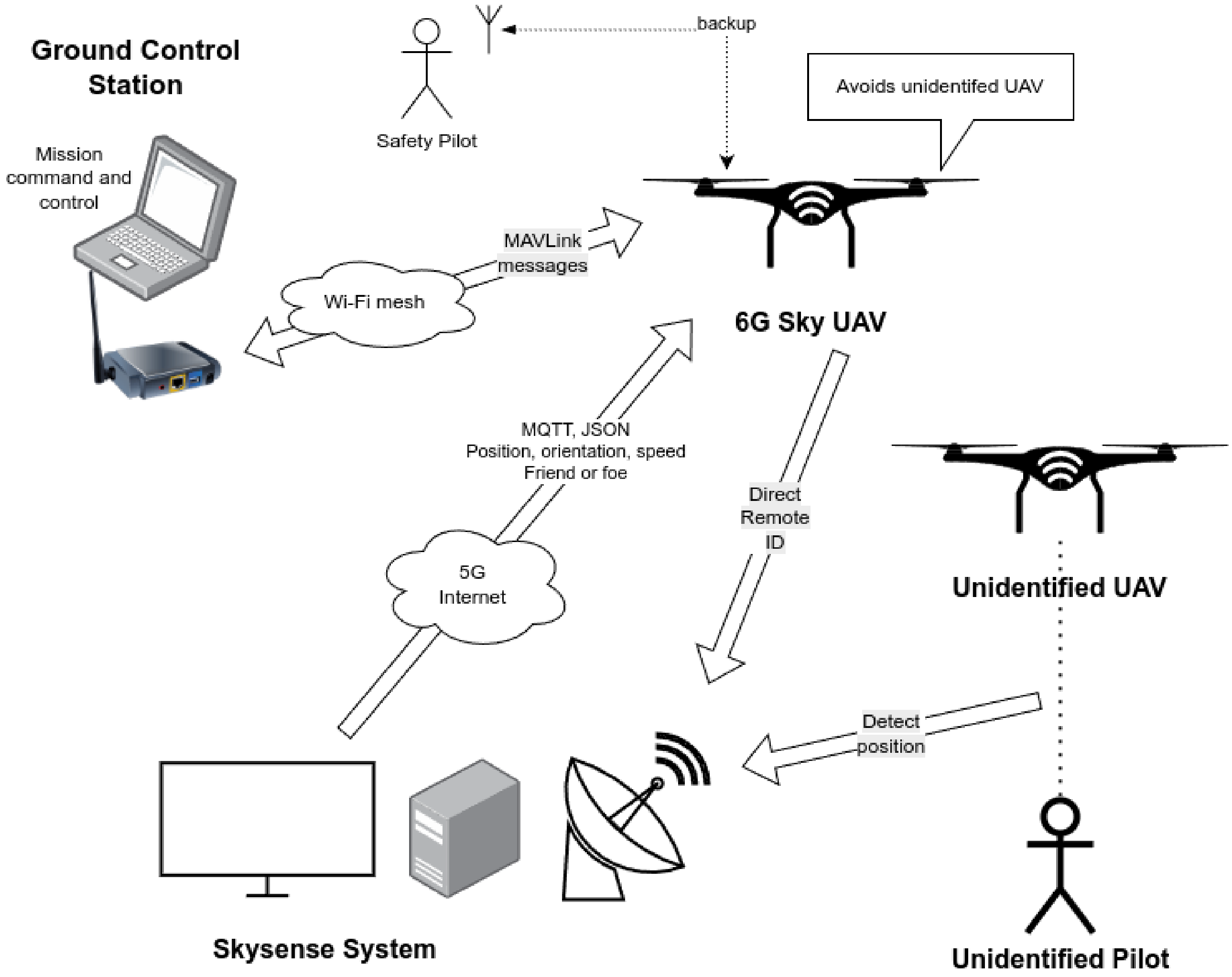


Figure 52. Sense and Avoid demonstration setup; Consisting of a ground drone sensor system (Skysense), a "friendly" 6G-SKY drone (Twins) performing an autonomous mission (LAKE), and an unidentified "intruder" drone

7.5 Demo Mission Description and Results

The demonstration showcases a basic autonomous drone mission focused on monitoring a point of interest. In this scenario, the 6G Sky drone (aka friendly drone) performs a circular flight pattern at a fixed altitude and radius, simulating a typical surveillance or inspection operation. The purpose of this path is to capture the designated area from multiple angles, mimicking real-world tasks such as perimeter monitoring or object observation (e.g., as performed in the logistics monitoring use case described in Section 6.1).

