ENABLING UNMANNED AERIAL VEHICLES FOR THE NEAR FUTURE APPLICATIONS

by

VÍCTOR SÁNCHEZ AGÜERO

in partial fulfillment of the requirements for the degree of Doctor in

Telematic Engineering

Universidad Carlos III de Madrid

Advisor: Francisco Valera, Ph.D.

May 2022

Enabling Unmanned Aerial Vehicles for the near future applications

Prepared by:
Víctor Sánchez Agüero, IMDEA Networks Institute, Universidad Carlos III de Madrid
contact: victor.sanchez@imdea.org

Under the advice of:
Francisco Valera, Universidad Carlos III de Madrid
Telematic Engineering Department, Universidad Carlos III de Madrid

This work has been supported by:

"Lo bueno si breve, dos veces bueno; y aun lo malo, si poco, no tan malo"

— Baltasar Gracián

Acknowledgements

In the first place, I would like to thank my advisor **Paco**, who allowed me not only to do this Ph.D. but also to grow personally and professionally under his guidance. He has been more than an advisor during all these years, sometimes someone halfway between a father and an older brother.

I would also like to thank my co-authors, who have contributed invaluably to the development of this thesis. Among all of them, I must highlight two: First, I want to thank **Ivan**, who has been a reference during these years. He has acted as a sort of co-advisor whose ideas, advice, and help have been essential. On the other hand, I would like to thank **Borja** for being an exceptional fellow and helping me unselfishly whenever I needed him. I am very grateful to have shared this adventure with him from beginning to end.

Furthermore, I would also like to thank professors and fellows at the university: Ángel, Carlos, Cristina, Dulce, Gonzalo, Ginés, J. Alberto, Jesús, Kiril, Marco, Miguel, Milan, Oscar, Patricia, Pedro, Pelayo, Roderick, Rubén, Sergio, Stefano, Winnie, and especially my colleagues at the office: **Jorge**, Borja, **José**, **Lewis**, and **Antonio**, for making life easier.

My most deep thanks to all my family (including in-laws) for their support. Mainly I would like to express my gratitude to my parents **Marita** and **Julián** for providing all my education and offering me a life full of opportunities and facilitates.

Last but not least, I would like to dedicate this work to **Teresa**, my traveling partner and my dear future wife.

Published and Submitted Content

According to the Law 14/2011 about plagiarism and Code of Good Practices of the Universidad Carlos III de Madrid Doctoral School, I hereby detail the list of papers I have (co)authored that are included as part of this thesis:

[**1**] **Victor Sanchez-Aguero**, Borja Nogales, Francisco Valera and Ivan Vidal. Flexible services deployment using Small Unmanned Aerial Vehicles for emergency situations. Published in *XXXIII Simposium Nacional de la Union Cientifica Internacional de Radio (URSI'18)*, 2018.

- This work is partially included and its content is reported in Chapter 1;
- The author's role in this work is focused on leading the writing of the paper as well as the analysis and identification of enabling 5G technologies connected to UAVs;
- The material from this source included in this thesis is not singled out with typographic means and references.

[**2**] Ivan Vidal, Paolo Bellavista, **Victor Sanchez-Aguero**, Jaime Garcia-Reinoso, Francisco Valera, Borja Nogales and Arturo Azcorra. Enabling Multi-Mission Interoperable UAS Using Data-Centric Communications. Published in *MDPI Sensors,* October 2018. https://doi.org/10.3390/s18103421

- This work is partially included and its content is reported in Appendix B;
- The author's role in this work is focused on design, implementation, and experimentation;
- Whenever material from this source is included in this thesis, it is singled out with typographic means and an explicit reference.

[**3**] Ivan Vidal, **Victor Sanchez-Aguero**, Francisco Valera, Borja Nogales, Jaime Cabezas, Carlos Vidal, Alicia Lopez, Daniel Gonzalez, Jose Diez, Laura Berrazueta and Manuel Merino. MILANO: una vision futura para un UAS tactico. Published in *VI Congreso Nacional de i+d en Defensa y Seguridad. DESEi+d'18,* Noviembre 2018, Valladolid, Spain. http://hdl.handle.net/10016/28959

- This work is partially included and its content is reported in Chapter 3;
- The author's role in this work is focused on the analysis of future research lines;
- The material from this source included in this thesis is not singled out with typographic means and references.

[**4**] Borja Nogales, **Victor Sanchez-Aguero**, Ivan Vidal, Francisco Valera. Adaptable and Automated Small UAV Deployments via Virtualization. Published in *MDPI Sensors,* December 2018. https://doi.org/10.3390/s18124116

- This work is partially included and its content is reported in Chapter 2;
- The author's role in this work is focused on the analysis of routing protocols;
- Whenever material from this source is included in this thesis, it is singled out with typographic means and an explicit reference.

[**5**] Christian Tipantuña, Xavier Hesselbach, **Victor Sanchez-Aguero**, Francisco Valera, Ivan Vidal and Borja Nogales. An NFV-Based Energy Scheduling Algorithm for a 5G Enabled Fleet of Programmable Unmanned Aerial Vehicles. Published in *Wireless Communications and Mobile Computing,* February 2019. https://doi.org/10.1155/2019/4734821

- This work is partially included and its content is reported in Chapter 6;
- The author's role in this work is focused on extensive state of the art revision as well as analysis and experimentation;
- Whenever material from this source is included in this thesis, it is singled out with typographic means and an explicit reference.

[**6**] **Victor Sanchez-Aguero**, Francisco Valera, Borja Nogales, Luis F. Gonzalez and Ivan Vidal. VENUE: Virtualized Environment for multi-UAV network emulation. Published in *IEEE Access,* October 2019. https://doi.org/10.1109/ACCESS.2019.2949119

- This work is fully included and its content is reported in Chapter 2;
- The author's role in this work is focused on leading the writing of the paper as well as focused on design, implementation, and experimentation;
- The material from this source included in this thesis is not singled out with typographic means and references.

[**7**] Borja Nogales, Ivan Vidal, **Victor Sanchez-Aguero**, Francisco Valera, Luis F. Gonzalez, Arturo Azcorra. Automated Deployment of an Internet Protocol Telephony

Service on Unmanned Aerial Vehicles Using Network Functions Virtualization. Published in *Journal of Visualized Experiments,* October 2019. http://dx.doi.org/10.3791/60425

- This work is partially included and its content is reported in Chapter 2;
- The author's role in this work is focused on a preliminary validation of the proposed solution;
- Whenever material from this source is included in this thesis, it is singled out with typographic means and an explicit reference.

[8] **Victor Sanchez-Aguero**, Francisco Valera, Ivan Vidal, Christian Tipantuña, Xavier Hesselbach. Energy-aware management in Multi-UAV deployments: Modeling and Strategies. Published in *MDPI Sensors,* May 2020. https://doi.org/10.3390/s20102791

- This work is fully included and its content is reported in Chapter 6;
- The author's role in this work is focused on leading the writing of the paper as well as focused on design, implementation, and experimentation;
- The material from this source included in this thesis is not singled out with typographic means and references.

[9] **Victor Sanchez-Aguero**, Ivan Vidal, Francisco Valera, Borja Nogales, Luciano Leonel Mendes,Wheberth Damascena Dias, and Alexandre Carvalho Ferreira. Deploying an NFV-Based Experimentation Scenario for 5G Solutions in Underserved Areas. Published in *MDPI Sensors,* March 2021. https://doi.org/10.3390/s21051897

- This work is fully included and its content is reported in Chapter 4;
- The author's role in this work is focused on leading the writing of the paper as well as focused on design, implementation, and experimentation;
- The material from this source included in this thesis is not singled out with typographic means and references.

[10] **Victor Sanchez-Aguero**, Luis F. Gonzalez, Francisco Valera, Ivan Vidal, Rafael A. López da Silva. Cellular and virtualization technologies for UAVs: an experimental perspective. Published in *MDPI Sensors,* April 2021. https://doi.org/10.3390/s21093093

- This work is fully included and its content is reported in Chapter 5;
- The author's role in this work is focused on leading the writing of the paper as well as focused on design, implementation, and experimentation;
- The material from this source included in this thesis is not singled out with typographic means and references.

[11] Victor Sanchez-Aguero, Miguel-Angel Fas-Millan, Francisco Valera, Ivan Vidal, Alejandro Paniagua-Tineo, Rafael A. Lopez da Silva, and Jose Manuel Manjon. Multi-interface network framework for UAV management and data communication. Published in *2021 IEEE Globecom Workshops (GC Wkshps): Workshop on Cellular UAV and Satellite Communications,* December 2021. https://doi.org/10.1109/GCWkshps52748.2021.9682019

- This work is fully included and its content is reported in Chapter 7;
- The author's role in this work is focused on leading the writing of the paper as well as focused on design, implementation, and experimentation;
- The material from this source included in this thesis is not singled out with typographic means and references.

[12] Victor Sanchez-Aguero, Francisco Valera, Ivan Vidal, Borja Nogales, Jaime Cabezas, and Carlos Vidal. A virtualization approach to validate services and subsystems of a MALE UAV. Published in *2022 IEEE INFOCOM Workshops: The 9th International Workshop on Computer and Networking Experimental Research using Testbeds,* May 2022. **Best paper award**.

- This work is fully included and its content is reported in Chapter 3;
- The author's role in this work is focused on leading the writing of the paper as well as focused on design, implementation, and experimentation;
- The material from this source included in this thesis is not singled out with typographic means and references.

[13] Victor Sanchez-Aguero, Francisco Valera, Ivan Vidal, Borja Nogales, Daniel Gonzalez del Río. Software Defined RPA Networks. **Submited** in *IEEE Communication Magazine.*

- This work is fully included and its content is reported in Chapter 8;
- The author's role in this work is focused on leading the writing of the paper as well as focused on design, implementation, and experimentation;
- The material from this source included in this thesis is not singled out with typographic means and references.

Other Research Merits

Other Research Merits includes the list of articles I have (co)authored, highly related to the topic of this thesis, but not included:

[14] **Victor Sanchez-Aguero**, Borja Nogales, Francisco Valera and Ivan Vidal. Investigating the deployability of VoIP services over wireless interconnected Micro Aerial Vehicles. Published in *Internet Technolgy Letters, Wiley*, 2018. https://doi.org/10.1002/itl2.40

[15] Borja Nogales, **Victor Sanchez-Aguero**, Ivan Vidal, Francisco Valera and Jaime Garcia-Reinoso. An NFV system to support configurable and automated multi-UAV service deployments. Published in *4th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use (DroNet 2018)* https://doi.org/10.1002/itl2.40

[16] Roderick Fanou, **Victor Sanchez-Aguero**, Francisco Valera, Michuki Mwangi, Jane Coffin. A System for Profiling the IXPs in a Region and Monitoring their Growth: Spotlight at the Internet Frontier. Published in *International Journal of Network Management,* January 2019. https://doi.org/10.1002/nem.2056

[17] Luis F. Gonzalez, Ivan Vidal, Francisco Valera, **Victor Sanchez-Aguero**, Borja Nogales and Diego R. Lopez. NFV orchestration on intermittently available SUAV platforms: challenges and hurdles. Published in *2019 IEEE INFOCOM Workshop: MiSARN 2019: Mission-Oriented Wireless Sensor, UAV and Robot Networking,* April 2019. https://doi.org/10.1155/2019/4734821

[18] Luis F. Gonzalez, Ivan Vidal, Francisco Valera, Borja Nogales, **Victor Sanchez-Aguero**, Diego R. Lopez. Transport Layer Limitations for NFV Orchestration in Resource Constrained Aerial Networks. Published in *MDPI Sensors,* December 2019. https://doi.org/10.3390/s19235220

[19] Ivan Vidal, Borja Nogales, Francisco Valera, Luis F. Gonzalez, **Victor Sanchez-Aguero**, Eduardo Jacob, Cristina Cervelló-Pastor. A Multi-site NFV Testbed for Experimentation with SUAV-based 5G Vertical Services. Published in *IEEE Access,* June 2020. https://doi.org/10.1109/ACCESS.2020.3001985

[**20**] Luis F. Gonzalez, Ivan Vidal, Francisco Valera, **Victor Sanchez-Aguero**. A Comparative Study of Virtual Infrastructure Management Solutions for UAV Networks. Published in *7th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications (DroNet 2021),* June 2021. https://doi.org/10.1145/3469259.3470486

Project Acknowledgements

This doctoral thesis has been carried out in the framework of the European and National projects listed below:

5GCity

Adaptive Management of 5G Services to Support Critical Events in Cities (Grant agreement TEC2016-76795-C6-3-R, Spanish Ministry of Economy and Competitiveness):

The overall objective of this project is to provide adequate support to the smart city paradigm in one of the most challenging situations for current communication systems to deal with: unexpected events involving a relatively high number of mobile users concentrated in a small area, such as traffic concentrations due to congestion or accidents, disasters, or emergencies where a large number of users are affected. The approach is to develop an adaptive solution to control 5G technologies and configure communication services while maintaining good scalability and high reliability and providing very low latency if necessary. Relevant entities such as the Polytechnic University of Catalonia, the University of the Basque Country, the University of Granada, and the i2cat foundation participate in this project.

The role and the activities of the author in the project are the following:

- Participating from February 2017 until the end of the project in June 2019.
- Mainly involved in Objective 3 (Design of a high performance and efficient NFV architecture for 5G aware Fog Computing deployments) and Objective 6 (Implementation of the 5GCity Global Testbed).

TRUE5G

Towards zeRo toUch nEtwork and services for beyond 5G (Grant agreement PID2019-108713RB681):

The TRUE5G project arises from the valuable results of 5GCITY, where most TRUE5G partners participated. The 5GCITY proposal conceived a horizontal approach to tackle the issue of Software Defined Networking (SDN)/Network Function

Virtualization (NFV) based 5G management for critical events, while the TRUE5G aims to extend these results to a multi-operator, multi-technology, ML-based solution to integrate different verticals and provide a heterogeneous 5G experimentation platform.

The role and the activities of the author in the project are the following:

- Participating from the beginning in June 2020 (active until May 2023).
- Mainly involved in Objective 1 (Smart network slicing management) and Objective 2 (Design a dynamic multi-domain and multi-technology service orchestration).

H2020 5GRANGE

Remote area Access Network for 5th GEneration (Grant agreement 777137):

The primary purpose of the 5GRANGE project is to design, develop, and implement a solution that enables the 5G network to provide Internet access in remote areas cost-effectively and efficiently. 5GRANGE aims for a cell radius above 50 km with at least 100 Mb/s at the edge, employing both licensed and unlicensed frequencies, while cognitive radio techniques will be used to protect incumbents. The developed solutions have been integrated into a Proof-of-Concept testbed to allow for field performance evaluation and real case demonstrations. Relevant entities such as the University of Oulu, the TU Dresden, INATEL, or Telefonica I+D participate in the project.

The role and the activities of the author in the project are the following:

- Participating from the beginning in November 2017 until the end of the project in October 2020.
- Mainly involved in WP5 whose objective is to design and deploy the necessary mechanisms to be able to provide 5G-RANGE with end-to-end IP network connectivity.
- Active participation in the prototyping and writing of deliverables.
- Final demonstration of the project.

H2020 Labyrinth

Unmanned Traffic Management 4D path planning technologies for drone swarm to enhance safety and security in transport (Grant agreement 861696):

Unmanned Aerial Vehicle (UAV) technologies are increasingly being presented as an effective way to improve the safety and efficiency of civilian transportation. However, for safety and security reasons, UAVs cannot fly at low altitudes in most civilian environments because remote guidance and control technologies have not reached the necessary level of maturity. It is essential to have an air traffic management system for UAVs to ensure airspace safety. The EU-funded Labyrinth project study centralized planning systems capable of communicating with UAVs operating in particular areas and calculating routes/paths to avoid collisions. The project will create a U-space-based technology to meet the needs of autonomous UAV applications. This technology will be installed (by the end of the project) in civil scenarios such as roads, emergencies, seaports, and airports. These scenarios will be carried out by verticals such as the Spanish Central Traffic Headquarters (DGT), the Municipal Emergency and Rescue Assistance Service from Madrid (SAMUR), the Spanish National Institute for Aerospace Technology, the Italian Port Authority of the Eastern Ligurian Sea.

The role and the activities of the author in the project are the following:

- Participating from the beginning in June 2020 (active until May 2023).
- Mainly involved in WP4 whose objective is to design, validate and deploy the communications platform supporting the different components of Labyrinth.
- Active participation in the prototyping and writing of deliverables.

Abstract

Until today, Unmanned Aerial Vehicle (UAV) operations only include a single aerial vehicle (in most cases) that performs reconnaissance missions by sending telemetry captured by different onboarded sensors (e.g., video, temperature, air quality) to the Ground Control Station (GCS). Single-UAV applications, despite their apparent simplicity, are used in many different and significant fields (e.g., surveillance of livestock, monitoring of power lines, traffic monitoring, rescue). Many applications of UAV swarms have already been seen. Still, they are usually stunts and exhibitions with no actual functionality.

Recent research trends are founded on multiple UAVs operating collaborative implementing more complex services, and generally integrated into the urban environment. It would lead to new scenarios that are not yet adequately deployed (e.g., package delivery, monitoring of sports events or crowds such as concerts or demonstrations, increasing coverage, support to emergency services in cities (fire, police, emergency)). However, several challenges must be faced before integrating these applications into our everyday lives.

The central objective of the thesis is to contribute to some of the significant challenges identified in the UAV communications services sector. In the first place, this thesis contributes with an emulation solution for validating environments with connected UAVs, including different use cases and verticals. Additionally, it contributes to communications solutions in complex connectivity environments based on experimentation where the Fifth Generation of cellular network technology (5G) softwarization technologies are integrated into the UAV ecosystem. In the last place, this thesis contributes to the proposal of new solutions to solve some limitations, such as the high energy consumption in combination with UAVs' limited flight autonomy or the complexity of traffic management and the establishment of the network infrastructure in such volatile environments.

Table of Contents

Acknowledgements vii

Published and Submitted Content ix

Other Research Merits xiii

Project Acknowledgements xv

Abstract xix

Table of Contents xxi

List of Tables xxvii

List of Figures xxix

List of Acronyms xxxiii

I Introduction **1**

1. Introduction **3**

1.1. Motivation 3

1.1.1. UAV open research challenges 4

1.2. Thesis contributions 5

1.2.1. Thesis contributions timeline 6

1.2.2. Software released as Open Source 9

1.3. Outline of the thesis 10

II Contributions to UAV communications validation **11**

2. VENUE: Virtualized environment for multi-UAV network emulation **13**

2.1. Related work and background 14

2.1.1. Network simulators 14
2.1.2. FANETs 15
2.1.3. VENUE contributions 16
2.2. Life without VENUE 17
2.2.1. Scenario definition 17
2.2.2. Simulation results 19
2.3. VENUE design and implementation 20
2.3.1. Framework architecture 23
2.4. Life with VENUE 25
2.4.1. Use case I: Routing protocols for FANETs 25
2.4.2. Use case II: FANET to extend 5G connectivity 31
2.4.3. Use case III: VENUE in other research articles 36
2.5. Discussion 36
2.6. Research impact/dissemination 38

3. A virtualization approach to validate services and subsystems of a tactical UAV 39
3.1. The MILANO system 40
3.2. UAS virtualization platform 43
3.3. Experiments 46
3.4. Discussion 49
3.5. Research impact/dissemination 50

III Contributions to experimental environments 51

4. Deploying an NFV-Based experimentation scenario for 5G Solutions in underserved areas 53
4.1. Background on 5G-RANGE technologies 56
4.1.1. Physical and MAC layers 56
4.1.2. Network layer 57
4.1.3. NFV lestbed description and VNF repository 58
4.2. Experimentation scenario: Methodology from design to validation 60
4.2.1. Description of the experimentation scenario 61
4.2.2. Initial deployment of the experimentation scenario 62
4.2.3. Configuration of GRE/IPsec tunnel endpoints 63
4.2.4. Validation of the experimentation scenario 64
4.3. Multimedia tests and final integration 65
4.3.1. Testing voice and data services 65
4.3.2. Final integration and validation 66

4.4. Discussion 69
4.5. Research impact/dissemination 70

5. Cellular and virtualization technologies for UAVs: An experimental perspective 71
5.1. Related work and background 74
  5.1.1. Communications 74
  5.1.2. Virtualization and UAVs 75
5.2. Testbed components description 78
  5.2.1. Cellular-assisted UAV communications 79
  5.2.2. Automated UAV control system 81
  5.2.3. Virtualization in UAVs: The power of containers in aerial networks 82
5.3. Testbed experimentation 85
  5.3.1. Communication performance evaluation 85
  5.3.2. 5G Standalone benchmark 87
  5.3.3. Indoor flight validation 89
  5.3.4. Indoor flight with connectivity loss 91
5.4. Discussion 91
5.5. Research impact/dissemination 92

IV Contributions to connected UAVs 95

6. Energy-aware management in multi-UAV deployments: Modelling and strategies 97
6.1. Related work and background 99
6.2. Problem statement 101
  6.2.1. Complexity analysis 102
6.3. Methodology 104
  6.3.1. Reported parameters 104
  6.3.2. Assumptions 105
  6.3.3. Performance metric 106
  6.3.4. Scheduler algorithm proposals 107
6.4. Simulation details and results 114
  6.4.1. Simulation parameters 114
  6.4.2. Ground-Truth solutions 116
  6.4.3. Simulation setup 117
  6.4.4. Validation scenarios 117
  6.4.5. Comparison of the UAV replacement strategies 119
6.5. Discussion 123