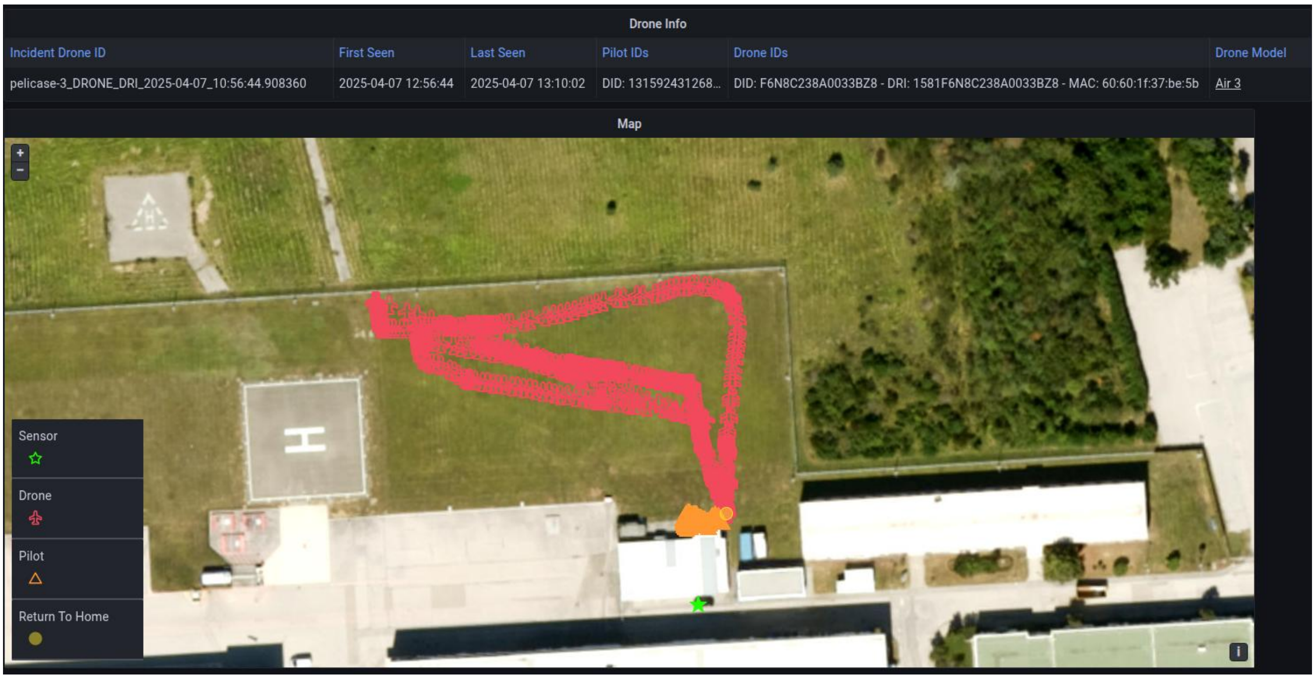
An intruding drone, operated manually, was introduced into the environment to simulate an unexpected aerial presence. To ensure operational safety and demonstrate the sense and avoid system, two concentric safety radii were defined around the friendly drone's position:

- Caution Zone (40 meters): When the intruding drone enters this outer zone, the friendly drone immediately pauses its mission and hovers in place. This allows for a temporary interruption without full mission termination, ensuring the drone remains in a safe and predictable state.
- Critical Zone (20 meters): If the intruding drone breaches this inner zone, the friendly drone triggers an automated landing. This evasive measure is designed to avoid potential collisions or interference at close proximity.

Both safety radii can be configured freely based on operational requirements, risk assessments, or environmental constraints. For this demo, values of 40 meters and 20 meters were chosen to clearly illustrate the system's response behaviors.

The pause and auto-land responses are two examples of evasion mechanisms used to mitigate potential conflicts. These were selected for clarity and simplicity in the demo. In real-world missions, alternative or more advanced strategies could be employed depending on the use case – such as path replanning, altitude changes, or more advanced reactive avoidance maneuvers – based on what is deemed most appropriate for the operational context.

Figure 53 illustrates the demo area including the positions of the intruding drone as captured by the Skysense sensor. Each red plane icon is a recorded position. The 6G-SKY drone was performing its mission near the right side of the fenced area. The intruding drone started its flight near the helipad (and flew back and forth between the helipad and the mission are. The steady stream of positions provided enabled the 6G-SKY drone to successfully evade the intruding drone.



Drone Info

Incident Drone ID	First Seen	Last Seen	Pilot IDs	Drone IDs	Drone Model
pelicase-3_DRONE_DRI_2025-04-07_10:56:44.908360	2025-04-07 12:56:44	2025-04-07 13:10:02	DID: 131592431268...	DID: F6N8C238A0033BZ8 - DRI: 1581F6N8C238A0033BZ8 - MAC: 60:60:1f:37:be:5b	Air 3

Figure 53. Demo area at Airbus OTN helipad and flight trajectory of the intruder drone

At the beginning of the mission, the intruding drone is located at a safe distance from the friendly drone and does not interfere with its operation (see Figure 54. The friendly drone continues its circular monitoring mission without interruption. As the mission progresses, the intruder drone is manually steered toward the flight area of the friendly drone. Approximately 56 seconds into the mission, the intruder drone breaches the caution zone (Figure 54). Upon detecting this, the friendly drone immediately pauses its mission and transitions into a hovering state at its current position. This behavior is designed to reduce the risk of conflict while maintaining readiness to resume the mission if the airspace becomes safe again.

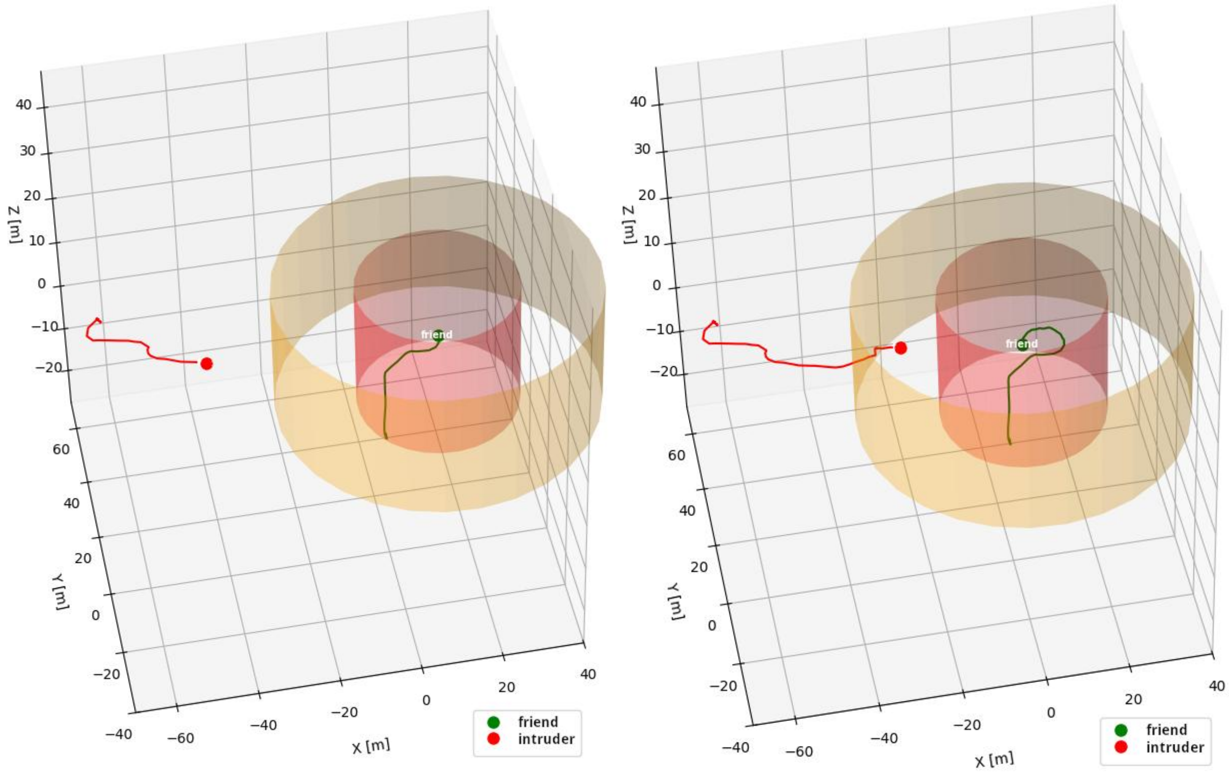


Figure 54. Intruder drone outside of caution zone (left, t=27s) and entering caution zone (right, t=56s)

The intruder drone is then steered away from the friendly drone. After the intruder drone exits the caution zone, the friendly drone resumes the mission (see Figure 55 continuing along the predefined circular trajectory.