6.6. Research impact/dissemination . . . 124

7. Multi-interface network framework for UAV management and data communication 125

7.1. Labyrinth details and requirements . . . 126
7.2. Labyrinth network framework . . . 128
  7.2.1. Inter-domain communication alternatives . . . 128
  7.2.2. Interface Manager . . . 130
7.3. Preliminary validation . . . 131
  7.3.1. Validation details . . . 131
  7.3.2. Experiments . . . 132
7.4. Discussion . . . 134
7.5. Research impact/dissemination . . . 135

8. Communication Infrastructure Manager for fully connected UAVs 137

8.1. Related work and background . . . 139
8.2. Communication Infrastructure Management . . . 141
  8.2.1. Reference scenario . . . 141
  8.2.2. Network layer connectivity . . . 142
  8.2.3. Data link layer connectivity (on top of Network Layer) . . . 143
  8.2.4. SDN switches . . . 143
  8.2.5. SDN controller . . . 144
  8.2.6. Network monitoring application . . . 144
8.3. Validation: Practical experiences . . . 145
  8.3.1. Choosing the minimum Round Trip Time (RTT) path . . . 146
  8.3.2. Load balancing . . . 148
  8.3.3. Real flight campaign . . . 150
8.4. Discussion . . . 152
8.5. Research impact/dissemination . . . 152

V Conclusions 153

9. Conclusions and future work 155

9.1. Conclusions . . . 155
9.2. Future work . . . 157

A. INTA Internship 161

A.1. About INTA . . . 161
A.2. Summary of activities . . . 162
A.3. Research outcome . . . 163

B. Efficient data dissemination 165

B.1. DDS practical experience 165

B.2. Main findings 172

References 173

List of Tables

1.1. Software released as Open Source. 9

2.1. Simulation parameters. 19

6.1. System parameters. 114
6.2. Simulation parameters A. 115
6.3. Simulation parameters B. 115

List of Figures

2.1. Simulation scenario 18
2.2. Simulation scenario 19
2.3. Simulation results 20
2.4. Framework architecture, node and global view 22
2.5. Scenario I: Routing protocols for FANETs. 27
2.6. Received throughput at destination UAV in one of the experiments (fails each 50 seconds). 30
2.7. Packet loss percentage in the first scenario for different replacements values. 31
2.8. Percentage of UAV Wi-Fi utilization. 31
2.9. Scenario II: multi-UAV network to extend 5G connectivity. 33
2.10. Real hardware used in the experiments. 34
2.11. OpenStack Control traffic in a Compute Node 35
2.12. OLSR signaling traffic at the Compute Node. 36
2.13. SIP and DNS traffic needed for the VoIP call. 36
2.14. Received Video and Audio at the destination. 37

3.1. MILANO testing campaign at Rozas Airport 41
3.2. MILANO Ground Control Station 42
3.3. Simplified INTA MILANO communication scheme 43
3.4. Virtual environment for MILANO platform 45
3.5. Traffic captured on the GCS Virtural Machine through the virtualization platform 48
3.6. Traffic captured on the GCS with connectivity loss introduced by VENUE 49

4.1. Overview of the testbed components and the experimentation scenario . . 54
4.2. High-level overview of the 5G-RANGE architecture. 57
4.3. Methodology to define, deploy, integrate and validate the experimentation scenario. 61
4.4. Data-plane protocol stack of the residential environment. 62
4.5. Performance evaluation of GRE/IPsec tunnel endpoints. 64
4.6. Data rates of SIP and Skype calls. 66

4.7. Data rates of video-on-demand service. 67
4.8. Transoceanic network path performance between 5G Telefonica Open Network Innovation Centre laboratory (5TONIC) and Inatel. 69

5.1. Different types of UAS following UTM: Beyond Radio LoS (BRLoS), and Radio LoS (RLoS). 72
5.2. High overview of the proposed testbed. 79
5.3. Details of UAV payload, and UAVs in flight. 80
5.4. Performance analysis using Ookla speed test. 87
5.5. Performance analysis upon the UAV-controller. 88
5.6. Measurements in indoor flight. 90
5.7. Measurements in indoor flight with connectivity loss. 92

6.1. Typical UAV use case using the proposed methodology. 99
6.2. Multi-UAV system during a mission for three target areas ($j = 3$) and four UAVs ($i = 4$). 102
6.3. Multi-UAV system states during a mission for three target areas ($j = 3$) and four UAVs ($i = 4$). 103
6.4. Differences between the analysed scheduling procedures. Example for $j = 2$ and $i = 3$ (2 UAVs in services and 1 UAV for replacement). 110
6.5. Proposed scenarios for algorithm performance evaluation. 111
6.6. Average throughput in different scenarios increasing the fleet size. 120
6.8. Approximation ratio ρ: betweenness centrality heuristic algorithm (BETA) vs. other suboptimal strategies. 122
6.7. Approximation ratio ρ: optimal strategy vs. heuristic strategies for Scenario I. 122
6.9. Number of UAV replacements using BETA in scenario III. 123

7.1. Elements and communication links in the Labyrinth U-space UTM system. 127
7.2. Labyrinth communications framework. 128
7.3. Validation scenario and components. 130
7.4. Telemetry data sent from the onboard computer (UAV) to the Telemetry Monitor (GCS). 133

8.1. Motivating scenario. 138
8.2. Reference scenario. 142
8.3. Proof-of-Concept (PoC) 1 scenario. 146
8.4. Traffic from UAV2 captured in the GCS (Red background - Losses). . . . 148
8.5. Traffic from UAV2 captured in the GCS for the load balancing experiment 149
8.6. INTA Unmanned Aircraft System (UAS) 151
8.7. Real flight scenario. 151

A.1. MILANO UAS at Rozas airfield. 163

B.1. Validation testbed including tactical UAV, GCS and small UAVs. 166
B.2. VXLAN and Linux bridge configuration. 168
B.3. DDS signaling (a,b); and data exchange (c,d). 169
B.4. Interruption of the reception of real-time video due to handover. 171
B.5. Measurements of the achieved throughput with: large PDU (left); and small PDU (right). 171

List of Acronyms

UAV Unmanned Aerial Vehicle

SUAV Small Unmanned Aerial Vehicle

UAS Unmanned Aerial System

MALE Medium Altitude Long Endurance

AODV Ad hoc On-demand Distance Vector

B.A.T.M.A.N Better Approach To Mobile Ad hoc Networking

DDS Data Distribution Service

MANET Mobile Adhoc Network

VANET Vehicular Adhoc Network

FANET Flying Adhoc Network

VENUE Virtualized Environment for Multi-UAV Network Emulation

NFV Network Function Virtualization

SDN Software Defined Networking

LXC Linux Containers

VNF Virtual Network Function

D2D Device to Device

QoS Quality of Service

MPB Mission Planned Based

OLSR Optimized Link State Routing

VM Virtual Machine

RPi Raspberry Pi

GCS Ground Control Station

VoIP Voice over IP

SBC Single Board Computer

VXLAN Virtual Extensible Local Area Network

UAS Unmanned Aircraft System

SAR Synthetic Aperture Radar

MALE Medium Altitude Long Endurance

LoS Line of Sight

BLoS Beyond Line of Sight

LXD Linux Daemon

UTM UAS Traffic Management

FAA Federal Aviation Administration

ATM Air Traffic Management

RLoS Radio Line-of-Sight

BRLoS Beyond Radio Line-of-Sight

SESAR Single European Sky ATM Research

3GPP 3rd Generation Partnership Project

GRE Generic Routing Encapsulation

IPsec Internet Protocol security

MANO Management and Orchestration

eMBB Enhanced Mobile Broadband

mMTC Machine Type Communications

URLLC Ultra-Reliable Low Latency Communications

5G-RANGE Remote area Access Network for 5th GEneration

5TONIC 5G Telefonica Open Network Innovation Centre laboratory

MAC Medium Access Control

QAM Quadrature Amplitude Modulation

MIMO Multiple-Input Multiple-Output

OSM Open-Source MANO

VIM Virtual Infrastructure Manager

NAT Network Address Translation

VPN Virtual Private Network

CPE Customer Premises Equipment

5G Fifth Generation of cellular network technology

6G Sixth Generation of cellular network technology

AP Access Point

CPU Central Processing Unit

RAM Random Access Memory

SIP Session Initiation Protocol

MTU Maximum Transmission Unit

BS Base Station

RTT Round Trip Time

KPI Key Performance Indicator

5GPPP 5G Infrastructure Public Private Partnership

EPC Evolved Packet Core

RAN Radio Access Network

MEC Multi-access Edge Computing

BETA Betweenness Centrality Heuristic Algorithm

C2 Command and Control

SA Stand Alone

NS Network Service

NFVI NFV Infrastructure

OS Operating System

FIM Fog Infrastructure Manager

VLL Very Low Level

CONOPS Concept of Operations

ATC Air Traffic Controller

ATSU Air Traffic Services Unit

BvLoS Beyond visual Line of Sight

LAN Local Area Network

IMU Initial Measurement Unit

UE User Equipment

SSH Secure SHell

ROS Robot Operating System

USB Universal Serial Bus

PC Personal Computer

ISP Internet Service Providers

HD High-Definition

NR New Radio

RTH Return To Home

DNS Domain Name System

IM Interface Manager

OvS Open Virtual Switch

PoC Proof-of-Concept

DAU Data Acquisition Unit

MTOW Maximum Take-Off Weight

DHCP Dynamic Host Configuration Protocol

CIM	Communication Infrastructure Manager
API	Application Programming Interface
GPS	Global Positioning System
IoT	Internet-of-Things
IP	Internet Protocol
JCR	Journal Citation Reports
LTE	Long Term Evolution
NAT	Network Address Translators
RF	Radio Frequency
UDP	User Datagram Protocol

Part I

Introduction

1

Introduction

1.1. Motivation

The emerging improvements in the Unmanned Aerial Vehicle (UAV) ecosystem enable the support of a plethora of innovative services both in the civil and military domains. UAVs are coming on the scene not only replacing conventional aircraft applications such as precision agriculture [21], surveillance [22], or border protection [23] but also for commercial/professional purposes in new areas such as civil construction [24], civil infrastructure inspection [25], medical supplies transportation [26], or package delivery [27] where crewed aircraft have not been used before. This information is supported by the Aerospace Forecast report by the U.S. Federal Aviation Administration (FAA)[1], which states that nowadays, most commercial UAV missions are devoted to research and development [28] or training and education [29].

Even settled UAV services are primarily founded on single-UAV use cases, the current research challenges are mainly focused on multi-UAV service provisioning (the so-called UAV swarm), whose communications are intrinsically related to the different Flying Adhoc Network (FANET) proposals [30] [31]. Different topics are studied to adopt this technology for a wide variety of applications [32], but not before solving many difficulties and challenges, as introduced in the following lines. Moreover, the utilization of multi-UAV networks, because of their rapid deployment, fast mobility, adaptability to different conditions, and flexibility, has recently attracted attention to support/extend the Fifth Generation of cellular network technology (5G) in extraordinary situations (e.g., massified events, natural disasters [33] [34], infrastructure failures [35]). The 5G networks will certainly bring faster uploading and downloading speeds in combination with a dramatic decrease in network latency. However, in exceptional or emergency circumstances, the deployment of 5G terrestrial infrastructure may not be economically viable. In addition, the deployment times of these extraordinary on-demand 5G network services should meet

---

[1] FAA Aerosp. Forecasts, accessed Tuesday 15th November, 2022. Available: https://www.faa.gov/data_research/aviation/aerospace_forecasts/

the Key Performance Indicator (KPI) [36] defined by the 5G Infrastructure Public Private Partnership (5GPPP), which states that new deployments must finish within 90 minutes. Accordingly, it is here where UAVs are also expected to play a crucial role [37]. If properly deployed and configured, UAV networks can provide fast-ubiquitous 5G access (which is also within the 5G KPIs), employing wireless communications solutions in a diversity of real-world scenarios [38] [39].

Despite the optimistic predictions, the sector is still waiting for consistent regulation and reliable solutions to enable multiple UAV commercial operations safely. In this regard, the scientific community is focused on the development of new solutions. Although the promising outcome provided by the utilization of UAVs, there are still many domains to be explored. Some challenges/problems require to be solved before stable and reliable solutions can be achieved [40] as detailed in the following subsection.

1.1.1. UAV open research challenges

This subsection covers some of the most significant challenges to be solved by the research community regarding UAVs. Moreover, some of them are detailed in more depth in the thesis chapters that attempt to address them.

The first big challenge encompasses the UAV communications. Unlike previous mobile ad hoc networks, multi-UAV networks have different characteristics depending on the nature of the mission [40]. For example, the UAV network can be stationary (e.g., UAV nodes to act as base stations [41]), have slow/intermediate mobility (e.g., UAV sensing [42]), or have fast mobility (e.g., multi-UAV military [43]) depending on the target application. Therefore, the probability of delay/disruption because of a node or link failure is also variable. Consequently, the proposed solutions must be flexible enough to adapt to any potential circumstance. On the other hand, legacy mobile network nodes have traditionally operated as clients. However, multi-UAV application nodes prospectively act as servers (e.g., routing packets and sending video or sensor information to a centralized control station). Therefore, considering these unique characteristics, it is necessary to assess the different challenges to obtain consistent and reliable solutions.

The characteristics mentioned above also affect the routing of data packets. The multi-UAV network topology often changes, resulting in the (dis)appearance of nodes and links that alter the performance of the communications. In this sense, routing protocols (or the technologies that enable communications) must allow: *(i)* a certain degree of mobility, *(ii)* to work for different architectural designs, or *(iii)* to incorporate reconfiguration mechanisms among others.

Additionally, UAV networks must be secured to guarantee the safety of the missions since any potential malicious attack affecting UAVs not only brings down the service they are carrying out (with the consequences that this implies) but also entails a material risk of accident in civilian environments.

Regarding the communication challenge some topics are handled in Chapters 2, 3, 4, 5, 6, 7, and 8.

Another representative problem of UAVs is that they have a limited energy capacity. Consequently, the flight time of UAVs, as well as the UAV network lifetime, is highly affected by the energy consumption, which is also related to different research fields such as optimal trajectory calculation or the transmission power of the communication alternatives used. In this sense, this fact also affects the equipment used as payload (e.g., sensors, cameras, packages for delivery), which likewise have to be reduced as much as possible regarding weight and volume, not impacting energy consumption. This challenge is handled in Chapter 6.

The optimal planning of routes and trajectories is a significant open challenge. UAV trajectories must be calculated by taking into account not only collisions with other objects that UAVs may come across (e.g., buildings, trees, traffic lights) but also taking into account different considerations that potentially affect the UAV network, such as maintaining connectivity with other network participants (e.g., UAVs, Ground Control Station (GCS), terrestrial participants), or energy constraints. At the same time, the UAV sector still awaits a consistent/uniform regulation to safely enable multiple UAV commercial operations in the same airspace. In this context, relevant institutions such as the FAA, Google, Amazon, or NASA work together on the UAS Traffic Management (UTM) project. In this regard, the European Union is also investing in many efforts over these research lines by the Single European Sky ATM Research (SESAR). The SESAR initiative is developing U-Space, a set of new services to uphold safe, effective, and secure access to European airspace for a large number of Unmanned Aircraft System (UAS). However, none of these proposals has reached a significant stage of maturity sufficient to be implemented.

Finally, the above challenges highlight the differences between multi-UAV and traditional wireless networks. In this sense, it is required to develop validation tools to test the proposed research solutions that aim to solve the previous challenges. This challenge is handled in detail in Chapter 2 and later used for experimentation in Chapters 3, 7, and 8.

1.2. Thesis contributions

The main objective of this thesis is to boost the UAV near-future applications. In order to accomplish this challenge, different contributions are proposed from a practical/functional approach.

First, we detected a lack of validation tools that allow an in-depth evaluation of multi-UAV research solutions and proposals. In this regard, this thesis contributes with an emulation platform designed with the unique attributes of multi-UAV networks in mind.

This platform aids the validation of all those proposals that, for several reasons, are not yet verified/demonstrated in real scenarios. After that, we present further developments following a practical approach predominantly guided by experimentation, where different 5G softwarization technologies, such as Network Function Virtualization (NFV), Software Defined Networking (SDN), or Data Distribution Service (DDS), are integrated with the UAV ecosystem. This work enables the development of different use-cases and testbeds and indicates all the research domains that need additional evolution before incorporating UAVs into civil society. Finally, we introduce some proposals to overcome the previously identified problems and challenges inherent to multi-UAV networks.

Since the thesis content is firmly based on research articles that are already published, we decide to be plausible as much as possible to the original content of the articles. Therefore, thesis chapters are self-contained, even at the risk of being repetitive in their introduction, since some ideas will be redefined to properly understand the chapter content without the necessity of reading the whole document.

The main contributions of this thesis have been published/submitted in 13 publications, of which five have been published in *MDPI Sensors*, one has been published in *IEEE Access*, one has been published in *Journal of Visualized Experiments* journal, and one has been submitted in *IEEE Communication Magazine* (all of them indexed in Q1 Journal Citation Reports (JCR) at the time they were published). Another publication has been published in *Wireless Communications and Mobile Computing* journal (indexed in Q3 JCR).

Finally, four publications have been published in workshops located at international and national conferences. (*IEEE Global Communication Conference*, *IEEE International Conference on Computer Communications*, *VI Congreso Nacional de i+d en Defensa y Seguridad. DESEi+d'18*, and *Simposium Nacional de la Unión Científica de Radio Internacional*).

1.2.1. Thesis contributions timeline

After active participation in different research projects (already mentioned at the beginning of the document), we discovered the lack of simulators, emulators, or validation tools that serve as starting point for developing innovative solutions. Consequently, the first contribution lies in this regard:

Contribution 1.*Emulation/simulation platform for UAVs.*
One of the most prominent challenges that still must be faced is the enormous gap between all these new solutions and their deployment in a real scenario since test fields are challenging (complex, but also quite restrictive due to regulations) and expensive to perform, and simulation alternatives are not appropriate because they have not been

specifically designed for these use cases. Consequently, to address the inherent challenges of multi-UAV system and solutions, an intermediate step between the design and the real validation process is required. In this sense, the main result of this lies in the development of Virtualized Environment for Multi-UAV Network Emulation (VENUE) platform.

- **Victor Sanchez-Aguero**, Francisco Valera, Borja Nogales, Luis F. Gonzalez, and Ivan Vidal. VENUE: Virtualized Environment for multi-UAV network emulation. *Published in IEEE Access.*
- Borja Nogales, Ivan Vidal, **Victor Sanchez-Aguero**, Francisco Valera, Luis F. Gonzalez, and Arturo Azcorra. Automated Deployment of an Internet Protocol Telephony Service on Unmanned Aerial Vehicles Using Network Functions Virtualization. *Published in Journal of Visualized Experiments.*
- Borja Nogales, **Victor Sanchez-Aguero**, Ivan Vidal, Francisco Valera. Adaptable and Automated Small UAV Deployments via Virtualization. Published in *MDPI Sensors,.*
- **Victor Sanchez-Aguero**, Francisco Valera, Ivan Vidal, Borja Nogales, Jaime Cabezas, and Carlos Vidal. A virtualization approach to validate services and subsystems of a MALE UAV. Published in *2022 IEEE INFOCOM Workshops: The 9th International Workshop on Computer and Networking Experimental Research using Testbeds.* **Best paper award**.
- Ivan Vidal, **Victor Sanchez-Aguero**, Francisco Valera, Borja Nogales, Jaime Cabezas, Carlos Vidal, Alicia Lopez, Daniel Gonzalez, Jose Diez, Laura Berrazueta and Manuel Merino. MILANO: una vision futura para un UAS tactico. Published in *VI Congreso Nacional de i+d en Defensa y Seguridad. DESEi+d'18,.*

After developing the VENUE platform and taking advantage of the opportunity provided by the tool to validate the proposals in their most preliminary stages, we decided to go one step further:

Contribution 2.*Integration of innovative technologies in UAV networks.*
The potential environment for service provisioning that multi-UAV systems are showing is a particularly appropriate context for different technologies such as NFV (enabling faster and more flexible deployment of network services), SDN (allowing a more straightforward network configuration and reducing the operational costs), or DDS (enabling efficient information sharing). However, due to the limited resources available in Small Unmanned Aerial Vehicle (SUAV) (e.g., volatile environments, reduced flight range, reduced computational capability), integrating these mentioned technologies is not a straightforward procedure. These technologies are not designed for multi-UAV

environments, and although they provide many advantages, they do not consider the unique characteristics of this environment. Therefore, certain modifications and considerations have to be taken into account before the integration. In this sense, the main contribution of this thesis lies in a complete methodology for the integration of multi-UAV networks and different technologies.