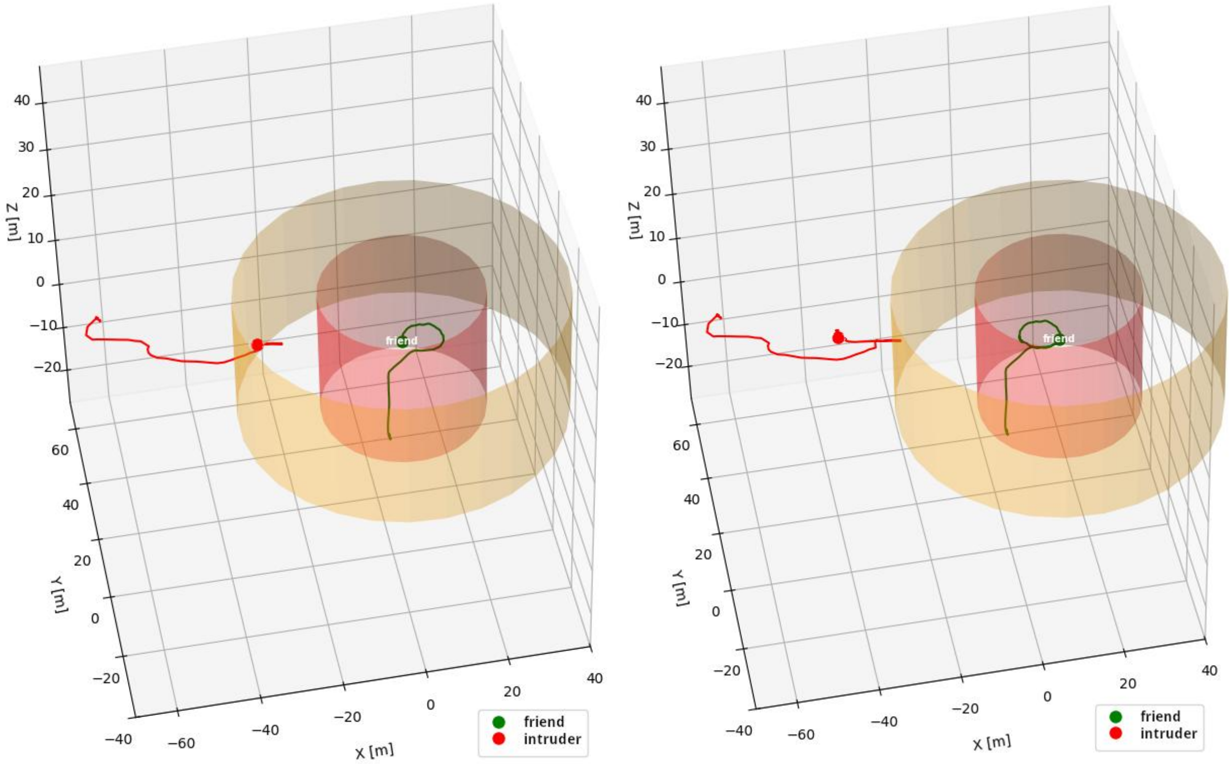


Figure 55. Intruder drone leaving caution zone (left, t=60s), Intruder drone outside caution zone (right, t=78s)

Later in the demonstration, the intruder drone is again directed toward the friendly drone’s operational area. It first enters the caution zone (Figure 56), and then proceeds further into the critical zone which is the inner safety radius established for high-risk proximity (Figure 56. In response to this more serious intrusion, the friendly drone aborts its mission and executes an automatic landing procedure. This action is taken as last-resort evasive maneuver to ensure safety in a scenario where further conflict avoidance is not possible.