- Ivan Vidal, Paolo Bellavista, **Victor Sanchez-Aguero**, Jaime Garcia-Reinoso, Francisco Valera, Borja Nogales and Arturo Azcorra. Enabling Multi-Mission Interoperable UAS Using Data-Centric Communications. *Published in MDPI Sensors.*

- **Victor Sanchez-Aguero**, Ivan Vidal, Francisco Valera, Borja Nogales, Luciano Leonel Mendes, Wheberth Damascena Dias, and Alexandre Carvalho Ferreira. Deploying an NFV-Based Experimentation Scenario for 5G Solutions in Underserved Areas. *Published in MDPI Sensors.*

With the experience of integrating different 5G technologies with UAVs and performing additional experimentation producing several use cases and testbeds, two proposals are provided aiming to solve some of the challenges mentioned above:

Contribution 3. *Energy consumption analysis in multi UAV networks.*
Although the energy management challenge is also present in standalone UAV short missions, its complexity is exponentially exacerbated when dealing with multi-UAV long-term operations. Multi-UAV environments may give rise to long-endurance missions (beyond battery life) that require uninterrupted service provisioning that is not achievable using a single UAV due to battery capacity constraints. In this line, this thesis contributes with an extensive analysis of the energy consumption problem and proposes a solution based on UAV replacements to extend the life of the multi-UAV network.

- Christian Tipantuña, Xavier Hesselbach, **Victor Sanchez-Aguero**, Francisco Valera, Ivan Vidal and Borja Nogales. An NFV-Based Energy Scheduling Algorithm for a 5G Enabled Fleet of Programmable Unmanned Aerial Vehicles. *Published in Wireless Communications and Mobile Computing.*

- **Victor Sanchez-Aguero**, Francisco Valera, Ivan Vidal, Christian Tipantuña, Xavier Hesselbach. Energy-aware management in Multi-UAV deployments: Modeling and Strategies. *Published in MDPI Sensors.*

Contribution 4. *Technology integration and UAV Traffic Management.*
Nowadays, UAVs cannot fly at low altitudes in most urban environments because UAV control technologies have not reached the necessary level of sophistication. This

thesis incorporates all the previously investigated technologies to create/produce robust communication systems to enable UAV applications in civilian environments. On the other hand, a UAV Traffic Management system is necessary to guarantee airspace safety. In this line, the main objective of the European project Labyrinth (where we collaborate) is to develop a complete UAV management system. This thesis contributes to the design of the communications architecture and the integration of the different communication technologies to produce a complete, secure, and reliable UAV management system

- **Victor Sanchez-Aguero**, Luis F. Gonzalez, Francisco Valera, Ivan Vidal, Rafael A. López da Silva. Cellular and virtualization technologies for UAVs: an experimental perspective. *Published in MDPI Sensors.*

- **Victor Sanchez-Aguero**, Miguel-Angel Fas-Millan, Francisco Valera, Ivan Vidal, Alejandro Paniagua-Tineo, Rafael A. Lopez da Silva, and Jose Manuel Manjon. *Published in 2021 IEEE Globecom Workshops: Workshop on Cellular UAV and Satellite Communications*

- **Victor Sanchez-Aguero**, Francisco Valera, Ivan Vidal, Borja Nogales, A Daniel Gonzalez del Río. Software Defined RPA Networks. *Submited in IEEE Communication Magazine.*

1.2.2. Software released as Open Source

Some of the results of the work carried out during the thesis have been published as open-source software. These results serve not only to replicate experiments accomplished in the articles but also to provide the community with tools for research. Table 1.1 shows the software released as open-source along with this thesis.

Table 1.1: Software released as Open Source.

Name	URL	Chapter
VENUE	https://github.com/vsaguero/VENUE	2
energyUAV	https://github.com/vsaguero/energyUAV	6
ansible parrot	https://github.com/vsaguero/pyparrot_scripts	6
Interface Manager	https://github.com/vsaguero/labyrinth	6, 8
UAV network monitoring	https://github.com/vsaguero/labyrinth	8

1.3. Outline of the thesis

The rest of the thesis is organized in different parts and chapters detailing the contributions aforementioned in the previous subsection:

Part II includes Chapters 2 and 3. The VENUE emulation platform is presented in detail in Chapter 2. Chapter 3 presents a virtualization platform to validate services and subsystems of a Tactical/Medium Altitude Long Endurance (MALE) UAV from the Spanish Ministry of Defense.

Part III includes Chapters 4 and 5: Chapter 4 presents a deployment that combines an NFV testbed including small UAVs, with a 5G access network created to serve remote areas. Chapter 5 presents a practical experimentation with the commercial 4G/Long Term Evolution (LTE) network to enable Beyond Radio Line-of-Sight (BRLoS) operations.

Part IV includes from Chapter 6 to Chapter 8: Chapter 6 presents the practical UAV replacement problem. Chapter 7 presents a multi-interface network framework for UAV management and data communication. Chapter 8 presents an evolution of the last chapter achievements towards an SDN paradigm.

Finally, part V includes Chapter 9 which concludes the thesis and presents future directions. Additionally, two appendices are also included, which contain information that complements the chapters.

PART II

CONTRIBUTIONS TO UAV COMMUNICATIONS VALIDATION

This part presents a proposal for one of the most common problems that the research and development community has to face at some stage: the validation of the different solutions and deployments. In this research area, there is currently a notorious gap between the design phase and the deployment phase since well-known network simulators are not designed with the constraints imposed by Unmanned Aerial Vehicle (UAV) in mind. Besides, service implementations (that are usually distributed into Single Board Computer (SBC) carried as payloads by UAVs) cannot be easily combined with the simulators.

Chapter 2 presents Virtualized Environment for Multi-UAV Network Emulation (VENUE). VENUE is an experimentation platform that allows testing the integration of multi-UAV Flying Adhoc Network (FANET) together with network services deployments. VENUE covers from the simulation/emulation phase up to the real equipment integration phase. The validation of the platform is also presented in this part through several UAV use cases that make use of Network Function Virtualization (NFV) technologies.

Additionally, Chapter 3 presents a virtualization approach to validate services and subsystems of a Medium Altitude Long Endurance (MALE) UAV. This approach is designed to facilitate the experimentation and development of new Unmanned Aircraft System (UAS) components/services and support the testing, validation, and maintenance stages.

2

VENUE: Virtualized environment for multi-UAV network emulation

Multi-Unmanned Aerial Vehicle (UAV) systems do present some challenges that must be solved for suitable performance. Some of these essential design challenges are associated with wireless communications because of the particular characteristics of multi-UAV networks [44]. These particularities are: *(i)* the mobility degree and mobility pattern (different from popular mobile wireless networks) since it can lead to intermittent connectivity states; *(ii)* the dynamic network topology (it changes depending on the target mission) because device to device communications must be maintained despite the changes in the topology; *(iii)* Small Unmanned Aerial Vehicle (SUAV) networks may include different types of sensors (requiring for diverse data delivery strategy, e.g., network paths, priorities).

In this scenario, one of the most prominent challenges that still must be faced is the enormous existing gap between all these new solutions and their deployment in real scenarios since field tests are difficult (complex but also quite restrictive due to regulations) and expensive to perform, and simulation alternatives are not appropriate because they have not been specifically designed for these use cases. Consequently, to address the inherent challenges of multi-UAV systems and solutions, an intermediate step between the design and the real validation process is required.

Taking into account the aforementioned considerations, this chapter presents the Virtualized Environment for Multi-UAV Network Emulation (VENUE) open-source validation platform for multi-UAV and Flying Adhoc Network (FANET) scenarios where different services based on Fifth Generation of cellular network technology (5G) programmable UAVs can be deployed and tested. This solution is built on top of Linux Containers (LXC) and the ns-3 network simulator (based on [45]) and provides an emulation framework that allows the integration of different Virtual Network Function (VNF) (or virtual entities in general), which can be later used into real SUAV hardware, together with a network simulator (used to emulate the specific characteristics of wireless channels).

Besides, the framework enables real hardware (that can be on-boarded into SUAVs as payload) to be directly integrated with the simulation environment, in order not only to test the performance of applications and developments but also to test the correct operation in the hardware that will host the developments in the real world.

In conclusion, this framework facilitates the prototyping and validation processes of multi-UAV services and provides an ecosystem to test the developments that will later be used on real infrastructures (in particular, UAV equipment).

The rest of the chapter is organized as follows: in section 2.1, the related work and background are reviewed. Section 2.3 details the development of the network emulator. Section 2.4.1 presents a use case where the platform is used to evaluate routing protocols for FANETs. Section 2.4.2 presents a Network Function Virtualization (NFV) use case where a multi-UAV network is utilized to increase the programmable network resources over a delimited geographic area, enabling the communications between two groups of users. Finally, section 2.5 concludes the chapter.

2.1. Related work and background

2.1.1. Network simulators

Addressing these aforementioned challenges and many others (e.g., to evaluate the performance of routing algorithms, to examine and to validate mobility models, or to understand the behavior of each node in the network [46]) is a hard and expensive process because it is difficult to verify the different proposals in a real (flying) environment. On this basis, multi-UAV network solutions are commonly trialed using network simulators. The research community provides multiple alternatives for network simulation and experimental validation, such as the Network Simulator 2 (ns-2)[1], the Network Simulator 3 (ns-3)[2], OMNET++[3], or Mininet[4]. Ns-2 provides substantial support for TCP simulation, routing, and multicast protocols over wired and wireless (local and satellite) networks. Ns-3 is also an open-source discrete-event network simulator which primarily targets for research and educational use. OMNET++ is an extensible, modular, component-based C++ simulation library and framework mainly intended to build specific network simulators. Mininet is used to create realistic virtual networks, running real operating system kernels, switches, and implemented application code, on a single machine. Although there is a remarkable interest in these validation platforms, and a large community of users following and evolving them, in certain particular scenarios, like the ones enabled by multi-UAV networks, it is difficult to have realistic results using these well-known network simulators because more specific modules are still

---

[1] ns-2, accessed Tuesday 15th November, 2022. Available: https://www.isi.edu/nsnam/ns/

[2] ns-3, accessed Tuesday 15th November, 2022. Available: https://www.nsnam.org/

[3] OMNeT++ Discrete Event Simulator, accessed Tuesday 15th November, 2022. Available: https://omnetpp.org/

[4] Mininet, accessed Tuesday 15th November, 2022. Available: http://mininet.org/

required. For example, beyond the evaluation of the communication channel that may be reasonably modeled by existing simulators, the analysis of new services and applications that run in the UAV equipment, e.g., novel FANET routing protocols, innovative sensor information distribution protocols, or new 5G softwarization technologies like Software Defined Networking (SDN) or NFV, requires a considerable improvement in current simulators to be able to run new service logics on top of them. Moreover, there are still different areas where additional developments would significantly improve the application of simulators for UAVs. For instance: to enable the interaction of real hardware with the simulated nodes, to add realistic energy consumption models based on measurements, the integration of the simulated network with a 5G core, etc.

As a result, in the state of the art, there are numerous examples of network simulator expansions to satisfy particular situations (even for issues not related to UAVs). NEMAN [47] extends ns-2 to allow running a virtual wireless network of hundreds of nodes on a single end-user machine. Dockemu [48] extends ns-3 to provide a flexible system to rapidly create networks (wired or wireless), incorporating the latest developments and a user-friendly method of installation and configuration. TapRouter [49] presents an application-emulating framework for Mobile Adhoc Network (MANET) with high performance and usability by integrating the ns-3 and lightweight virtualization technology.

There are in fact specific multi-UAV simulation platforms but they are mainly focused on flight operations (e.g., landing, refueling, mobility) or focused on applications on top of the multi-UAV system that are not related with network communication scenarios. RotorS [50] is a modular Micro Aerial Vehicle framework based on Gazebo [51] targeted to tackle higher level tasks, such as collision avoidance, path planning, and vision based problems, like Simultaneous Localization and Mapping (SLAM). OpenUAV [52] is a development to avoid the high barrier imposed by the use of flight UAV simulations due to the need for powerful computers and the time required for initial set up. There are also several examples of papers focused on flight tasks [53].

Some other works that do consider UAV communications are, in general, evaluating their performance directly using general purpose network simulators [54] [55] [56], assuming significant simplifications, e.g., outdated mobility models, limited simulation times, or limited traffic patterns.

2.1.2. FANETs

On the other hand, it should be noted that one of the most relevant fields where VENUE can be applied are FANETs. Motivated by the emerging development of SUAVs, the FANET [44] concept has been increasingly attracting the attention of both military and civil environments because of its simplicity, versatility, flexibility, and usability. FANETs are ad hoc networks built by aerial mobile nodes, and one of their most remarkable advantages is to be able to provide feasible Device to Device (D2D) wireless communications between network nodes without any need for additional infrastructure, like traditional MANET or Vehicular Adhoc Network (VANET). Previous research in the area of MANETs and VANETs has indeed served as FANETs starting point, although it cannot be directly applied because MANETs and VANETs have been designed for devices with limited speed or restricted mobility patterns, and multi-UAV networks are in general, very dynamic in nature. Therefore, it is required to study FANETs as a new network family due to their different requirements regarding Quality of Service (QoS), mobility models, or data delivery [46].

Before a multi-UAV system can provide stable and reliable services, there are diverse challenges to be considered and one of the most relevant ones is related to the number of nodes and links that coexist and cooperate in the network. UAVs do regularly change their position (at least for battery replacement), and these movements do obviously modify the network topology and so the connections created among the nodes. A simple change in the location of a UAV that happens to be connecting different parts of the network may lead to a topology partition and service performance degradation. The same may happen when a UAV has to be replaced (because of battery exhaustion, for instance) forcing the new incoming UAV to have proactive/reactive restoration mechanisms to be properly configured.

2.1.3. VENUE contributions

VENUE allows testing a wide variety of scenarios including protocols, services, and technologies embedded in the on-board computer of the UAVs. This includes FANETs communication technologies based on standard protocols such as Optimized Link State Routing (OLSR), Ad hoc On-demand Distance Vector (AODV) or Better Approach To Mobile Ad hoc Networking (B.A.T.M.A.N), or based on new trends such as SDN. This contribution also includes the possibility of testing from traditional services and applications such as voice over Internet Protocol (IP) or video streaming, to innovative services based on virtualization or 5G technologies such as NFV.

Another VENUE significant contribution is the possibility to interact with real hardware that is frequently used as a UAV payload. The real payload allows not only to test the development functionality but also to test if the hardware has adequate/enough resources. Modifications had to be made to the ns-3 source code to allow the interaction on different entities (virtualized or real hardware), as this functionality is not entirely supported.

The platform also enables to test scenarios that require network nodes mobility. In this respect, it supports Mission Planned Based (MPB) mobility pattern [57], i.e., predetermined trajectory information, which is usually planned in advance. This way, UAV mobility patterns can be specified by the platform user following a predetermined format described in [58]. The platform supports the installation of a predefined mobility pattern for each of the UAVs in the emulation. Therefore, it opens the possibility to use any of the mobility models defined in [59]. Each UAV follows MPB information with realistic flight traces.

Finally, the platform incorporates a number of pre-created modules that enable scenarios with the utilization of different routing protocols (e.g., OLSR, SDN based routing solutions), as well as a set of supporting tools to test user applications and developments. These modules can be flexibly incorporated into user-defined test scenarios, as they have been implemented using virtualization containers.

2.2. Life without VENUE

This section exemplifies our pursued approach to validate multi-UAV communications before VENUE platform development. In particular, it describes the experiment performed in [7] to evaluate whether multi-UAV communications are achievable using multi-UAV technologies (Wi-Fi in our case) and whether communication paths between the different UAVs are enabled utilizing well-known protocols in other areas such as MANETs and VANETs: OLSR and AODV. The obtained results highlight that this strategy can be used, but at the same time, highlight all the restrictions and limitations of traditional network simulators in such an environment.

2.2.1. Scenario definition

To validate the usage of both MANET protocols (OLSR and AODV), we considered the simulation scenario depicted in Figure 2.1, highlighting that the included UAV cloud platform represents a FANET. In this scenario, 16 UAVs placed in a grid with a constant position with a separation of 70 m (maximum distance reached in the simulation implementation with a reasonable link quality which covers an area bigger than 78,400 m2) compose the FANET with the aim of providing a network service during 1 h (3600 s). With respect to the UAVs batteries, we assumed that those are mainly limited by flight energy consumption since other energy components demanded by the network services (e.g., Access Point (AP), DCHP or Routing functionalities being executed by the UAV) are not comparable with the energy required by flight operations. For this simulation, we established a service time of around 20 min (1200 s time between failures) for each UAV (battery lifetime) based on the technical specifications of the DJI Phantom 3 model[5]. In a second test, we reduced the time between failures to 5 min to get insight into the behavior of the selected protocols in extreme (unrealistic) conditions. Since battery life can be affected by different factors such as environmental conditions, maximum charge or flight conditions, we used a uniform distribution to model it characterized by the range

---

[5] DJI Phantom 3, accessed Tuesday 15th November, 2022. Available: https://www.dji.com/es/phantom-3-pro

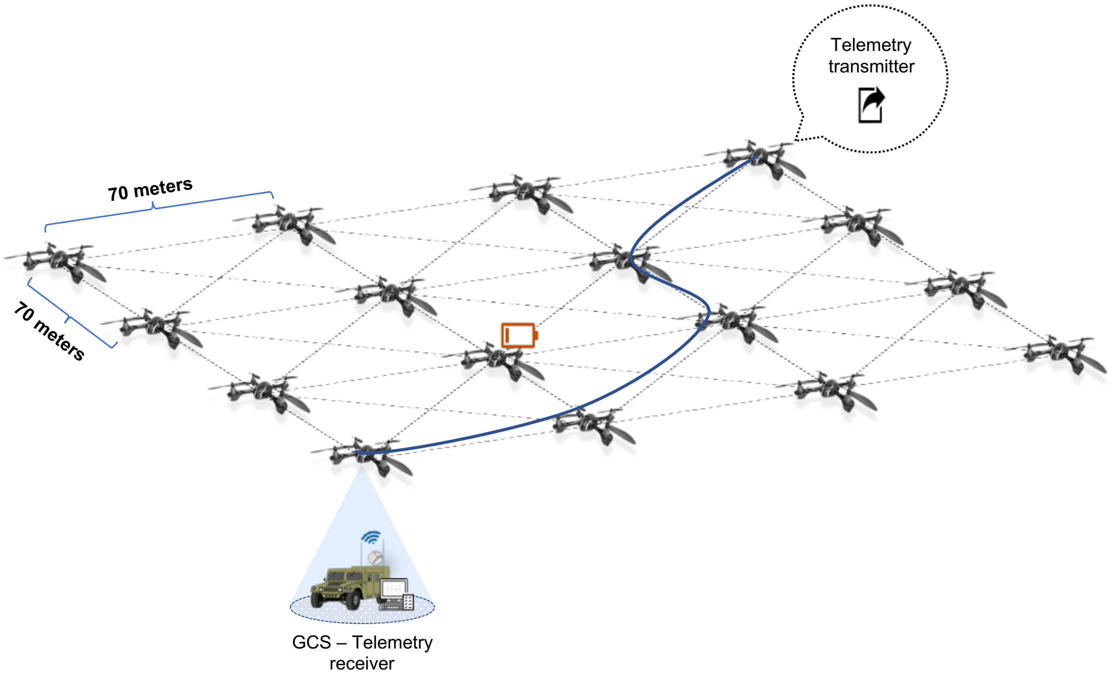


Figure 2.1: Simulation scenario

$[a = 0, b = 40]$, which provides an unfavorable dispersion of values with an average of 20 min. In a typical mission, when an UAV battery is under a certain threshold, the UAV will fly back to the Ground Control Station (GCS) to replace it and another UAV will come onto the scene to provide the service of the replaced UAV. Normally, this replacement time is quite low since battery failures can be predicted from the GCS and the new UAV can be sent to perform the replacement in advance. For that reason, we used an exponential distribution $\lambda = 1/3$, which gives an average replacement time of 3 s to model these low replacement times. Figure 2.2 shows an example of fail and replacement events for one of the performed simulations (note that the SUAV acting as AP and the one transmitting the telemetry information are not failing during the simulation since they are assumed to be statically positioned and the only power consumption is due to the network function provisioning).

To perform the simulations on the wireless ad hoc networks with different mobile nodes, we used ns-3 (introduced in section 2.1.1). Specific existing modules for ns-3 are used to simulate the FANET protocols. Node movements have been calculated in advance in order to approximate them as much as possible to a real UAV application. These individual movement traces are later installed on each node to perform the simulation. The network interfaces available in the different UAV are common IEEE 802.11b interfaces.

To measure the network performance, we set an *Iperf* flow between two edge nodes

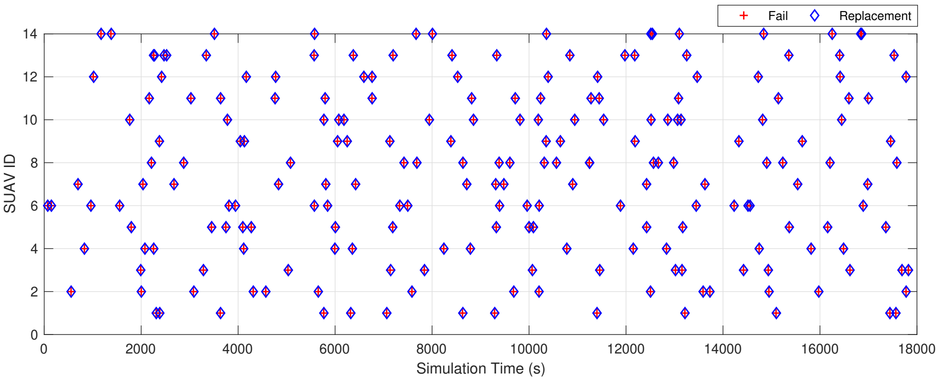


Figure 2.2: Simulation scenario

(User Datagram Protocol (UDP) flow with a rate of 32 Kb/s) emulating the telemetry information sent by one of the UAV (more details about telemetry information can be found in [4]). All the simulation details can be found in Table 2.1.

Table 2.1: Simulation parameters.

Parameters	Value
Traffic	Constant Bit Rate
Transmission Rate	32 Kb/s
Network Protocol	UDP
Simulation Time	18,000 s
Number of UAVs	16
Mobility Model	Static
Failures	Uniform Distribution $a = 0$, $b = 40$
Replacements	Exponential Distribution $\lambda = 1/3$
Simulation Area	280 m × 280 m

2.2.2. Simulation results

To compare the different protocols, we defined three relevant metrics: *(i)* the convergence time needed to find a path between two nodes (characterized in fact by the average total throughput, meaning that the less throughput achieved, the longer it takes to converge), *(ii)* the overhead that is generated by the routing protocol (overall total control traffic) and *(iii)* the optimal path usage measured in term of the percentage of packets that are sent through the shortest path. All the configurable parameters and timers for

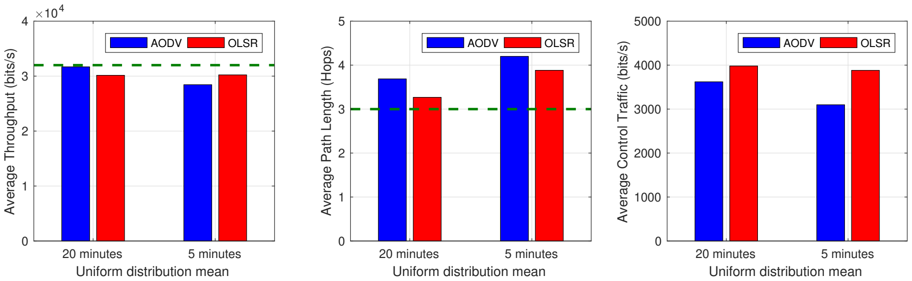


Figure 2.3: Simulation results

AODV and OLSR are based on the protocol default values. After ten simulations (results are the average of all the experiments), we can conclude that both protocols may be a suitable alternative for this scenario. Convergence times, represented by the throughput, are quite similar (slightly better in AODV with failures every 20 min, although OLSR performs better when failures are increased, as shown in Figure 2.3a). OLSR normally selects shorter paths, as illustrated in Figure 2.3b, both for the realistic scenario and the worst case scenario. In fact, this is quite reasonable since OLSR, as reactive routing protocol, is converging to the optimal path when it is available, as opposed to AODV that maintain the selected path until a failure takes place. As expected, the overhead introduced by AODV (Figure 2.3c) is smaller than the overhead introduced by OLSR, however, both are negligible compared with the regular service throughput. However, these results, even promising, are too poor to guarantee that real applications would operate properly since they are limited by different factors such as mobility patterns, the actual application of routing protocols (not only in simulation environments), or the application of developments and technologies that are not well/still supported by simulators. These reasons have driven the development of VENUE (explained in section 2.3).

2.3. VENUE design and implementation

VENUE has specifically been designed to satisfy different aspects that are not fully supported in current network validation solutions (simulators/emulators) highlighted in section 2.2, and that is important to be able to validate current FANET related technologies.

As it is detailed in this section, the most relevant requirements that have guided VENUE design are *(i)* The necessity to provide mobility models to accommodate real UAV applications and that are undoubtedly important since they define the FANET physical topology evolution during the service time (VENUE can introduce two-dimensional (x,y axes) mobility patterns in a predefined altitude (z-axis)). Second, *(ii)* the necessity not only to simulate services but also to be able to emulate them, testing the real applications that can later be migrated into real hardware. VENUE can serve as an emulator of the physical channel linking real applications and processing real packets (the platform can model and select the communication channels and technologies between the different participants of the FANET). This feature also enables VENUE to serve as a validation platform until almost the final integration with the hardware devices where the services will be installed (in fact this hardware can also be directly attached to the platform to validate this phase too as it will be seen). Finally, *(iii)* the necessity to support 5G softwarization technologies, such as NFV and SDN [60] [61] [62]. These technologies are being introduced into the UAVs field as innovative alternatives to be able to support flexible (and agile) service provisioning (with NFV) but also as a possibility to evolve current ad hoc networks routing protocols adapting them to the particularities of FANETs (with SDN). VENUE improves the support provided in ns-3 to be able to integrate multiple external nodes into ad hoc networks emulating the communication channel. These external nodes can incorporate the corresponding network functions into the system and test them all together.

The following items present VENUE main characteristics and summarize its main strengths listing different advantages and functionalities:

- The framework operates with real applications, e.g., LXC, Virtural Machine (VM), real hardware, that can be directly used in real infrastructure afterward. This functionality allows reducing prototyping and validation cycles and reducing the time-to-market or time-to-operation period.

- The framework is suitable to emulate both wired and wireless (infrastructure-based and ad hoc) networks. Since this implementation is designed to trial FANET scenarios, all the examples provided in the framework are based on the 802.11 Wi-Fi technology. Still, with small effort, this platform may also assist in wired networks scenarios.

- The framework is suitable for both IPv4 and IPv6.

- The platform enables more than one real host (virtual or physical) to interact with the network emulation because of the source code modifications (available in the patch file [58]). This functionality is crucial since there are usually numerous participants in a FANET.

- The framework incorporates pre-created LXCs that include the installation and configuration of different FANET routing protocols, such as the OLSR [63], SDN technologies (like RYU [64] controller and OVS[6]), and network analysis tools (*iPerf*[7]

---

[6] Open vSwitch, accessed Tuesday 15th November, 2022. Available: https://www.openvswitch.org/

[7] GUEANT, iPerf, accessed Tuesday 15th November, 2022. Available: https://iperf.fr/

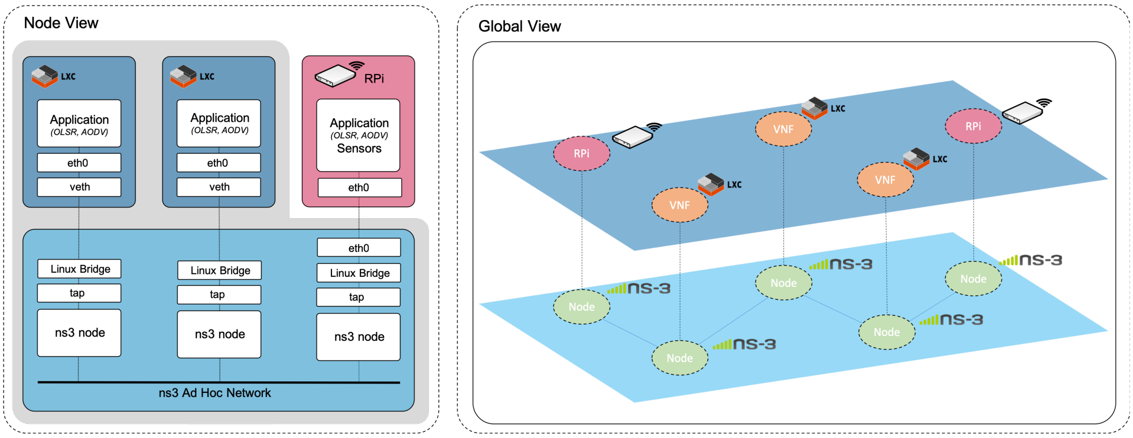


Figure 2.4: Framework architecture, node and global view

and *Trafic*[8]), to facilitate testing multi-UAV systems applications and developments.

- Ns-3 simulator is installed and configured inside a LXC. Thus, the emulator can be used as a standard VNF, with all the advantages it brings (the platform can be instantiated by any Virtual Infrastructure Manager (VIM) such as OpenStack and is portable to any Linux machine without any specific configuration).

- Ns-3 generates standard network traffic traces[9] that assist the process of code debugging and traffic analysis. The results can be studied using regular tools like *Wireshark*[10], which is utilized for network troubleshooting, analysis and also allows us to analyze the traffic that passes through a network and thus is able to solve or even prevent possible problems that may arise.

- By default, the framework incorporates the MPB mobility model [65]. The platform user can introduce real flight traces (VENUE works with two dimensional traces, but the analysis is considered to be correct since landing and takeoff are made far from the network service area and in the network service area in most situations it is reasonable to consider the UAVs flying in the same plane) following a predefined format to enable realistic multi-UAV mission.

- The platform incorporates a set of scripts that configure the emulation environment for a smooth development and prototyping process. This functionality allows the user to create complex multi-UAV networks in a simple way enabling advanced experiments to be carried out.

---

[8] 5GinFIRE trafic, accessed Tuesday 15th November, 2022. Available: https://github.com/5GinFIRE/trafic

[9] TCPDUMP/LIBPCAP, accessed Tuesday 15th November, 2022. Available: https://www.tcpdump.org/

[10] Wireshark, accessed Tuesday 15th November, 2022. Available: https://www.wireshark.org/

2.3.1. Framework architecture

Figure 2.4 summarizes the architecture of VENUE whose components are described in the following subsection. The global view of the system (complete architecture) is shown in the right part of the picture. The emulation layer is in charge of creating and modeling the FANET, and it also emulates UAV mobility. The top layer represents the real nodes (virtual entities and general-purpose hardware). Those nodes contain the developments and applications that enable the multi-UAV scenario (routing protocols, network services, data collection/transmission, etc.). The integration of the two layers not only allows linking real nodes through an emulated FANET but also provides each UAV with mobility. In the left part of the figure (node view), the connection between the ns-3 nodes and the real nodes is highlighted. Moreover, it includes all the required components to make the association possible such as Linux Bridges (software used to join two or more networks that behave like a virtual network switch) or TAP interfaces (network interface entirely supported in software) [45].

To implement the emulation layer the ns-3 network simulator has been selected. Ns-3 is a discrete event-based network simulator frequently used in the investigation of ad hoc mobile networks. It implements a wide variety of routing protocols from both wired and wireless networks. Ns-3 also allows the configuration of several network parameters; meanwhile, it provides extensive data collection modules for exhaustive analysis. However, the main reason to choose ns-3 is that in addition to its simulation capabilities it implements a module that allows network emulation, i.e., this module allows external entities (real or virtual) to interact with the ns-3 environment. The emulation module provides a fully real-time controllable and reproducible environment where the routing protocols and application developments can later be used in UAV equipment without modifications. This functionality introduces a notorious added value as compared to pure ns-3 simulations since the ns-3 developments cannot be used in real equipment.

In FANETs, the number of participant nodes may be high, and in order to be able to validate these scenarios beyond the integration of real devices (to test real hardware inside a FANET with numerous participants), VENUE must be able to handle inputs from multiple virtual devices allowing that way to provide a scalable solution. Thus, the experimentation is not only limited by the availability of physical devices (which in these cases are usually expensive) but also by the selected virtualization technology that must be as lightweight as possible aiming to run several singular applications (each of the SUAVs inside the multi-UAV system) on a single server. For this purpose, the LXC[11] have been selected. However, LXC is the tool that VENUE provides to ease prototyping; the developed application could then be used on any virtualization platform or directly on the physical UAV device. In Figure 2.4 (Node view), it can be seen that all the required

---

[11] Linux Containers, accessed Tuesday 15th November, 2022. Available: https://linuxcontainers.org/

applications to enable communications between the UAVs (e.g., OLSR, SDN) are in the LXC (or in real hardware). This feature allows the tested applications to be portable to commodity equipment.

A LXC is a set of processes that are detached from the rest of the Operating System (OS). Unlike standard VM, LXC share the kernel with the OS and separate the application from the rest of the system. LXC is portable and modular, including in the production stage (for instance, the same LXC can run in different hosts using an NFV platform). These characteristics make the prototyping and development process faster in comparison with traditional test environments. Nevertheless, the containers must be compatible with the underlying operating system (because LXC share the kernel). The selected hypervisor to manage the LXC is the Linux Daemon (LXD)[12] because of its simplicity. LXD adds new possibilities and functionalities compared to the conventional system container management of LXC, e.g., container migration or physical devices passthrough.

These LXC have proven to be usable in UAV regular payload equipment (see our previous work in [15] [4] where the design of the solution is based on NFV and lightweight VNF).

Although one of the most significant strength of the platform is the possibility of virtualization, this framework also allows the interaction with real hardware, allowing VENUE users to get an insight about the behavior of real hardware during a mission. Typically, UAV embedded equipment is reduced in size and limited in both computing capacity and battery. Therefore, although the developed application may have the correct functionality, it cannot be guaranteed that the hardware in charge of its execution will have enough resources. Thanks to this integration, the limitations of the multi-UAVs payload hardware can be estimated.

To integrate real hosts (including either real hardware, Raspberry Pi (RPi) in Figure 2.4, or virtual hosts) into the ns-3 emulation, it is necessary to use the TapBridge Model[13]. This module allows the replacement of specific nodes (previously determined by the user) from the ns-3 network by real hosts. The TapBridge Model overwrites the ns-3-device MAC address by the overlying real-host (virtual entity or UAV payload) MAC address. After this association, the real-host considers the ns-3 net device as a local device and the TapBridge Model sends all the ns-3 node incoming network traffic through a virtual TAP interface (which is connected to the LXC or the UAV equipment through a Linux Bridge as it is shown in Figure 2.4). Similarly, the Tap Bridge Model sends all the outgoing traffic (virtual entity or UAV payload) through the emulated ad hoc network. Thus, real devices can communicate with each other using the underlying network created by the

---

[12] Linux Containers - LXD, accessed Tuesday 15th November, 2022. Available: https://linuxcontainers.org/lxd/

[13] Tap NetDevice, accessed Tuesday 15th November, 2022. Available: https://www.nsnam.org/docs/release/3.14/models/html/tap.html

ns-3, as it can be seen in Figure 2.4.

TapBridge uses an existing TAP interface previously created and configured by the user. Nonetheless, in VENUE, the process of creating and configuring the system environment has been automated and the user only needs to provide some input parameters, according to the experimentation scenario. More details can be found in [45] and [58].

However, the default TapBridge device presents problems to perform more than one "<ns-3 host>" MAC association in wireless networks (to connect more than one real device to the emulated network). For VENUE it has been required to apply some modifications to the source code of the ns-3 to make the connection between several real hosts and several ns-3 nodes. These bugs have been reported and are under revision. However, in the meantime, a patch file with the corrections is provided to apply those changes in [58].

2.4. Life with VENUE

This section describes some of the experiences carried out after the platform development. These experiments validate the routing protocols that enable the paths between UAVs (subsection 2.4.1) and test whether virtualization technologies such as NFV function in environments not only with limited resources but also volatile and intermittent connectivity (subsection 2.4.2). In addition, subsection 2.4.3 enumerates the use of VENUE in other research articles. Moreover, VENUE has also been used in other thesis chapters to assist developments validation.

2.4.1. Use case I: Routing protocols for FANETs

The first use case presents a simple FANET scenario in order to show how it is possible to validate the integration of real software (virtual entities in this particular case) with the VENUE platform, which is in charge of emulating not only the communication channel (Wi-Fi) but also the mobility pattern of each UAV. More specifically, in this testbed VENUE is used to evaluate routing protocols in a FANET scenario. Following this methodology and using similar metrics to the ones proposed in this scenario, VENUE users can, for instance, select the most suitable routing solution for their own FANET scenario/mission or identify at a glance the most relevant UAVs inside the FANET from the communications perspective.

The goal of this chapter, and of this use case in particular, is not to evaluate a specific routing algorithm but to show the platform for an exhaustive evaluation. Therefore, the same analysis can be replicated with any other routing protocol selected by the user of the VENUE. For this use case we have selected OLSR as an example.

2.4.1.1. Scenario Motivation

Multi-UAV systems may present heterogeneous mobility patterns, involving from a slow-changing topology to a dynamic and fast-changing topology. In fact, it is common to find challenges such as nodes with a high mobility rate (depending on the mission nature), damaged links, or battery constraints. Consequently, changes in the network topology are frequent in comparison to current mobile wireless networks. Moreover, environmental factors can significantly affect a FANET. Meteorological agents, e.g., wind gusts, rainfalls, high/low temperatures, resemble essential in the proper functioning of the system and cannot be accurately predicted before the beginning of the mission. Thus, the routing protocols algorithms require to go beyond the needs of usual MANETs and VANETs, and in consequence, the proposed solution must be flexible enough to allow different levels of dynamism.

Developing an autonomous and cooperative FANET requires reliable and robust communications between UAVs, which must collaborate to efficiently accomplish missions. To determine and configure the potential multi-hop network paths across the UAV swarm, a routing/forwarding algorithm is needed. In these scenarios, data from each UAV can be sent through a possible connection with external infrastructure, that may act as a relay node, such as a GCS or a Satellite link. Long-range communications facilities, e.g., Satellite or Line-of-Sight radio, imply heavy payloads and consequently flight restrictions and battery consumption. Another alternative is to directly send data through the FANET and eventually use a GCS to communicate towards other UAVs or ground infrastructure beyond the FANET.

2.4.1.2. Scenario description

This scenario represents a service provided by a seven SUAV fleet that is used to enable communications, e.g., Voice over IP (VoIP) calls, video broadcasting, 5G network access, when the conventional cellular network is not available or is insufficient, e.g., emergencies, massified events. In this kind of FANET missions, SUAVs usually do have a fixed position and in fact, are perched on land wherever possible to save battery.

Topological changes in these scenarios are typically generated because of battery constraints that imply SUAVs replacement. In this scenario, we consider that when a SUAV battery is under a determined threshold, the SUAV flies back to the GCS to charge/substitute batteries while another SUAV comes onto the swarm to provide the service of the replaced SUAV. This phenomenon affects the performance of the FANET because the rest of the devices should be updated to continue providing the network service. For this purpose, SUAVs must incorporate autonomously reconfigurable routing solutions that collect information about the current status of the FANET and configure the network paths autonomously.

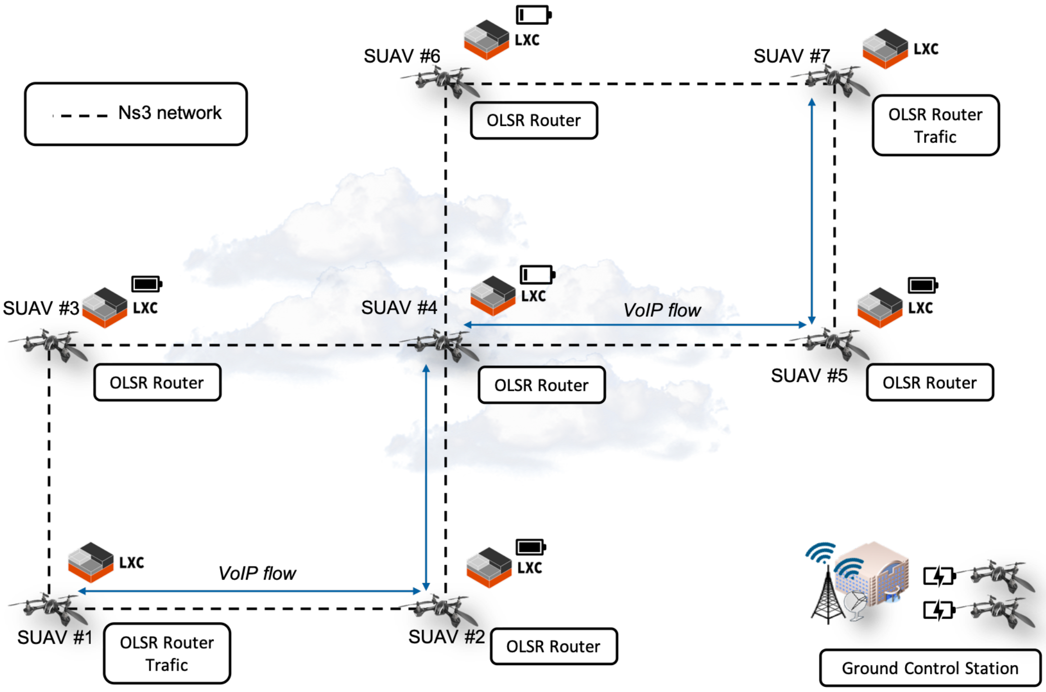


Figure 2.5: Scenario I: Routing protocols for FANETs.

The SUAVs are placed in a grid (50 meters between each SUAV) with a static position (hoovering) as represented in Figure 2.5 forming a FANET. Each SUAV is represented with a LXC using the VENUE platform. To measure the network performance, the source (SUAV #1) and destination (SUAV #7) LXC have *iPerf* installed. *iPerf* is a tool intended for active measurements which reports the bandwidth, loss, and other parameters. A traffic flow is sent from the source to the destination representing a VoIP call (64 Kbits/s of data rate and 126 bytes of packet length). The established ad hoc Wi-Fi network follows the 802.11a standard.

It has been selected a simple use case (in this case with only 7 UAVs) that allows understanding the functionality of VENUE without complicating the configuration and description of the scenario itself. The decision of placing the UAVs on a grid is intended to provide UAVs with more than one path to reach the possible destination. However, there is a central node through which all communications must pass, to illustrate in the analysis that the design of the network topology can significantly affect the service performance.