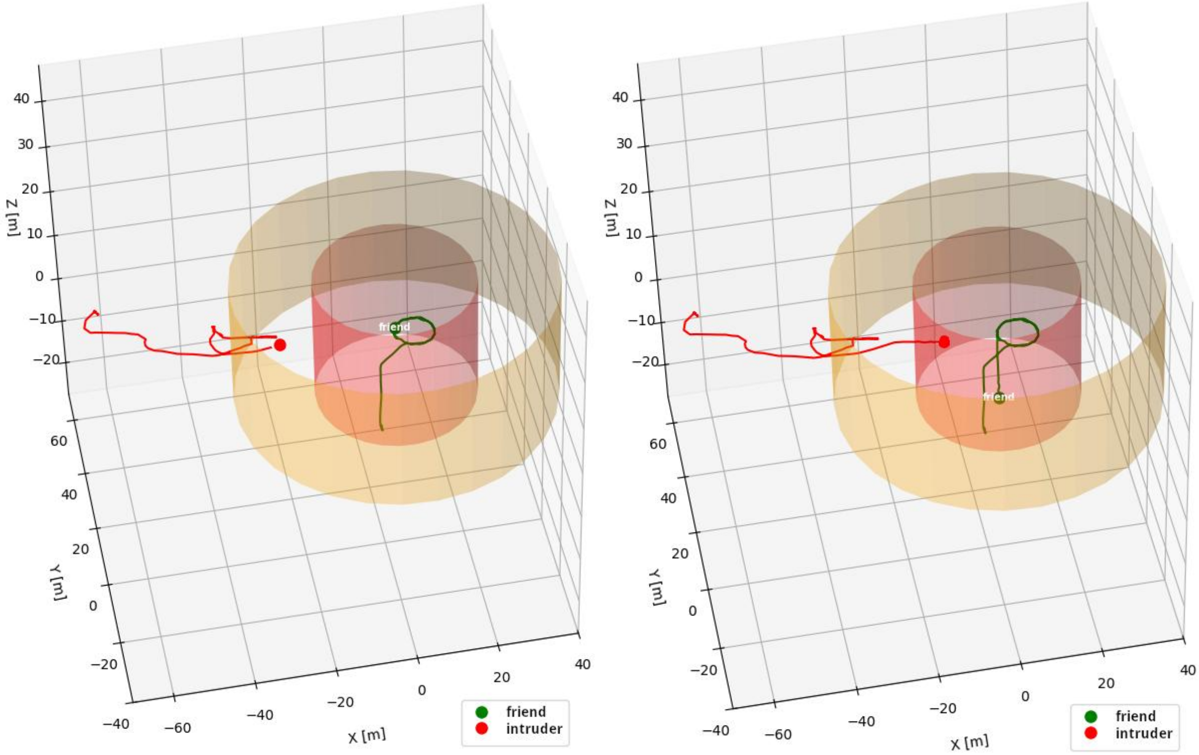


Figure 56. Intruder drone re-entering caution zone (left, t=93s), Intruder drone entering critical zone (right, t=119s)

7.6 Discussion

The demonstrated sense and avoid system relies on two critical components to function effectively:

1. Detection of the intruding drone: This is achieved through ground-based sensors capable of identifying and tracking the intruder's position in real-time. The reliability and accuracy of this detection layer hence directly influences the system's ability to respond promptly and appropriately to potential airspace conflicts. During the experiments the detection rate was 100% without any false positives. The latency of the detection depends on several factors such as drone transmission update rate, signal capture and processing delay, decoded data processing. In the experiments the latency was around 1s, with a worst-case latency of around 1,4s. Based on this latency and the speed of the drone, the accuracy can be calculated. During the experiment, a DJI Air3 was used which has a max. speed of 19 m/s, so the worst-case accuracy is varying around 19m up to 27 m (1s up to 1.4s * 19 m/s). However, during the demo the accuracy was much higher, as the drone was not operated at its maximum speed. Thus, the safety radii were set to only 20 m and 40m.

2. Communication with the friendly drone: Once an intrusion is detected, relevant information must be quickly transmitted to the friendly drone to initiate the appropriate response (e.g.,

pause and hover, or land). This requires a robust and low-latency communication link between the ground system and the drone. The resilience and reliability of this network are essential, as any loss or delay in communication could compromise the drone's ability to execute timely evasive actions.

While this demonstration used simple hover and auto-landing maneuvers as illustrative evasive actions, more sophisticated behaviors – such as dynamic path replanning or altitude adjustments – could be implemented in operational settings. The choice of avoidance strategy should be informed by the mission context, available sensor data, regulatory constraints, and the expected behavior of potential intruders (cooperative or not).

While the demonstrated system does not rely on a dedicated 6G infrastructure, it reflects several key directions in the developments towards 6G. One of these is the concept of "network as a sensor", where the communication infrastructure itself contributes to environmental awareness. Although we used a different system not based on a dedicated 6G infrastructure, the Skysense system shows how passive RF sensing and real-time data sharing between aerial platforms can enhance situational awareness and safety in shared airspace. In future 6G scenarios, such sensing capabilities could be integrated directly into the network infrastructure, enabling cooperative detection, positioning, and coordination without the need for separate sensing systems. Combining this approach with low-latency communication protocols (e.g., MQTT in this demonstration) and autonomous decision logic mirrors the convergence of communication, sensing, and control envisioned in 6G. These kinds of systems, when integrated at scale, could support UTM and other safety-critical applications that rely on distributed intelligence and high reliability.

8 Task 5.6: Demonstration of HAPS networking

Lead: Deutsche Telekom, **Contributors:** Airbus, Fraunhofer.

8.1 Overview

The purpose of this task is to test a Remotely Piloted Aircraft System (RPAS), provided by a subcontractor, as a flexible HAPS platform capable of integrating various payload systems for proof-of-concept demonstrations in the low stratosphere. The platform provides pressurized and unpressurized payload bays to accommodate prototypical payloads not originally designed for stratospheric thermal and pressure conditions. The main test site is located in Germany.

As part of the demonstration, multiple link types will be tested, including HAPS-Ground, HAPS-low altitude UAV, HAPS-HAPS, and HAPS-Satellite links. The test scenarios focus on validating KPIs that are critical for 6G network differentiation, such as link quality, capacity, reliability, and latency.

In addition, new RF spectrum areas are being explored for 6G. One of the objectives of the HAPS demonstration is to establish a communication link between the HAPS platform and UE operating in the frequency range that are under consideration for 6G.

8.2 Demonstration Setup

8.2.1 HAPS Platform

As the HAPS platform for the demonstration, we use Grob Aircraft G520NG[10] (Figure 57) operated by Grob Aircraft. It is one of the world's largest fully composite manned and unmanned aircraft, providing a suitable platform for OPV/UAV applications (Optionally Piloted Vehicle).

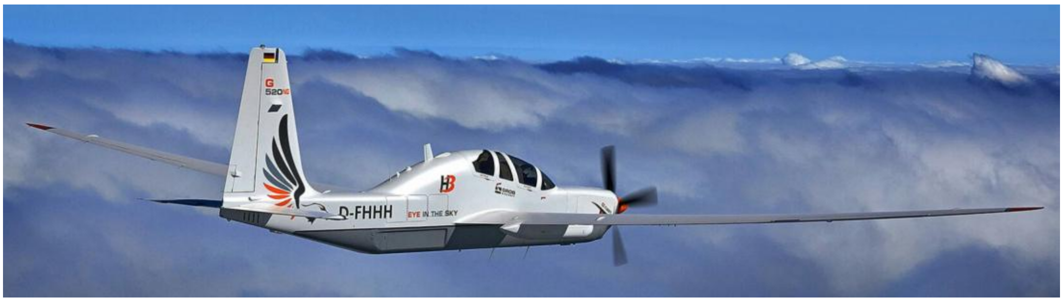


Figure 57. Grob G520NG

[10] https://grob-aircraft.com/en/g-520ng.html, accessed 2025-06-06

The capabilities of the platform are detailed in Table 7.

Table 7. G520 Platform capabilities

Max. payload	1145 kg
Cargo volume	3.84 m3 + 1.35 m3 pressurized cabin space
Max certified operating altitude	13,716 m (FL450)
Operating conditions	Visual Flight Rules / Instrument Flight rules day and night / flight into known icing conditions

The communications equipment is integrated into different payload segments of the aircraft as shown in Figure 58. For our demonstration payload, components are kept in pressurized cockpit and are integrated to modular "U-Bay" payload area. Satellite antenna is integrated into a special fairing on top of the fuselage (see Figure 59).

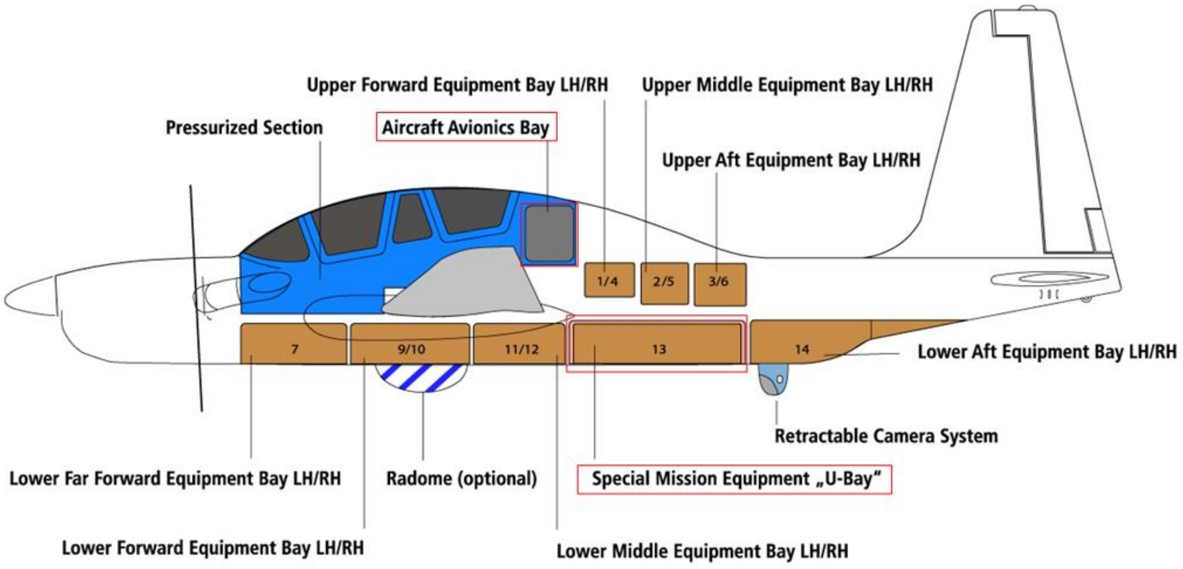

Upper Forward Equipment Bay LH/RH
Upper Middle Equipment Bay LH/RH
Pressurized Section
Aircraft Avionics Bay
Upper Aft Equipment Bay LH/RH
1/4
2/5
3/6
7
9/10
11/12
13
14
Lower Aft Equipment Bay LH/RH
Retractable Camera System
Lower Far Forward Equipment Bay LH/RH
Radome (optional)
Special Mission Equipment „U-Bay"
Lower Forward Equipment Bay LH/RH
Lower Middle Equipment Bay LH/RH