Typically, a SUAV battery has an autonomy of around 20 flight minutes (in similar cases where the SUAVs are landed to save battery, the lifetime of each device is longer). Taking into account the response time of regular routing protocols this replacement time is enough so as to allow a proper operation of the network (as it is shown in [15]), so in order to see how VENUE can operate, in this experiment we will increase the failure frequency to an extent that the FANET routing will be really disturbed by these failures (and consequently data moving through the FANET would also be disturbed). We have selected (unreal) battery lifetimes from 10 to 400 seconds, to force several changes per minute. The main objective of this chapter motivates the selected replacement values (despite being unreasonable values beforehand in a real scenario). The primary purpose is to show how VENUE can be used, how to understand the results, and to propose a possible analysis). The battery lifetime is modeled for each UAV following a uniform distribution with a variation of ± 5 seconds ($U(a, b), a = mean - 5, b = mean + 5$).

Moreover, the initial battery status does not necessarily have to be the same, so it is possible to model also and we have set for this example a random offset (random value between *[0 mean]*) that has been included to model this phenomenon.

More information on these type of scenario is detailed in our previous work [15] [4] [5].

The source and destination nodes are assumed to be on the ground, and their battery will not be consumed during the experiment so that there is a continuous connection established between the source and the destination and the routing protocol can be evaluated. For simplicity, the replacement period for each SUAV (i.e., the time needed for a UAV in the GCS to provide the service of the replaced UAV in the scenario) is fixed to 10 seconds (i.e., the mission planner usually coordinates UAV replacements), although VENUE allows any pattern to be included, and this value may vary for instance depending on the followed strategy from the control station or the size of the service provider's fleet, i.e., the number of available UAVs in the system.

The routing protocol used in this experiment as an example is OLSR [63], which is one of the most popular and well-known protocols in MANETs. Flying nodes store the updated list of destinations and the routes to them, and OLSR provides mechanisms to periodically refresh the routing tables and to maintain the network topology information.

Olsrd[14] implementation has been selected to install the routing protocol in the LXC. Olsrd is designed to be run as a standalone server process. All the configurable parameters are based on the standard values [63]. Each experiment has been repeated 30 times to obtain proper results.

2.4.1.3. Metrics

To evaluate the routing protocol, we characterize three relevant metrics. The first metric is the *(i)* packet loss that allows obtaining an idea of the convergence time required to find a path between the Source and Destination nodes, i.e., the more packets lost, the more time needed to update network paths. Another analyzed metric is the *(ii)* OLSR control traffic. This metric determines the overhead generated because of the routing protocol signaling, which may be critical in limited bandwidth systems. The third metric is the *(iii)* percentage of use of each SUAV. This metric reveals the significance

---

[14] OLSRd, accessed Tuesday 15th November, 2022. Available: http://www.olsr.org/mediawiki/index.php/Main_Page

of each UAV (depending on its position in the network). This metric may assist to the mission planner to get an insight about which UAVs are relevant, and in which UAVs the battery consumption can be higher, for example. Also, this section presents some network parameters that are important to identify what type of multimedia services can be deployed over the multi-UAV network such as the jitter, the end-to-end delay, and the maximum available bandwidth. All these parameters can be easily obtained from the network traces generated by ns-3 and the network tool that can be integrated into VENUE (e.g., *iPerf* in this experiment).

2.4.1.4. Results and conclusions

Before the previous metrics measurements are presented, and in order to better understand the results, Figure 2.6 shows a snapshot of the tests where it is possible to appreciate for a single experiment the moment in which the replacements of the SUAVs take place (bottom), and the corresponding received throughput at the destination SUAV (up). When a SUAV in the source-destination path is replaced, a consequent drop in the received throughput at the destination occurs. The figure reveals that the impact of the substitutions in the throughput depends on the replaced SUAV. When the replaced one is SUAV #4 (which connects the two parts of the network), the drop remains at least for 10 seconds, i.e., the replacement time. However, with the replacement of other SUAVs, the drop may remain for a smaller period (or not), depending on the time it takes the routing protocol to find another path. In a simple scenario such as the one proposed, it is not complex to identify which SUAVs have the most significant impact on the network service. However, in populated systems, this analysis helps to detect which are the SUAVs with an essential role in the FANET.

Figure 2.7 illustrates the percentage of packets lost in the system depending on the battery lifetime. To represent this data (corresponding to the 30 experiments), we have selected the boxplots. Through the boxplot, the center and dispersion of the data distribution can be easily perceived. The red line matches the median value of the data. The blue box (interquartile range) represents 50% of the data (from 75% to 25% of the values). Finally, the whiskers correspond to the rest of the values of the group. As expected, the more frequent the replacement is, the higher the packet loss percentage remains. Significantly, the value of the packet loss is always over 50% with replacement times below 50 seconds. The replacements are indeed taken to the extreme since the UAVs battery-lifetime is way much longer in standard conditions. However, it can be concluded that for environments with severe changes, i.e., either due to very high mobility or failures in the links, the OLSR protocol may not perform correctly, while it performs quite acceptable with replacements over 100 seconds. This examination may be useful for scenarios where different routing protocols are considered assisting the mission planner in selecting the most suitable option to carry out the service efficiently.

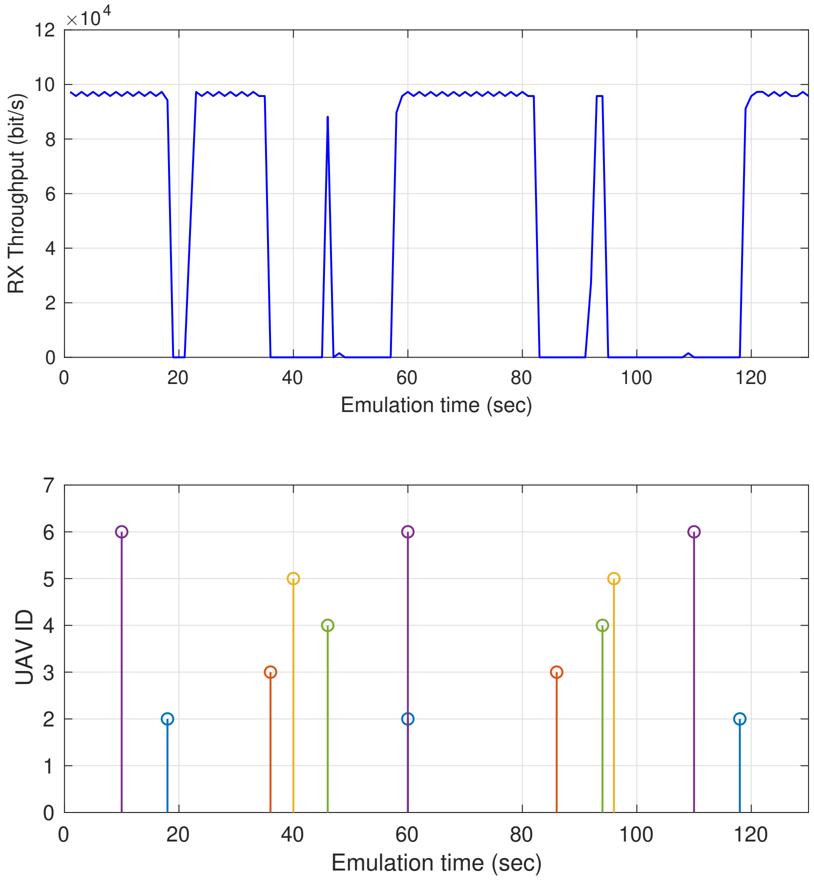


Figure 2.6: Received throughput at destination UAV in one of the experiments (fails each 50 seconds).

Finally, Figure 2.8 presents the percentage of utilization of all the SUAVs defined as the data packets that pass through a SUAV divided by the total number of packets generated by the source. In the graph, it can be appreciated that SUAV #1 has 100% of utilization since it is the source that generates the traffic flow. It can also be seen that SUAV #4 has more usage than the rest of SUAVs because it connects the two parts of the multi-UAV network and all the packets (except for the ones that are lost) go through it. This information, in combination with the knowledge obtained from Figure 2.6, is essential for the mission planner to be able to detect if a particular SUAV has a prominent role in the network and also to check if the system has been correctly designed.

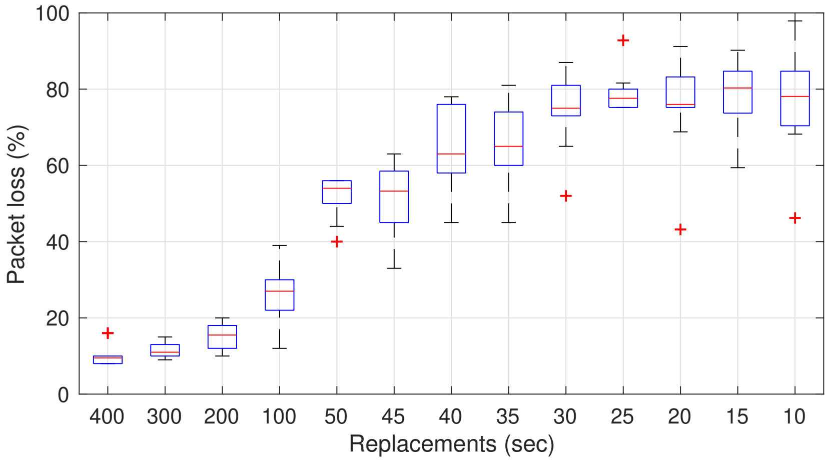
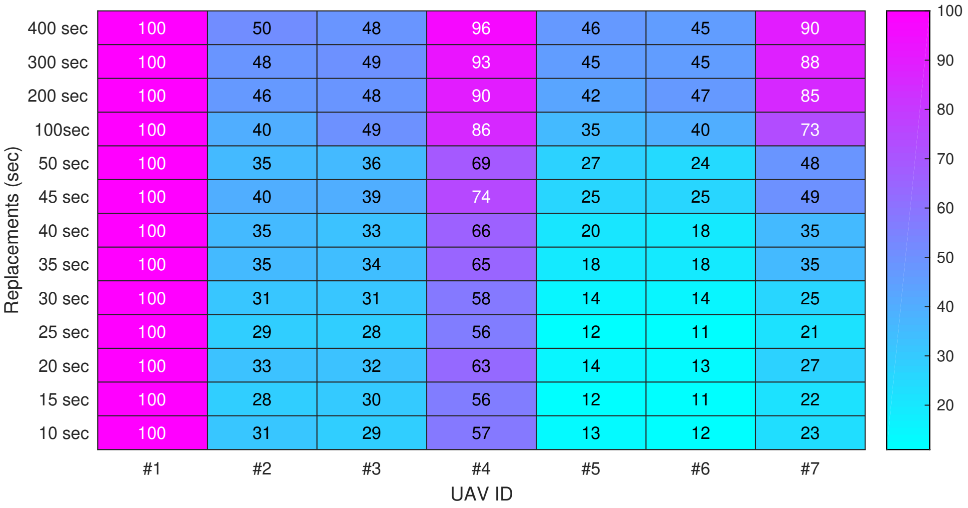


Figure 2.7: Packet loss percentage in the first scenario for different replacements values.



Replacements (sec)	#1	#2	#3	#4	#5	#6	#7
400 sec	100	50	48	96	46	45	90
300 sec	100	48	49	93	45	45	88
200 sec	100	46	48	90	42	47	85
100sec	100	40	49	86	35	40	73
50 sec	100	35	36	69	27	24	48
45 sec	100	40	39	74	25	25	49
40 sec	100	35	33	66	20	18	35
35 sec	100	35	34	65	18	18	35
30 sec	100	31	31	58	14	14	25
25 sec	100	29	28	56	12	11	21
20 sec	100	33	32	63	14	13	27
15 sec	100	28	30	56	12	11	22
10 sec	100	31	29	57	13	12	23

Figure 2.8: Percentage of UAV Wi-Fi utilization.

Other values that can help to determine what type of services can be deployed over the FANET network such as the average jitter (3 ms in our test scenario), the average delay from the source node to the destination node (24 ms), or the average control traffic throughput, i.e., OLSR routing protocol traffic sent by each node (5 Kbit/s) are also provided.

2.4.2. Use case II: FANET to extend 5G connectivity

The second use case is an example of how VENUE can serve to validate a much more complicated scenario: an NFV platform that uses a FANET to allow a smooth deployment of a VoIP service (including, for instance, the automatic deployment of virtualized Session Initiation Protocol (SIP) servers, Domain Name System (DNS) servers, or OLSR routers using OpenStack). Also, the VNFs have been instantiated into real hardware that is ready to be included as SUAV payload (RPi). It must be clarified that a Management and Orchestration (MANO) system was not used to configure the VNFs in order to simplify the experiment explanation. However, detailed information about this configuration can be found in our previous work [4] [15].

In this experiment, VENUE will be used to emulate a scenario closer to the real flight conditions. In this scenario, it is not only expected that the VNFs can interact with each other or that the whole service or the FANET itself can be adequately established. All these things are assumed to have been tested in advance with regular trials. The main goal is to verify the scenario under the changing conditions that there will be when SUAVs take off. However, it will be possible to appreciate in the FANET the effects of the SUAVs movement, the loss of network connectivity implications in the routing protocols. It is particularly relevant (because these previous things can already be done with some existing simulators) to see the effect of this intermittent connectivity in the real deployed VNFs, in the real services under execution and in the NFV orchestrator that is managing the whole service.

2.4.2.1. Scenario Motivation

Nowadays, one of the new technologies that the 5G networks are promoting are the softwarization alternatives like the NFV that allows to quickly deploy different network services on different devices, automatically orchestrating the distribution of virtual entities (VNF) through the network on the proper hardware.

Multi-UAV systems are being proposed as candidate alternatives to serve as computation nodes for these VNFs due to their flexibility to physically deploy the services wherever they are required: crowded events (e.g., concerts, demonstrations) where the conventional base station is overloaded or is not powerful enough, in emergencies, where connectivity is critical to ease emergency tasks or the physical infrastructure, does not exist, to support city areas with malfunctioning base stations, in search and rescue operations in remote areas (mountain, sea), etc. However, resource constraints that are inherent to SUAVs or their payloads (small Single Board Computer (SBC)) have as a consequence the appearance of new challenges (for FANET to provide a stable and reliable service) that have to be solved before their correct deployment and operation. For this deployment, we used the prototype NFV system developed in our previous work [15] [4].

2.4.2.2. Scenario Description

This scenario includes a three SUAVs fleet and a GCS intended to be used, for instance, to enable communications in emergencies as it can be seen in Figure 2.9. The service has been instantiated using a set of virtual functions and virtual networks that operate on top of the FANET to provide a flexible and dynamic connectivity backbone and service deployment.

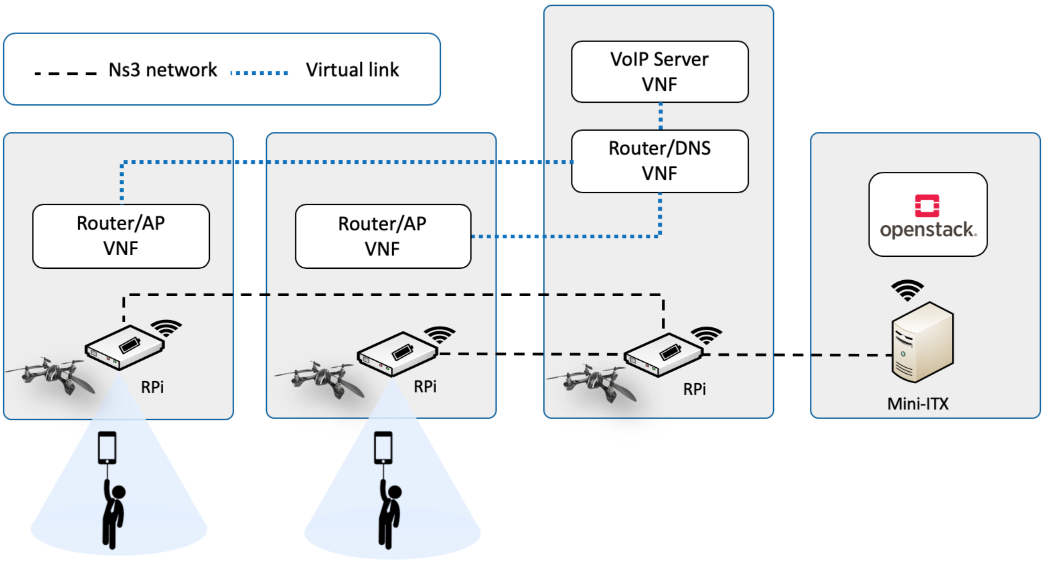


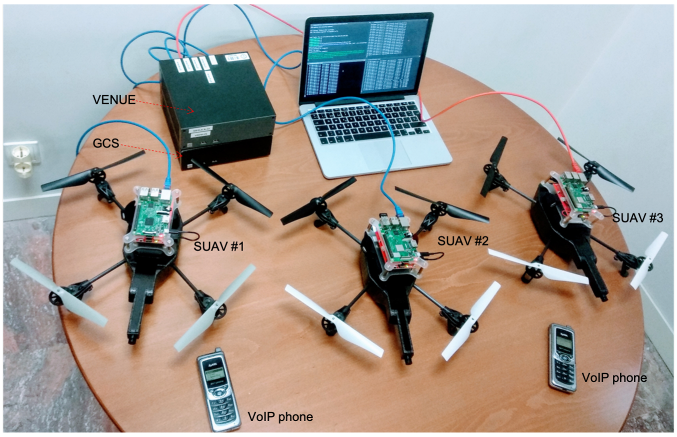
Figure 2.9: Scenario II: multi-UAV network to extend 5G connectivity.

Each SUAV will carry as payload a RPi 3B[15], SBC. All the RPis include an external battery-power supply (3.7 V and 3,800 mAh) so that, the network service operation does not affect the SUAV battery itself (which is intensively required by the SUAV engines). The selected hardware is not a random choice. The small size of the RPi (85.60 mm×56 mm×21 mm), in combination with its reduced weight, allows almost any commercial UAV, e.g., DJI Phantom 4, to fly loading these devices without any problem. In Figure 2.10(b), it can be seen how the payload has been incorporated into the aircraft. In Figure 2.10(c), it can be appreciated the UAV flying with the payload. The GCS is a mini-ITX computer (Intel Core i7 2.3 GHz, 16GB RAM, 128GB SSD, 4 GbE ports) that also acts as a cloud operating system (OpenStack [66]). This equipment can be appreciated in Figure 2.10(a). As it is shown in Figure 2.9, the physical network topology is emulated using VENUE (VENUE emulates the realistic conditions of the wireless ecosystem), and on top of the network, a virtual network service has been deployed. Two SUAVs (#1 and #2) also provide real Wi-Fi access points enabling end-users to utilize the network. The created (emulated by VENUE) Wi-Fi network follows the 802.11a standard. The created (real) access points follow the 802.11n standard. VENUE provides mobility to the commodity equipment as can be appreciated in the following section figures. Otherwise, making these scheduled replacements would not be possible.

In order to create a network service, different VNFs have been used: *(i)* two VNFs

---

[15] RPi, accessed Tuesday 15th November, 2022. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/



(a) Scenario II Testbed: SUAVs, Single Board Computers, GCS, VoIP terminals and VENUE emulation platform

(b) DJI Phantom 4 with RPI and battery as payload

(c) DJI Phantom 4 flying with RPI and battery as payload

Figure 2.10: Real hardware used in the experiments.

implement the router functionality, *(ii)* another VNF implements a router and also includes a DNS service and finally, *(iii)* another one implements a VoIP server based on the SIP [67] server (Kamailio[16]), to allow the ground users to "register" wireless terminals in the VoIP server and maintain telephone conversations with other users. On the other hand, two APs have been configured (including a Dynamic Host Configuration Protocol (DHCP) server) that allow users to connect to the network deployed by the SUAVs.

All these VNFs are instantiated and configured using OpenStack. The ground control station and the devices hosting the deployed VNFs use OLSR to enable the communications. Remarkably, the virtual networks are deployed using the Virtual Extensible Local Area Network (VXLAN) [68].

A complete VoIP call (including the signaling process to start the call) has been performed to test the whole network service. The ZyXEL Prestige 2000W terminals have been utilized to make the call. Besides, a video stream is also sent through the network

---

[16] Kamailio SIP Server, accessed Tuesday 15th November, 2022. Available: https://www.kamailio.org/w/

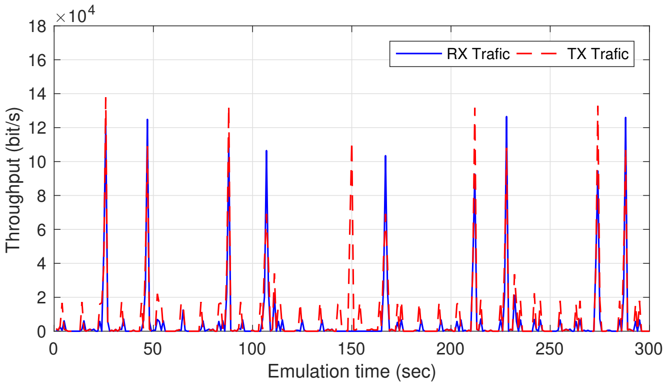


Figure 2.11: OpenStack Control traffic in a Compute Node

using the VLC[17] tool. Finally, during the mission, SUAV #2 is replaced to force an additional topology change.