Figure 58. Grob G520NG Payload Integration options

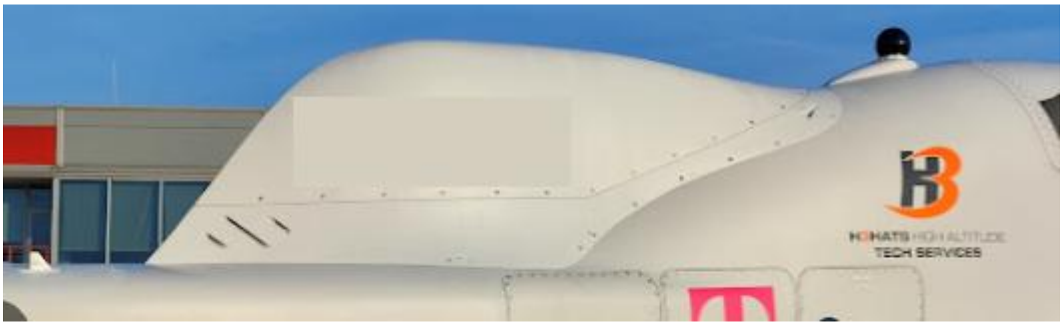


Figure 59. Fairing extension for Satellite Antenna

8.2.2 6 GHz Communication Demonstrator

The high-level architecture of the demonstrator is shown in Figure 60. To emulate a 6 GHz communications system, we use a 3GPP Band 1 Base Station and commercial UE modified to use cabled connection (Figure 61) instead of standard built in antennas, connected on both sides to frequency transverters to provide over-the-air link on >6 GHz frequency range.

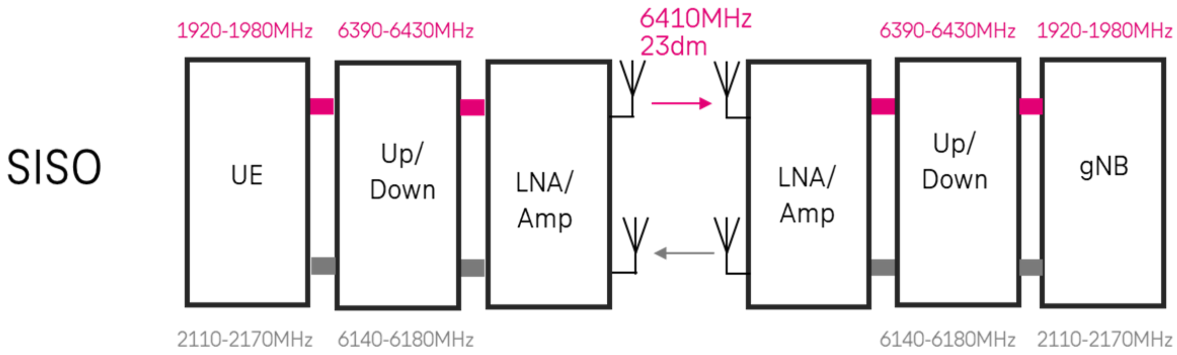


Figure 60. High-level 6 GHz demonstrator architecture

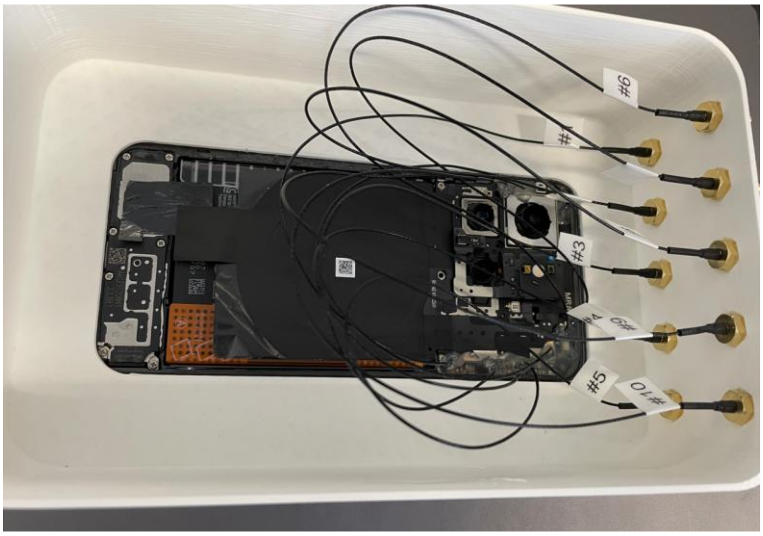


Figure 61. UE with cabled RF paths

8.3 Planned Flight Trials and Measurement Campaigns

The full flight trials and detailed measurements for the HAPS demonstration are planned to take place in Q3 2025, once all necessary system components and operational approvals are in place. These include the final preparation of the UE, Base Station, Up/Down converters and LNA/Amp.