2.4.2.3. Results and conclusions

Network traffic is captured using the *Wireshark* tool to analyze this scenario. Captures are performed in SUAV #3 both in the ad hoc network interface (emulated network by the VENUE platform) and in the AP interface (real), in order to analyze all the traffic generated by the service.

Figure 2.11 shows the control traffic between the OpenStack controller and SUAV #3 compute node necessary to manage the computing, storage, and networking resources of the NFV platform. As a result of the replacement of SUAV #2, the figure reveals a drop in the received traffic while the computer node keeps on sending traffic to the controller steadily.

Similarly, OLSR signaling traffic can also be appreciated in Figure 2.12. Likewise, during SUAV #2 replacement, there is a stop in the received traffic. OLSR average throughput is around 3 Kbit/s (which is negligible as compared to multimedia services).

Figure 2.13 illustrates SIP signaling traffic to start the multimedia VoIP call. Moreover, the DNS traffic is also represented in the figure. The DNS is required to resolve the name of the SIP server and register the VoIP terminals. Finally, Figure 2.14 reflects the voice traffic received by one of the wireless phones. The call was made without errors and with appropriate sound quality. Figure 2.14 also shows the received video traffic at the destination. As it can be seen, after SUAV #2 replacement, the two services are correctly recovered.

---

[17] VLC media player, accessed Tuesday 15th November, 2022. Available: https://www.videolan.org/vlc/index.es.html

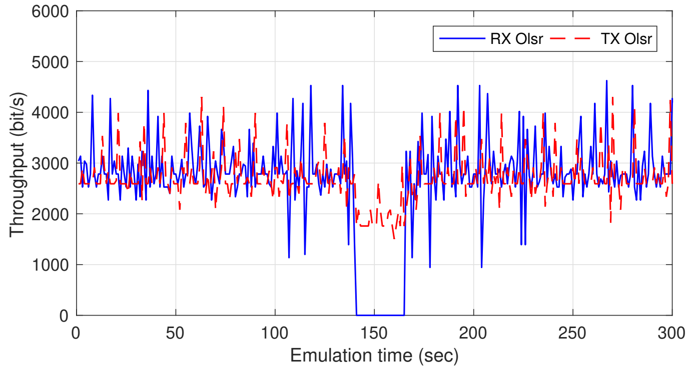


Figure 2.12: OLSR signaling traffic at the Compute Node.

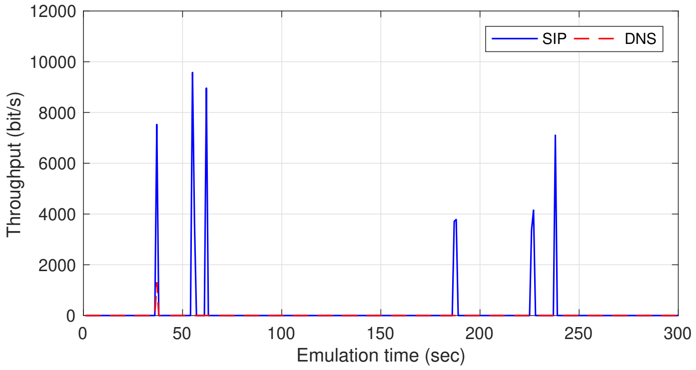


Figure 2.13: SIP and DNS traffic needed for the VoIP call.

2.4.3. Use case III: VENUE in other research articles

The VENUE platform has been employed in further research articles during the course of the Ph.D. In particular, VENUE is used to deploy a virtual platform representing the communication system of the MILANO Unmanned Aircraft System (UAS) from the Spanish Ministry of Defense. More details about this article can be found in Chapter 3. VENUE has also been used to emulate UAVs' Wi-Fi channel and mobility patterns in the articles (Chapters 7 and 8). Additional details regarding these articles can be found in Chapter 7. VENUE has also been used for other articles not included in the thesis [7].

2.5. Discussion

This chapter has presented VENUE: an emulation platform for FANETs that enables the validation of different network service deployments in multi-UAV systems. VENUE provides a controllable and reproducible environment that allows experimenters to extract multiple conclusive and reliable results.

VENUE is based on the ns-3 simulation software and LXC light-weight VNFs.

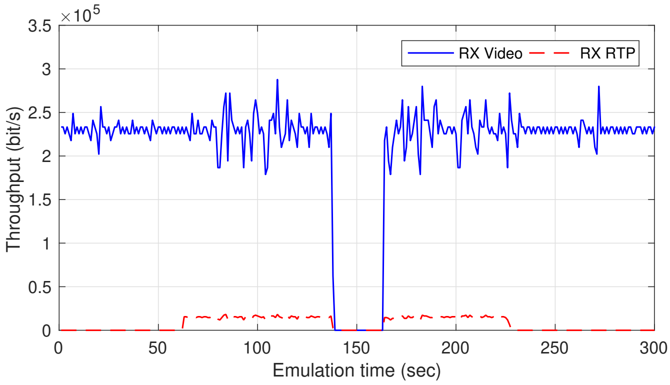


Figure 2.14: Received Video and Audio at the destination.

Furthermore, the platform allows the VNFs to interact with general-purpose UAV equipment. However, for this aim, the source code of ns-3 needs to be modified (a patch file including the corrections is also provided in [58]). This functionality allows several benefits, such as to add realistic energy consumption models based on measurements or the integration of the simulated network with a 5G core.

As a differentiation factor in comparison to similar developments, VENUE framework includes the ns-3 simulator as a VNF which enables modular prototyping; it also incorporates pre-created LXC already configured with FANET routing protocols (OLSR and SDN) to promote the use of the platform. Finally, mobility has been incorporated into the network nodes to simulate a plausibility multi-device environment.

To conclude, the chapter validates VENUE functionalities in two different use cases. The first use case verifies the integration of real software (virtual entities in this case) with VENUE platform and evaluates FANET routing protocols. The second use case validates a network service functionality deployed on top of real UAV hardware. The results shown in the proposed use cases reveal that VENUE platform is suitable for the prototyping and development of multi-UAV systems, reducing to the gap between the development and production stage.

There are some developments that are expected to be included in VENUE in the near future. The first one is to provide different mobility models to cover multiple UAV applications. By including this functionality, the user of the platform does not need to precompute any mobility pattern, boosting the platform usage. Second, to provide LXC with the preinstallation of diverse routing alternatives, covering from the most popular MANET protocols, e.g., OLSR, AODV, B.A.T.M.A.N, to other innovative solutions applied, particularly to UAVs. Finally, the intention is to include the development of a configurable SDN based extension to allow testing different alternatives to face the routing challenge in FANETs that also takes into account the efficient energy consumption.

As soon as multi-UAV services begin to be widely deployed, flexible FANET modeling platforms like VENUE will be increasingly useful to facilitate all these deployments.

2.6. Research impact/dissemination

The outcome of the work behind this chapter has resulted in two publications. The first publication is in the MDPI *Sensors* journal. MDPI *Sensors* has an impact factor of 3.576, and it is ranked in the Q1 Journal Citation Reports (JCR). The second publication is in the IEEE *Access* journal. IEEE *Access* was included in the 2020 JCR and received an impact factor of 3.667 (Q1). This paper has been presented at an open workshop entitled "Research activities of mutual interest", held at IMDEA Networks Institute, where dissemination activities were carried out among different European Telematic Engineering departments.

Fifty-three research articles have cited both articles. Moreover, this content has been part of Deliverable 5.3 ("Final report on Network-level mechanisms implementation") of the H2020 5GRANGE European project.

3

A virtualization approach to validate services and subsystems of a tactical UAV

This chapter presents a virtualization platform designed to facilitate the experimentation and development of new Unmanned Aircraft System (UAS) components/services and support the testing, validation, and maintenance stages.

The whole development cycle of new applications and services, including the testing, validation, and maintenance process, becomes quite challenging since these new services normally imply complex configurations and verification procedures. Although simulation is a necessary tool to develop systems, it is essential to validate the proposed developments in real equipment. The problem is that validating research developments and novel technologies in flight campaigns with such a large Unmanned Aerial Vehicle (UAV) has an elevated cost in terms of human and economic resources, combined with regulatory restrictions and administrative difficulties such as air space segregation. Besides, flying this type of large UAV for services during the development phase entails considerable risks. Finally, onboarding the whole hardware system is also requiring considerable effort, so it is really important to be able to test everything in advance with conditions that can resemble the real ones as much as possible. Thus, preventing to disassemble the system for further re-configurations.

To face these challenges, this chapter presents a service validation platform (extending the development explained in Chapter 2) for the communication components of the MILANO system[1], a strategic UAS developed by the Spanish National Institute for Aerospace Technology (INTA). This platform is based on virtualization techniques, aiming to serve as a test environment for new communication developments and technologies to be incorporated into the MILANO system. Authors have already faced these challenges before for small UAVs [7] [2]. The goal now is to evolve this experience and apply it in much larger systems such as the MILANO, being able not only to test different communication components like routers (on the aircraft and the ground station) but also

---

[1] RPA, a new technology, accessed Tuesday 15th November, 2022. Available: https://www.inta.es/INTA/en/quienes-somos/historia/los-rpa/

to incorporate physical equipment such as the UAV camera, the data acquisition unit, or the pilot workstation into the experiments. In addition, the Virtualized Environment for Multi-UAV Network Emulation (VENUE) emulation platform (already explained in Chapter 2) has been incorporated to reproduce the wireless communications between the UAV and the Ground Control Station (GCS) and to examine how the different developments behave under different potential conditions, such as loss of connectivity.

The MILANO system can carry out surveillance and observation missions by providing real-time images in the visible and infrared spectrum of the areas of interest flown over. This system is also flexible enough so as to incorporate new payload elements into the UAV. Therefore, by integrating a Synthetic Aperture Radar (SAR), the system could perform surveillance and observation missions in all-weather conditions and above the cloud ceiling. In addition to these previous features and after finishing the first phases of the system development, the focus has also been set on transforming this UAS into a strategic reference for research. In this line, Universidad Carlos III de Madrid and INTA participate in the European project Labyrinth[2]. The technology developed in this chapter will be used in the project, whose main objective is to ensure UAV traffic control and safety in different scenarios such as civil road transport, air transport, emergencies, and waterborne transport.

The rest of the chapter is organized as follows: section 3.1 reviews the main components of the MILANO system. Section 3.2 describes the virtualization platform highlighting its main characteristics, whereas section 3.3 presents the set of experiments defined to validate the utility of the virtualization platform. Finally, section 3.4 outlines the main conclusions of the chapter and depicts some future research lines.

3.1. The MILANO system

MILANO is a strategic Medium Altitude Long Endurance (MALE) aerial system designed to perform reconnaissance, surveillance, and target acquisition missions. The aircraft has a maximum speed of 120 kt (220 km/h), a service ceiling of 28,000 ft (8,500 m), and an endurance of more than 20 hours. The aircraft has 900 kg of maximum take-off weight and 12.5 m of wingspan. Due to all these characteristics, together with its outstanding carriage capability regarding weight (up to 150 kg), volume (up to $0.6\,\mathrm{m}^3$), and electric power (1 kW) available onboard MILANO is also an ideal aerial platform for research and development purposes.

During the flights, the UAV and the integrated payloads are controlled and monitored from the GCS (Figure 3.2). The UAV operators can interrupt the planned (predefined) mission at any moment, commanding the aircraft using the available semi-automatic and semi-manual modes (sending commands with joysticks filtered by the flight control

---

[2] Labyrinth: Ensuring drone traffic control and safety, accessed Tuesday 15th November, 2022. Avilable: http://labyrinth2020.eu/

system). Automatic take-off, landing, and taxi of the vehicle at the airport are also implemented.

The mission is always planned in advance before take-off and validated on the GCS. The validation considers orography, aircraft performance, communications, and weather conditions. Additional tools assisting the operators (pilots) to evaluate potential damages to population and properties on the ground or other airspace users are also available on the GCS (e.g., digital terrain models, population density maps, static and dynamic geofencing, and geo-cage capabilities and aeronautic cartography).

During the mission, the uplink commands (from the GCS to the UAV, typically named telecommand) and the downlink telemetry and payload outputs (from the UAV to the GCS, commonly, telemeasurement of data and images) are provided by both a radio Line of Sight (LoS) communication subsystem (any frequency between 70 MHz and 6 GHz with bandwidths up to 80 Mb/s) and a Beyond Line of Sight (BLoS) satellite communication system. Both subsystems use standard TCP/IP communication protocols, enabling the system to perform multi-UAV and multi-GCS missions. The initial version of this communication system was already trialed and validated several years ago by the authors in a previous INTA UAS, the SIVA (see [69]). This system is now being studied for adaptation in MILANO, which first flight tests took place in 2021.

Figure 3.1: MILANO testing campaign at Rozas Airport

In order to certify the whole system, the NATO Standardized Agreement 4671 (STANAG-4671) has been selected as the reference standard. This standard is intended to allow military UAVs to operate in other NATO airspace. This process requires extensive structural analysis and testing, environmental and electromagnetic compatibility tests. Shreds of evidence of compliance with the stated requirements have already been provided

Figure 3.2: MILANO Ground Control Station

to the certification authorities.

The system is now on the extensive flight testing phase of the development (Figure 3.1) and first field tests took place in 2018 at the Rozas Airborne Research Center (Lugo, Spain)[3]. Flight testing procedures which normally will take a considerable amount of time can in this case be simplified due to the good capability and quality of the telemetry as defined for the air vehicle and by the accuracy and repetition of the manoeuvres performed.

The system shall evolve to allow operations as a network of multiple UAVs and multiple ground control stations since the goal is to extend the area covered and the persistence of the system over the area of interest. The following capabilities are envisioned:

- Definition of the network architecture where UAS of different capabilities and manned assets could also be integrated.
- Mission planning, monitoring, and control capabilities for multiple MILANO UAVs from a single GCS.
- Computer vision for automatic recognition of elements of interest.

---

[3] CIAR, Rozas Airbone Research Center, accessed Tuesday 15th November, 2022. Available: https://www.inta.es/CIAR/es/

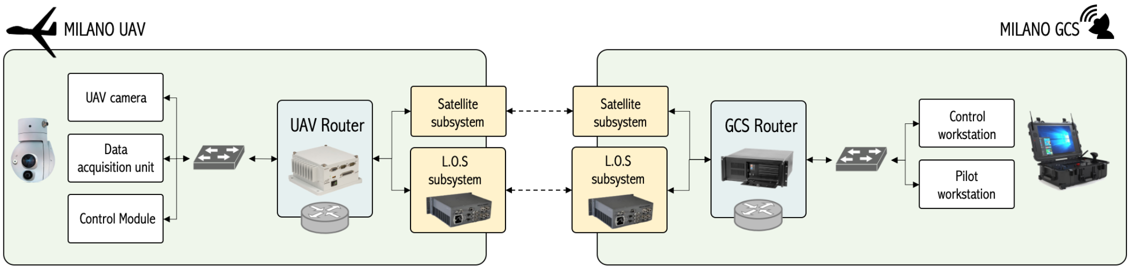


Figure 3.3: Simplified INTA MILANO communication scheme

- Support virtual services running on both the UAV and the GCS that can be deployed on-demand depending on specific mission requirements.
- Enhancement of mission risk evaluation tools available on the ground control station.
- Implementation on the MILANO ground control station of some U-space [70] functions as defined for the H2020 Labyrinth project.
- Fault tolerance functionalities.

3.2. UAS virtualization platform

The communications system design of the MILANO has particular characteristics that require specific network functions. The TCP/IP standard is used to enable multi-UAV and multi-GCS missions so that the communication system requires routing functionalities in both the UAV and the GCS to properly operate with the different IP address spaces. This architecture increases the complexity in configuration and management operations, but it offers significant benefits and opportunities, such as the possibility of traffic isolation, network-layer security, or implementation of different mechanisms. More details can be found in [69].

Figure 3.3 shows the communications scheme of the MILANO system. As previously mentioned, both the UAV and the GCS include their own router to allow internal connections and to exchange information between both domains while keeping the traffic between the networks properly isolated. The connectivity between the aircraft and the control station is provided by a LoS subsystem or via satellite. Different equipment (also displayed in Figure 3.3) in the UAV and the GCS will be taken into account for the performance of the experiments.

To validate this mentioned equipment, we currently find several limitations such as the need to radiate (turn the LoS subsystems on), which is not always allowed in laboratory environments, the difficulty of power supplying the equipment on the UAV that is in charge of communications, or the impossibility of testing specific scenarios if it is not in a real flight campaigns such as intermittent connectivity or low signal levels. Moreover, the communication components (e.g., routers) are already in a production state. Therefore, adding new developments to them without previous experimentation and validation is not an appropriate policy. For these reasons, the goal is to build a virtual platform as similar as possible to the real environment to enable further experimentation and development.

This section describes the evolution of a virtualization platform to support the deployment/development/validation of the communication service components and other types of applications that may be installed on the MILANO system in combination with the emulation of the communication channels and mobility patterns.

To create a virtual environment and provide the platform with the flexibility to adapt it to different UAS, we have chosen to deploy the communication components as Virtural Machine (VM) using the OpenStack software [66]. A VM is an abstraction of a physical machine as a softwarization unit. Then, the VM includes the necessary software that can represent any other device (e.g., a computer, or a router), executing an operating system, and providing the same functionality as a physical device. The goal is then to have a software instance of the physical machines to run all the developments as if it were the real system. In addition, the selection to use VMs instead of other lighter virtualization alternatives (e.g., Linux Containers (LXC), Dockers) is because VMs provide Hardware Abstraction Layer so the potential users of the platform can develop without concern about compatibility with the underlying devices. In addition, the MILANO router is considerably big and includes enough resources to run VMs properly.

OpenStack is an open source project that supports the creation of a cloud scalable environment where resources in terms of compute, network, and storage can be allocated to VMs in the system depending on the requisites. In addition, once installed and properly configured, it holds high autonomy from the point of view of the user to deploy new instances (i.e., VMs with a specific status and configuration). OpenStack not only makes possible to instantiate VMs but also to create custom virtual networks to interconnect those VMs. In this way, we have the VMs running on the compute node (i.e., the OpenStack entity in charge of executing the VMs), but we can also virtually recreate the topology they have in the systems. Moreover, operating with Openstack allows us to work with VMs that could potentially be migrated to different environments. In this particular case, both VMs used for the UAV router and the GCS router will be used in the MILANO physical equipment since it also supports virtualization. The UAV router and the GCS router shown in Figure 3.3 are server computers that allow running not only routing operations (to be executed natively on the host), but also VMs or other virtualization technologies. Due to this, we can assume that the UAS is a cloud platform with compute nodes. These would be used to deploy VMs with any other service/functionality of interest. This advantage allows configuring these devices before their installation on the real supplies, saving the configuration of equipment previously

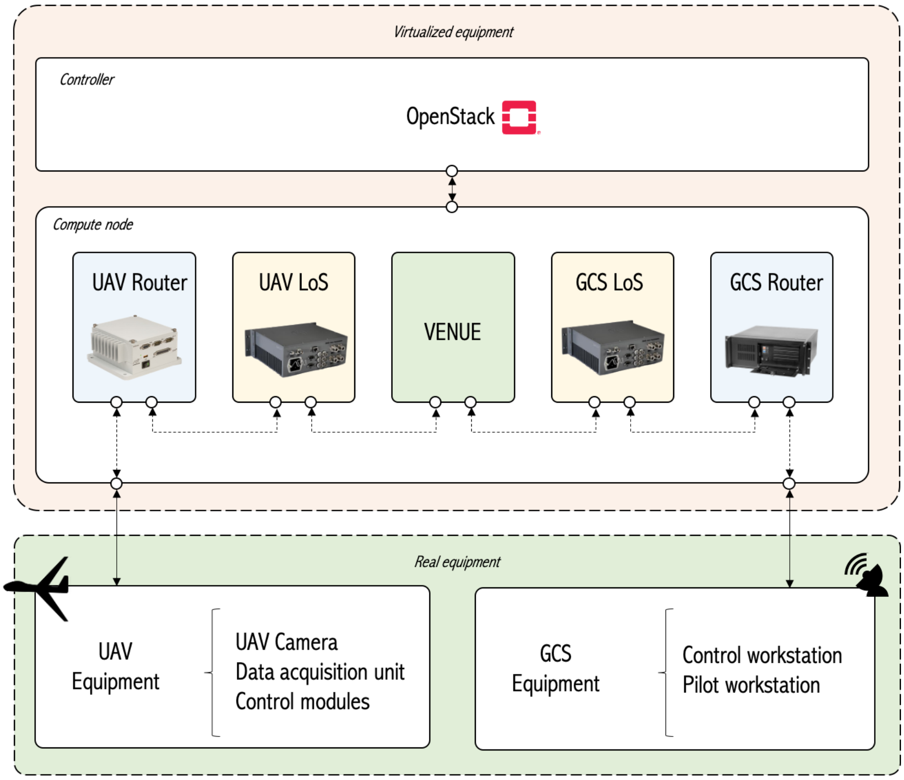


Figure 3.4: Virtual environment for MILANO platform

tested and already in the production stage.