A detailed measurement plan, including specific test points, will be executed during the flights. Log collection will be performed to enable the evaluation of key performance indicators (KPIs) such as link quality, capacity, reliability, latency, and system stability. This chapter will be updated once the flight trials and measurement campaign have been conducted.

9 Contributions to Sustainability

Throughout the 6G-SKY project, sustainability has been a guiding principle across system design, testing, and demonstration activities. Our research and proposed solutions align with global sustainability objectives, particularly the United Nations Sustainable Development Goals (SDGs), such as SDG 7 (Affordable and Clean Energy) and SDG 9 (Industry, Innovation, and Infrastructure). By leveraging an optimized integration of terrestrial and non-terrestrial networks, the work focuses on enhancing energy efficiency, reducing environmental impact, and fostering long-term infrastructure resilience.

In the context of 6G-SKY work package 5 the following contributions to sustainability can be identified:

- Extensive use of advanced lab emulation significantly reduced the need for energy-intensive field trials, limiting emissions from travel, UAV flights, and the operation of ground equipment. This approach supports more sustainable R&D processes, while maintaining rigorous system validation.
- In terms of practical applications, 6G-SKY's UAV-based solutions aim to improve operational efficiency in industries such as logistics. The drone swarm demonstrated for automated container inspection can help reduce unnecessary vehicle movements (trucks, stackers) in cargo terminals, thereby lowering fuel consumption and associated CO_2 emissions. Moreover, the system paves the way for supporting the transition toward autonomous, electrified logistics operations.
- The HAPS networking work within 6G-SKY points to new opportunities for sustainable rural connectivity. HAPS can help mobile operators reduce grid energy use by allowing partial or temporary deactivation of terrestrial base stations in low-demand areas.

10 Conclusion

The 6G-SKY project has successfully demonstrated key enablers for future 6G and NTN communications through an integrated program of lab validation and real-world testing. The project achieved reliable and seamless handover between terrestrial and non-terrestrial networks, with 100% success for both voice and data services. Latency performance met expectations and remained consistent even during handovers.

Advanced multi-link communication architectures were validated, integrating Wi-Fi mesh, 5G campus networks, and satellite links. While Multipath TCP (MPTCP) provided redundancy, its buffering effects highlighted the need for future protocol enhancements to support latency-sensitive, high-data-rate applications. AI-driven network intelligence was also advanced through the development of an explainable AI framework for UAV handover decisions, which has shown promising initial results.

Practical demonstrations across multiple domains validated the applicability of the project’s innovations. Use cases included autonomous drone swarms for container inspection, drone-based mobile relays supporting ground vehicle connectivity in challenging environments, and autonomous sense and avoid capabilities for airspace safety. The HAPS networking testbed established within the project will enable continued exploration of multi-connectivity links critical for 6G systems.

In summary, 6G-SKY has delivered important technical results and practical insights that contribute to the ongoing evolution of 6G architectures. The project outcomes provide a solid foundation for future research, standardization activities, and industrial exploitation in the domain of integrated aerial and non-terrestrial networks.

Glossary

Abbreviation	Description
3GPP	3rd Generation Partnership Project
(5G) NR	(5G) New Radio
(5G) SA	(5G) Stand Alone
(D)A2A	(Direct) Air-to-Air
(D)A2G	(Direct) Air-to-Ground
AWGN	Additive White Gaussian Noise
CW	Continuous Wave
D2D	Device-to-Device
D2I	Device-to-Infrastructure
DL	Downlink
FDD	Frequency Division Duplexing
HAP(S)	High Altitude Platform (Station)
GEO	Geostationary Earth Orbit (satellite)
gNB	“gNodeB”, 5G base station
GPS	Global Positioning System
KPI	Key Performance Indicator
LEO	Low Earth Orbit (satellite)
LOS	Line-of-Sight
LTE	Long Term Evolution
MAC	Medium Access Control
MEO	Medium Earth Orbit (satellite)
MIMO	Multiple Input Multiple Output
MPTCP	Multipath Transmission Control Protocol
MQTT	Message Queuing Telemetry Transport
NLOS	Non-Line-of-Sight
NTN	Non-Terrestrial Network
OAI	OpenAirInterface
OCR	Optical Character Recognition

OPV	Optionally Piloted Vehicle
RAN	Radio Access Network
RPAS	Remotely Piloted Aircraft Systems
RPi(4)	Raspberry Pi (4) single board computer
RRC	Radio Resource Control
RTT	Round-Trip-Time
SDR	Software Defined Radio
SISO	Single Input Single Output
TDD	Time Division Duplexing
TN	Terrestrial Network
UAV	Uncrewed Aerial Vehicle - Drone
UE	User Equipment
UGV	Uncrewed Ground Vehicle
UL	Uplink
USRP	Universal Software Radio Peripheral
V2V	Vehicle-to-Vehicle
VoNR	Voice over New Radio

List of Figures

List of Tables