The two entities required to create the platform are displayed in the middle part of Figure 3.4 (virtualization equipment). In the first place, we find the OpenStack controller, which is in charge of managing all the OpenStack components and services such as networking, workload provisioning, or storage, among others. The controller must have connectivity with the compute node where the VMs run. The virtual networks between VMs (themselves) and the interfaces of the compute node are created from the controller. On the other hand, we find the OpenStack compute node, where the VMs run. More details about this configuration can be found in our previous works [4, 19]. In particular, the Openstack version selected for the UAS virtualization platform is Queens [71]. Within this platform, both the OpenStack controller and the compute node are installed on mini-ITX computers (Intel Core i7 2.3 GHz, 16 GB RAM, 128 GB SSD, 4 GbE ports) (middle part of Figure 3.4).

The LoS subsystem deployed within the Virtualized equipment as VMs (see Figure 3.4) have been connected using the VENUE platform (Chapter 2), to provide verisimilitude to the development with the real environment. This development has also been included as a VM and instantiated using the OpenStack. VENUE is an experimentation platform that allows examining the integration of UAS together with network services deployments. VENUE covers the emulation phase, in which all the equipment of the UAS is virtualized, and it also covers the equipment integration stage, allowing the MILANO real equipment to interconnect using the VENUE emulator. In this particular case, the VENUE emulator has been applied to emulate the wireless link between the UAV and the GCS LoS radio cards. This emulation platform enables the selection between different models of wired and wireless links. In the case of MILANO, we have selected a customized wireless link with similar characteristics to the system we intend to validate (10 km range and 80 Mb/s bandwidth).

In addition, VENUE provides each VM (the ones assigned a wireless link) with a predetermined mobility pattern. In this case, the GCS remains static in the same position while the UAV moves are based on the mobility trace captured from a real MILANO mission (based on the field tests carried out at the Rozas Airborne Research Center). This selected mobility pattern allows both elements to be within the range of coverage during the experiments. Following this methodology, it is possible to test how the system would react to different situations such as loss of coverage/connectivity, which can not be validated during real flight campaigns due to the extreme risk.

On the other hand, different real equipment hosted in the UAV and the GCS have been connected to the virtual platform (right and left part of Figure 3.4). The purpose of using this equipment together with the platform is to validate its functionality and correct operation. It should be noted that this equipment includes explicit INTA developments that do not have an open source license and therefore, they cannot be replicated or provided as VMs.

The equipment onboarded into the UAV is the *(i) UAV camera*, which sends the uncompressed video to the *(ii) data acquisition unit*, responsible for encoding and distribution, and the *(iii) control modules*, component that transmits the status of the aircraft sensors and control actuators. Meanwhile, the equipment included in the GCS is the *(iv) control workstation* in charge of monitoring the UAV camera video and the data collected from sensors, and the *(v) pilot workstation* in charge of transferring the required flight instructions to the UAV.

3.3. Experiments

This section presents a set of experiments that have been carried out using the virtualization platform described in section 3.2 in order to explore the possibilities of such a validation system. These experiments cover all the steps from the design/development stage of the communication components, to the final validation with the physical equipment of the MILANO UAS. With the purpose of showcasing the different deployment phases, the same experiment has been replicated over different networking settlements incrementally moving from the virtual validation to the real scenario.

During the first step that corresponds to the design phase, the main communication elements are all deployed as VMs. These components are the UAV router, the GCS router, and the LoS subsystems (this is depicted in the Compute node component of Figure 3.4). The LoS subsystems (i.e., UAV LoS and GCS LoS) have been interconnected using VENUE in order to be able to emulate the MILANO wireless link with a range of 10 km and a bandwidth of 80 Mb/s. Devices that connect to VENUE are assigned a wireless link and a mobility pattern. In this experiment, both LoS equipment are connected to VENUE, representing the UAV and GCS endpoint in each case. The mobility pattern of the UAV is based on the mobility traces of the UAV in a previous INTA mission. However, during these first tests the UAV is always within the connectivity range, so no connectivity loss is expected.

This virtual environment has allowed configuring the equipment for a correct operation, such as assigning to the VMs the IP addresses of the real equipment and adding the required configurations for a correct operation (e.g., IPSec tunnels, IP routes). To verify that everything is working as expected, we have carried out connectivity tests using the Linux *ping* command between all the virtualized equipment.

In the second phase, the virtual environment deployed in the first stage has been completed incorporating MILANO real equipment into the virtual platform, as shown in Figure 3.4 (both sides). This physical equipment corresponds to the *UAV camera*, the *data acquisition unit*, the *control module* (onboarded in the UAV), the *control workstation*, and the *pilot workstation* (located in the GCS).

In order to check the correct operation of the whole system, a network traffic capture was performed using *Wireshark* (a network protocol analyzer) at the GCS router to monitor data packets that are crossing a specific point in the network. As it can be appreciated in Figure 3.5, the GCS router receives three types of traffic flows from the UAV and transmits two traffic flows to the UAV generated by the workstations. The first flow (depicted with a blue line) corresponds to the high quality video stream produced by the *UAV camera* (average traffic of 3.05 Mb/s). This video is sent from the *UAV camera* to the *Data acquisition unit*, responsible for its compression and final transmission to the control workstation. It should be noted that the y-axis has a logarithmic scale so that all the displayed traffic streams can be appreciated correctly due to the difference in size between them. The x-axis corresponds to the time on a linear scale.

The second and third flows (purple and green lines) correspond with data from sensors onboarded in the UAV (average traffic of 68.41 kb/s), and data of the current status of the UAV actuators (average traffic of 22.58 kb/s). Data from sensors (e.g., GPS, altitude, temperature, pressure, accelerometer) are also collected by the *data acquisition unit*, which is responsible for its transmission, while the status of the UAV actuators is directly sent from the *control module*.

At the same time, the GCS sends two types of flows to the UAV, which correspond to the remote control of the camera (representedted with yellow line), which is controlled

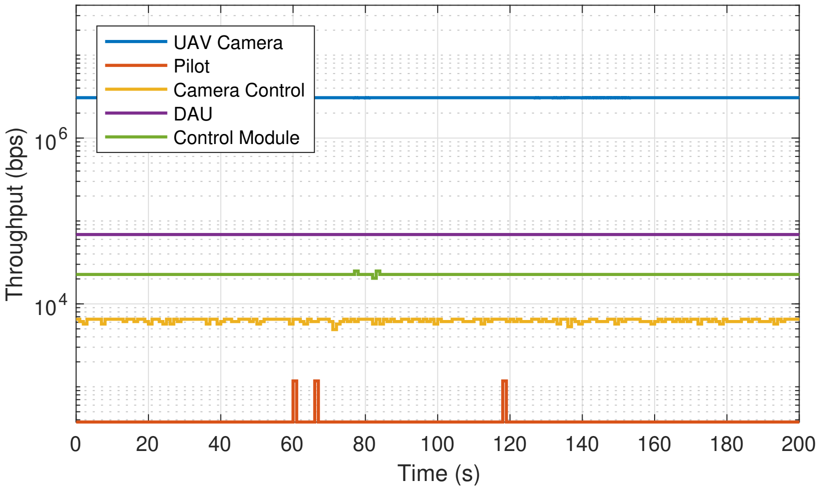


Figure 3.5: Traffic captured on the GCS Virtual Machine through the virtualization platform

from a joystick on the GCS (lateral movements, zoom) and to the remote control of the pilot (red line). As it can be seen, the camera control traffic keeps constant (average traffic of 6.32 kb/s) since, by default, a heartbeat pulse is sent to the camera. However, the pilot traffic is only displayed in the graph (average traffic of 381 b/s) when an instruction is manually sent from the station through the joystick. The correct operation has been verified by experienced INTA operators on-site; moreover, the traffic pattern received (Figure 3.5) corresponds to that expected, and there is no loss of information.

Once the design has been validated in the virtual platform, there are two options to transfer it to the real system: *(i)* import into the real platform the VMs used in the virtual platform (obviously allowing for exactly the same functionality), or *(ii)* transfer the configurations tested in the virtual platform to the real system. This procedure mainly depends on whether the UAV and GCS equipment support virtualization and their available resources are sufficient. In this particular chapter, it is the configuration that has been transferred to the MILANO equipment.

The same test were repeated, but this time using all the physical MILANO system equipment, including the communication hardware components. As shown in the first part of Figure 3.6 (first 100 seconds), and as expected, the traffic captured during the second experiment is similar to the traffic of the previous experiment (Figure 3.5) performed using the virtual machines. This demonstrates how the platform can easily be used to further validate any forthcoming improvement/development that may be included within the communications system before its costly integration into the real system. In addition, the mobility pattern (resembling a real flight) incorporated into the UAV through the use of VENUE results in a communication loss (UAV out of the radio frequency range) with the GCS for approximately one minute to verify one of the multiple experimentation

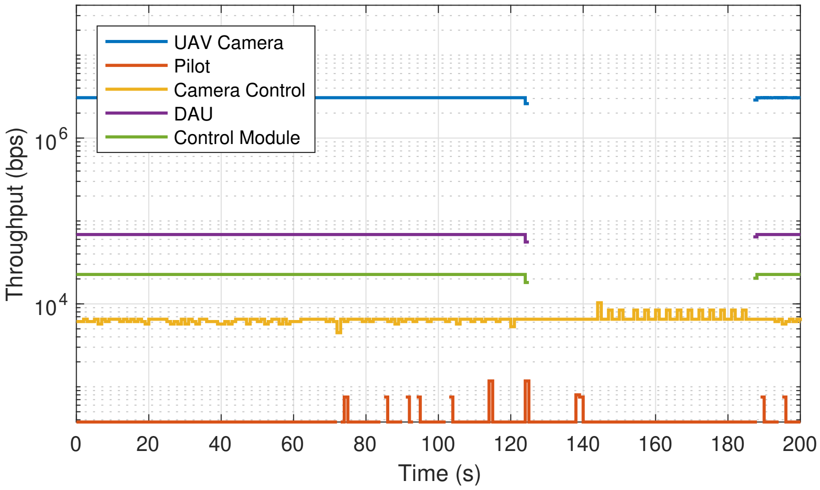


Figure 3.6: Traffic captured on the GCS with connectivity loss introduced by VENUE

options it enables. As shown in Figure 3.6, communications coming from the UAV are dropped during that period (since the displayed traffic is captured at the GCS). This test allows checking real equipment behaviour in this type of situations where channel fading may lead to a transmission technology swapping: LoS subsystem may be changed by satellite communications when signal quality is reduced for instance.

3.4. Discussion

This chapter presents a virtualization platform that has been used to validate the communication system of a strategic UAS of the Spanish Ministry of Defense, the MILANO, an all-weather surveillance, and observation system. In a near future MILANO will be developed to serve as an experimentation platform and evolved as a benchmark for international research. In addition, since the MILANO system supports the virtualization technology, the presented platform produces a test environment for some real components of the UAS, which could be executed as VMs. Thus, components may be developed as VMs and tested on the platform system before bringing them to the MILANO.

The main communication components of the MILANO have been virtualized. These virtualized components have been compared with the onboarded physical equipment and the hardware available in the GCS with essentially identical results, showing how the platform can efficiently validate improvement/development that may be included in the communications system before its costly integration into the real system.

This work opens future lines of research in the framework of the European project Labyrinth. Some examples of these research lines are to incorporate new communication alternatives beyond the line of sight, such as 5G cellular networks. Another example is to include the virtualization of functions and services on top of the UAV system, providing the opportunity to the UAV operators to deploy different applications and services on demand according to the specific needs of each mission.

3.5. Research impact/dissemination

This chapter results from an 5-month internship realized at the INTA (Instituto Nacional de Técnica Aeroespacial) (more details can be found in Appendix A).

This chapter is based on our article "MILANO: una vision futura para un UAS tactico" published in *VI Congreso Nacional de i+d en Defensa y Seguridad. DESEi+d'18, Noviembre 2018.* which was selected among the ten best-rated articles for the *ISDEFE I+D+i ANTONIO TORRES* award. This line of work, as well as the collaboration with INTA, have remained active during the last years, giving rise to a publication in the *9th International Workshop on Computer and Networking Experimental Research using Testbeds (CNERT)*, located at the IEEE *International Conference on Computer Communications (INFOCOM), May 2022.* which was selected as **Best Paper Award**.

PART III

CONTRIBUTIONS TO EXPERIMENTAL ENVIRONMENTS

This part presents different experimental contributions in the research area of Unmanned Aerial Vehicle (UAV). This experimental work covers different needs/gaps that have been identified after a deep analysis of state of art at the start of the thesis.

Chapter 4 presents a Fifth Generation of cellular network technology (5G) deployment for low coverage areas based on Network Function Virtualization (NFV) technologies in which different domains are incorporated, including a domain with UAVs.

Chapter 5 presents a complete analysis of the performance obtained using the 4G/LTE cellular network in low-resource/constrained devices typically used as a UAV payload. Once it is demonstrated that this 4G/LTE meets the minimum requirements (Key Performance indicators presented by the 3GPPP), indoor flights confirm the approach.

4

Deploying an NFV-Based experimentation scenario for 5G Solutions in underserved areas

The Fifth Generation of cellular network technology (5G) is becoming a reality. Researchers and industry are working conscientiously to build ground-breaking solutions that *(i)* increase transmission speeds, *(ii)* reduce latency, *(iii)* decrease energy consumption, or *(iv)* improve connectivity. The 3rd Generation Partnership Project (3GPP) [72] is working on a series of releases to compose the 5G network scenarios that are being developed to fulfil the requirements imposed by Enhanced Mobile Broadband (eMBB) [73], Ultra-Reliable Low Latency Communications (URLLC) [74], and Machine Type Communications (mMTC) [75]. Release 15 [76] of 3GPP specifications focused on high data rates for eMBB, with a peak downlink rate of 20 Gb/s and 10 Gb/s uplink rate. Release 16 [77] is focusing on reducing the end-to-end latency and increasing robustness for URLLC, and Release 17 [78] is under specification to reduce power consumption on power-limited devices and increase the number of connections for mMTC.

Besides these promising improvements, presently the International Telecommunication Union estimates that 46.4% of the world population cannot properly access the Internet, and connectivity in remote/low populated areas is normally considered to require considerable effort and investments [79]. However, one of the key performance indicators[1] to be achieved at an operational level of the 5G ecosystem, is to provide ubiquitous access including rural and remote areas. In continental-size countries, such as Brazil, one major application scenario is the remote and rural area coverage, where a large parcel of the population is living without connectivity and where informatization of the fields is required to increase farms productivity. Since costs are the major hurdle for mobile network operators, 5G must also present an operation mode that allows for extensive coverage, reducing the number of Base Station (BS) and spectrum costs. In addition, this solution will have substantial social benefits, such as guaranteeing universal access to education and healthcare, or the digital integration of different isolated communities and countries.

---

[1] 5GPPP, KPIs, 2020, accessed Tuesday 15th November, 2022. Available: https://5g-ppp.eu/kpis/

Over the past few years, there has been considerable proliferation of experimentation projects in 5G technologies. However, there is no a widely commercial deployment of the 5G technology (most of the deployments are still based on legacy infrastructure and standalone 5G solutions have not been deployed yet in many places). Accordingly, most of the experimentation is done under consortium projects involving relevant entities, including end users, technology developers, and operators. In Europe, many projects have been funded by the European Union to provide 5G experimentation facilities and foster the adoption of 5G technologies by vertical sectors. These projects include 5G-VINNI [80], the 5GINFIRE [81], the 5Growth [82], the 5G-DIVE [83], or the 5G-EVE [84].

This chapter presents an experience report after building a use case scenario with 5G technologies (Figure 4.1) within the Horizon 2020 Remote area Access Network for 5th GEneration (5G-RANGE) project [85], a 3-year European and Brazilian cooperation effort, whose main objective is the design and implementation of a remote area access network solution under 3GPP specifications and the 5G standard technologies. The contribution presented in this chapter has been deployed at the Instituto Nacional de Telecomunicações (Inatel), in Santa Rita do Sapucaí, Brazil, and the 5G Telefonica Open Network Innovation Centre laboratory (5TONIC) in Madrid, Spain. 5TONIC is one of the leading laboratories for 5G experimentation based in Madrid, Spain. The 5TONIC laboratory is established to provide an open ecosystem where members from business, industry, and academia collaborate with the telecoms research projects.

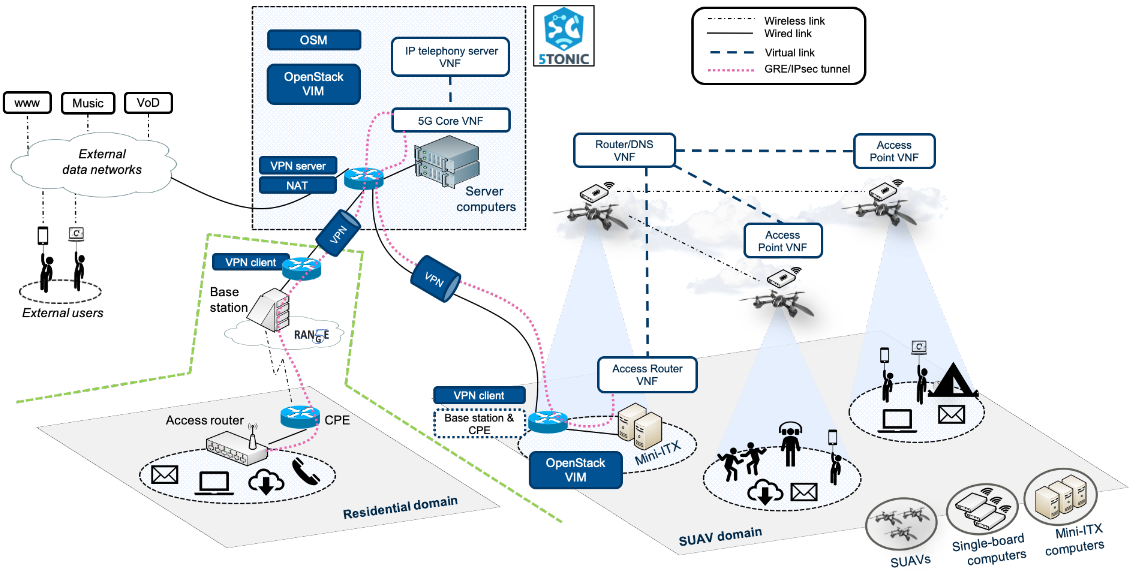


Figure 4.1: Overview of the testbed components and the experimentation scenario

This work is based on two of the main achievements of 5G-RANGE such as a multi-site Network Function Virtualization (NFV) testbed designed for experimentation with Small Unmanned Aerial Vehicle (SUAV) [19], and the new physical (radio) and Medium

Access Control (MAC) layers developed to overcome the long-range link under 5G key performance indicators, targeting a cell radius of 50 km with 100 Mb/s at the edge and closing the connectivity gap in remote and rural areas [86].

However, this chapter shows the whole experimentation scenario that has been built integrating both solutions and presents them in two different domains that can interact with each other. One of the domains includes the physical and MAC layers that allow for Dynamic Spectrum Allocation [87] by exploiting vacant TV channels using Cognitive Radio techniques [88]. Spectrum sensing, low out-of-band emissions, fragmented spectrum access [89] and cognitive cycle are some of the features that must be incorporated in the physical and MAC layers to allow 5G networks to be used to close the connectivity gap in remote and rural areas. The second domain includes the SUAV scenario where network functions can be automatically deployed to support casual network communications beyond the boundaries of the 5G-RANGE radio cells, or to cover shadow areas within the radio cells themselves (e.g., fire extinction, search and rescue operations, festive events, field inspections).

The chapter is also showing the methodology to perform this type of complex integration with different domains located in remote locations. The domains are connected using a Virtual Private Network (VPN)-based overlay network (so that they can transparently be deployed anywhere) and through a baseline 5G core software.

In addition, it is shown how all the network layer components in both domains (as suggested by the 3GPP) are virtualized and executed as Virtual Network Function (VNF), using a Management and Orchestration (MANO) platform defined by the ETSI in the context of NFV paradigm. Wireless Access Point (AP) that offer network access connectivity to end users within their vicinity may be deployed as VNFs in several SUAVs, some other SUAVs or Ground Units could deploy virtualized network routers, supporting the exchange of data between users in the geographic area, or a gateway function can be available at specific locations, enabling for data communications between users and external data networks through a 5G Core (which would also be provided in the form of one or several VNFs in the operator domain). The chapter shows how this approach enables an easier and more flexible incorporation of additional domains to the scenario, aiding the development of proof-of-concept activities involving Internet, third-party, and operator-specific services, such as web browsing, email, video on demand, and Internet Protocol (IP) telephony. Finally, the chapter provides different measurements performed over the infrastructure.

The rest of the chapter is organized as follows: section 4.1 covers the background on the 5G-RANGE technologies. Section 4.2 presents the methodology to build the experimentation scenario. Section 4.3 presents some results and the final integration of the testbed. Finally, section 4.4 concludes the chapter.

4.1. Background on 5G-RANGE technologies

To support effective network communications over remote areas, 5G-RANGE has followed a practical approach: *(i)* the development of novel physical and MAC layer mechanisms, able to efficiently handle data communications over long distances; *(ii)* the adoption of well-known and widely used technologies, as well as recognized standards under development in the context of 5G networking, to provide end-user terminals with end-to-end network connectivity, and support the provision of an operator, third-party, and Internet services. In addition, the design of mechanisms to enable cost-effective network services over delimited geographic areas (e.g., to support network communications beyond the boundaries of the 5G-RANGE radio cells) is also considered.

This section identifies and provides a brief overview of these technologies, outlining the technological background that has driven the design and the development of the experimental testbed described in the following section.

4.1.1. Physical and MAC layers

The 5G-RANGE project has developed an innovative new radio physical layer (a contribution made mainly by INATEL) using a powerful channel code scheme to enable a robust long-range link. A polar code [90] with a variable coding rate is used to protect the data received from the upper layers, increasing the system performance in terms of bit and block error rates. The encoded data are mapped into Quadrature Amplitude Modulation (QAM) symbols, where modulation order can be selected according to the channel conditions (from 4-QAM up to 256-QAM).

After the QAM mapping, the data symbols are applied to the Multiple-Input Multiple-Output (MIMO) block, which can operate in two different modes. In the first mode, a space-time block coding [91] is used to enhance the system robustness for users located far from the BS. This mode allows the receiver to combine the data transmitted by the two antennas, resulting in a maximum diversity gain that is twice the number of receive antennas. The second operation relies on space multiplexing [92], where the two transmit antennas are used to send different data blocks, doubling the overall data rate of the system. Typically, space-time block coding is used to enhance the system robustness for users located far from the BS, while space multiplexing is used to increase the data rate for those users close to it.

This physical layer is controlled and configured by the MAC layer, according to the channel conditions. The main novelty introduced at this layer is the cognitive cycle, exploiting vacant TV channels using cognitive radio techniques [88]. The User Equipment (UE) can be instructed by the BS to perform a spectrum measurement at its location infers whether the channel is available or occupied. The Collaborative Spectrum Sensing Optimized for Remote Areas [93] block acquires samples collected from the channel and

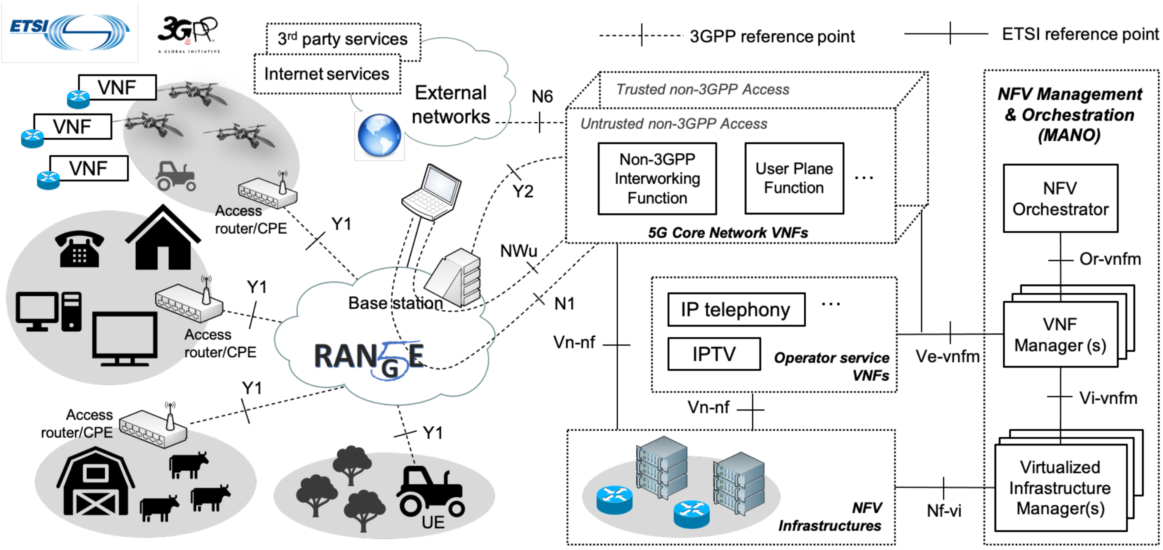


Figure 4.2: High-level overview of the 5G-RANGE architecture.

each requested UE performs a Primary User detection algorithm. The outcome of this measurement is one of two hypotheses: H0, stating that the channel is available or H1, stating that the channel is occupied. The measurements are sent to the BS, where a dynamic spectrum allocation [87] function is responsible to fuse all the measurements into a single decision variable. Data from a geolocation database can also be used to aid the decision process, i.e., allowing the selection of the channels that shall be investigated as potential idle channels. Once a set of channels is identified as vacant, a resource scheduler allocates the user data to protocol data units, which will be delivered to the physical layer. These protocol data units also carry the configuration of the physical layer to be employed by each user (e.g., code rate, QAM mapper and MIMO scheme). Data recovered by the physical layer are also delivered to the MAC layer using the same protocol data unit structure.

4.1.2. Network layer

The network layer has a fundamental role to support end-user communications. It complements the physical and MAC layers of the radio access network with the required features to provide the UE with secure end-to-end network connectivity towards other UE and external networks. Figure 4.2 outlines the role of the network layer within the 5G-RANGE architecture.

The connectivity service offered by the network layer is realized with the use of a 5G core network, as defined by the 3GPP [72]. The 5G core network identifies a set of well-defined protocols and interfaces allowing the interconnection of non-3GPP access networks (e.g., the 5G-RANGE access network), which may support trusted or untrusted connectivity to a 5G core network or, more generally, to a public mobile network infrastructure. Moreover, it enables the establishment of protected communications towards the UE, using standard tunneling protocols over the access network (for non-3GPP accesses, these are Generic Routing Encapsulation (GRE) [94] and Internet Protocol security (IPsec) [95]).

From an architectural perspective, the 5G-RANGE network layer relies on ETSI NFV. This way, the different elements of the 5G core network, as well as the constituent functions of operator and third-party services, can be executed as VNFs. The lifecycle management of all the VNFs framed at the network layer follows the procedures indicated by the ETSI MANO framework. Figure 4.2 outlines the 5G-RANGE architecture.

In 5G-RANGE, ETSI NFV is also considered to support the cost-effective deployment of lightweight network functions on localized remote geographic areas. In this respect, the approach taken by the project leverages resource-constrained platforms that can be embedded in vehicles that may exist or be deployed in the remote area (e.g., SUAVs, harvesters, tractors, etc.). These vehicles are consequently transformed into functional mobile compute nodes, offering computing, storage and network resources that can be under the control of a MANO platform to support the execution of VNFs. They have the potential to be placed around specific locations and be interconnected, allowing the on-demand creation of functional NFV infrastructures over localized geographic areas. This NFV infrastructure can be used to complement the access network resources, and provide cost-effective telecommunication services controlled by a MANO platform. This way, the proposed approach allows the on-demand creation of a functional NFV infrastructure over a localized geographic area. This infrastructure can be used to complement the access network resources and provision cost-effective telecommunication services under the control of a MANO platform. In addition, the NFV infrastructure facilitates the dissemination of data across the deployment area, through the multi-hop ad hoc network built by the mobile compute nodes.

In 5G-RANGE, this approach has been explored in different use cases, including service provisioning beyond the boundary of radio cells [7], supporting emergency communication services in remote areas (Chapter 6), and disseminating vertical-specific data, particularly in smart farming scenarios [19].

4.1.3. NFV lestbed description and VNF repository

As previously commented, the goal of this work is to build the scenario shown in Figure 4.1. For this purpose, we have used an NFV testbed that is available at the 5TONIC laboratory based in Madrid, Spain. A detailed description of the NFV testbed can be found in [19]. For the sake of completeness, in the following we present a summary of its main features and components. The testbed includes a functional MANO platform based on Open-Source MANO (OSM)[2], an ETSI-hosted project providing an NFV orchestration software stack. A well-known and widely adopted cloud computing solution, OpenStack, was selected to provide the Virtual Infrastructure Manager (VIM) functionalities. Both the OSM stack and the OpenStack controller run in independent virtual machines, easing the management of the MANO components and their vertical scaling to satisfy operational needs. The MANO platform supports the automated deployment of VNFs over an NFV infrastructure composed by three server computers, accounting for a total of 24 vitrual Central Processing Unit (CPU), 96 GB of Random Access Memory (RAM), and 6 TB of storage. These VNFs may also access Internet services, through a Network Address Translators (NAT) function provided at 5TONIC edge router.

To increase the potential for experimentation with 5G-RANGE technologies, the baseline testbed supports the flexible incorporation of network domains, by means of an overlay network architecture based on a VPN service hosted at 5TONIC. Each of these domains may include hardware and software components prototyped by 5G-RANGE partners. It may also host an NFV infrastructure that can be attached to the 5TONIC MANO platform. This way, the addition of new domains allows configuring moderately complex experimentation scenarios, enabling the validation of 5G-RANGE developments along with other assets produced or adopted by the project. Each network domain can be independently managed and evolved, for instance, with the introduction of new functionalities implemented in the context of the project.

Following the aforementioned approach, the testbed integrates a SUAV domain with a portable NFV infrastructure, enabling the creation of experimentation scenarios with mobile compute nodes. The portable NFV infrastructure encompasses six single-board computers, Raspberry Pi (RPi) 3 Model B+, each with two Wi-Fi interfaces. Given their size and weight, these adequately represent the type of resource-constrained platforms that could be onboarded onto SUAVs. The infrastructure also includes five mini-ITX computers, which may serve as ground equipment to deploy more resource-demanding SUAVs. To enable flight experiments, the portable NFV infrastructure is completed with four Parrot Bebop 2[3] SUAVs, each transporting a RPi. An additional mini-ITX computer hosts a VPN client and an OpenStack VIM, each running in a virtual machine. The former behaves as a network router with a virtual link to 5TONIC. The OpenStack VIM exposes the resources of the portable NFV infrastructure to the 5TONIC OSM stack.

On the other hand, the testbed includes the prototypes of a 3GPP UE and a 5G core network. The UE can also behave as an access router, supporting the connectivity of additional end-user devices. Both prototypes implement the data-plane protocol stack defined by 3GPP for an untrusted non-3GPP access [72]. The access router can be

---

[2] Open Source MANO, accessed Tuesday 15th November, 2022. Available: https://osm.etsi.org/

[3] Parrot Bebop 2, accessed Tuesday 15th November, 2022. Available: https://www.parrot.com/es/drones/parrot-bebop-2

provisioned as a VNF at each network domain, in case an NFV infrastructure is available at the domain. Alternatively, it can be deployed as a hardware device, because an implementation of the access router has been made available on a single-board computer, RPi model 3B+. Every access router function is connected to the 5G core network function through a GRE over IPsec tunnel, as dictated by 3GPP specifications for non-3GPP accesses. The 5G core network component has been provisioned as a VNF. Hence, different experiments can use independent instances of this network function, which can be deployed by the MANO platform on the 5TONIC NFV infrastructure. In addition, the 5G core network prototype implements a GRE/IPsec tunnel endpoint and provides connectivity to external networks. From that perspective, it can be seen as a baseline implementation of the data-plane protocol stack of a Non-3GPP InterWorking Function and a User Plane Function, as defined by 3GPP. The access router and the core network functions have been implemented using the Linux *ip-gre* module, and the *ipsec-tools* and *racoon* Linux packages. Additionally, a VNF has been implemented providing the functionalities of an IP telephony server based on the open-source Session Initiation Protocol (SIP) server *Kamailio*. It can be deployed over the 5TONIC NFV infrastructure, supporting the registration of end-user SIP phones and the establishment of voice and video calls in experimentation scenarios.

To enable experimentation activities with resource-constrained compute nodes (e.g., single-board computers onboarded on SUAVs), the testbed offers two additional VNFs: an AP VNF, providing the functions of a Wi-Fi access point and a DHCP server, and a Router/DNS VNF. Both VNFs have been prototyped as lightweight software functions using Linux and virtualization containers, so that they can be executed on the single-board computers of the SUAVs domain. All these components are available under an open-source license in the SUAV network layer repository[4].

4.2. Experimentation scenario: Methodology from design to validation

Figure 4.1 shows the scenario that has been created with the goal of validating 5G-RANGE technologies in the context of a specific use case. The subsequent subsections detail the followed methodology from the design to validation. To facilitate the presentation of Sections 4.2 and 4.3, and the understanding of the applied methodology, Figure 4.3 provides the overall flowchart illustrating the definition, deployment, integration and validation of the experimentation scenario.

---

[4] Open Source NFV Packages and Descriptors, accessed Tuesday 15th November, 2022. Available: https://vm-images.netcom.it.uc3m.es/5GRANGE/

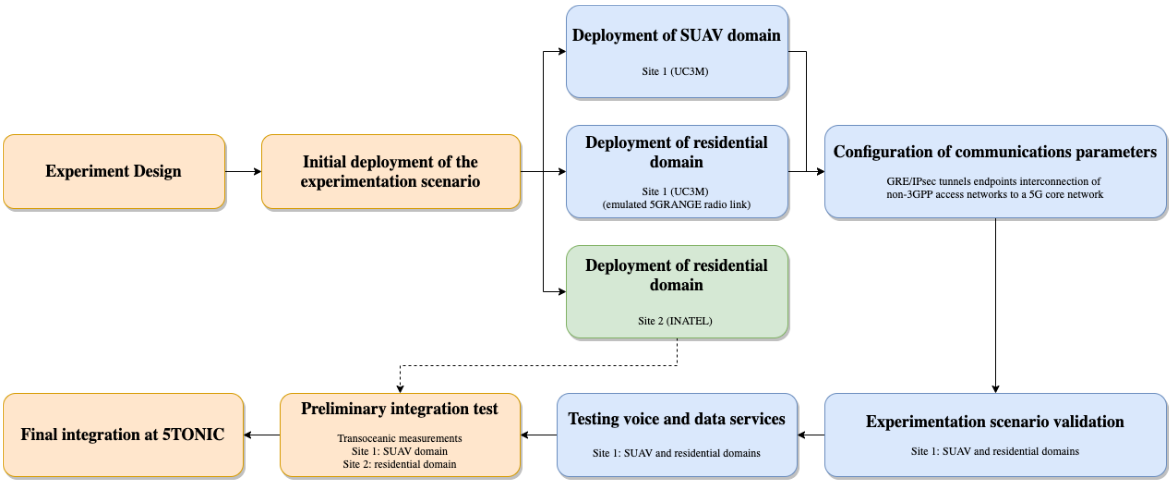


Figure 4.3: Methodology to define, deploy, integrate and validate the experimentation scenario.

4.2.1. Description of the experimentation scenario

In the experimentation scenario, an access router is physically available at a residential environment in a remote area, serving as a wireless access point. The access router enables data exchange between residential users and external networks, through the GRE/IPsec tunnel established with the 5G core network. Hence, users within the remote area may access Internet and operator-specific services, such as web browsing, email, audio/video live streaming, or IP telephony. The radio access network is supported by a BS and a Customer Premises Equipment (CPE), which implement the physical and MAC layer protocols of 5G-RANGE. Figure 4.4 shows the protocol stack involved at the different network functions of the residential domain, the access network, and the 5G core network to support data exchange.

The experimentation scenario also reproduces a situation where similar Internet and operator-specific services are to be provided to users beyond the limits of a 5G-RANGE radio cell (e.g., users in a festive event, or emergency response teams in a fire extinction or search and rescue operation). For this purpose, several SUAVs are deployed over the area, hosting a set of network functions that enable the provision of those services. In this case, the SUAVs are hovering in a static position, providing their intended services on a defined geographical area. These network functions include two wireless access points embedded as VNFs on two SUAVs, which serve as wireless hotspots to end users within their vicinity. A network router and a DNS server are jointly deployed as a VNF on a third SUAV. This VNF presents a virtual link towards an access router, which is virtualized on a ground equipment within the radio coverage of a 5G-RANGE radio cell (in a realistic scenario, this wireless link could rely on a multi-hop network path conformed by several wireless routers and SUAVs). The access router behaves as a GRE/IPsec tunnel endpoint

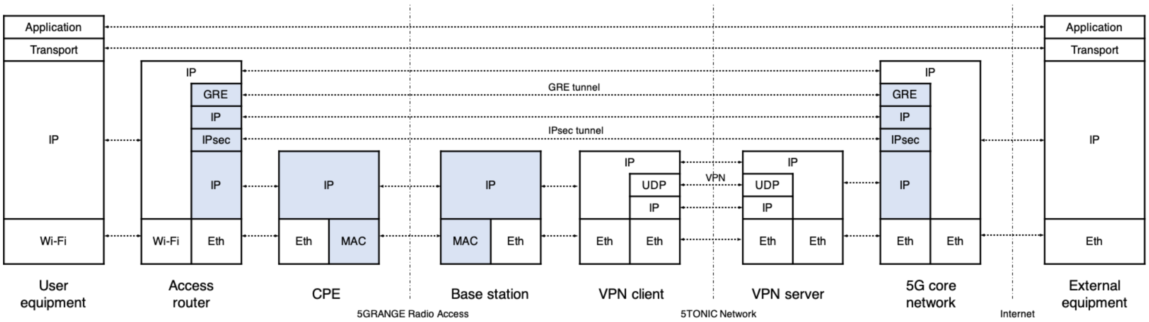


Figure 4.4: Data-plane protocol stack of the residential environment.

towards the 5G core network, supporting data communications between end users and external networks. Communications among users in the residential and SUAV domains are supported through their radio access networks and the 5G core network.

4.2.2. Initial deployment of the experimentation scenario

The experimentation scenario has been created as a composition of two network domains: one hosting the components of the residential environment (the residential domain); and a second one with the SUAV infrastructure (the SUAV domain). As a first step to build the experimentation scenario, we used our testbed resources to build the residential domain at 5TONIC. This domain includes two single-board computers (RPi 3 Model B+), representing a 5G-RANGE BS and a CPE. This initial deployment of the experimentation scenario obviates the physical and MAC layer components of the 5G-RANGE access network (these will be integrated in the experimentation scenario at a later stage). Instead, the BS and the CPE are interconnected through a 100 Mb/s switch, providing the equivalent maximum throughput of the 5G-RANGE access network. The BS is connected to a mini-ITX computer, which deploys a VPN client. This implements a virtual link towards 5TONIC, making each device at the domain accessible from the laboratory. The CPE is connected to an access router function, supported by another single-board computer (RPi 3 model B+). The access router provides the functions of a wireless access point, offering network access connectivity to an end-user device (a laptop), and implements a GRE/IPsec tunnel endpoint towards the 5G core network.

The 5G core network component is part of a network service that has automatically been deployed through the MANO platform of the 5TONIC NFV testbed. The network service includes a set of VNFs that are instantiated on the SUAV domain, offering the functionalities of wireless access points, routers, and other supporting functions on SUAV units. An access router VNF is deployed on a ground compute node (a mini-ITX computer), supporting the exchange of information with external networks. For this purpose, the access router behaves as a GRE/IPsec tunnel endpoint towards the 5G core network VNF. The latter is deployed at 5TONIC premises along with an IP telephony server VNF, which supports the establishment of calls among end users in our experimentation scenario. The deployment of the whole network service was accomplished at 5TONIC premises, where a specific location for indoor flights is available. Finally, an additional mini-ITX computer deploys the VPN client that handles the communication of the SUAV domain with other testbed components.

4.2.3. Configuration of GRE/IPsec tunnel endpoints

The use of GRE, IPsec, and the VPN service, may cause excessive fragmentation on data packets. When packets arrive to the access router or the 5G core network, they are fragmented by GRE before being processed by IPsec. This is because the default Maximum Transmission Unit (MTU) of the GRE interface is 1476 bytes, a lower value than the typical size of regular data packets (1500 bytes). The GRE tunnel interface splits each data packet into two fragments, encapsulating them into new IP packets with a GRE and an outer IP header. With a size of 1500 bytes, the first of these packets is also required to be fragmented after being processed by IPsec (after appending the IPsec protocol overhead, the packet exceeds the MTU of the outgoing link). A similar situation occurs at the VPN endpoint, which processes three data packets and performs an additional fragmentation of the first packet (after appending the VPN protocol overhead, the first packet exceeds the MTU of the outgoing link). These fragmentation processes lead to increased overhead in terms of protocol headers and encryption of smaller packets, which necessarily impacts the achievable throughput.

As suggested in [96], this situation can be mitigated with an appropriate configuration of the MTU at the GRE tunnel interfaces. With a suitable value lower than 1500 bytes, data packets would only be fragmented at the GRE tunnel once, producing fragments with a size such that subsequent IPsec and VPN protocol overheads could be accommodated without additional fragmentation processes. Considering the reference MTU values indicated in [96], along with the protocol overhead of the VPN service, we have set the MTU on the GRE tunnel interface to 1360 bytes in our experimentation scenario.

For this purpose, we performed a set of experiments to verify their functional behavior and their performance in the provision of network communications. These experiments have also granted a better understanding on the impact of protocol overheads introduced by the GRE and IPsec processes needed to establish the proper connectivity towards the 5G core network, as well as of the VPN service that interconnects each network domain to the 5TONIC infrastructure. To evaluate the effect of this setting, we have done a performance evaluation, deploying all the elements of Figure 4.4 as virtual machines at the 5TONIC NFV infrastructure (except the CPE and the BS components). The deployment served to reproduce the communication scheme shown in the figure, including a virtual UE and a virtual external equipment, and excluding the components that are specific to the physical/MAC layers of 5G-RANGE (i.e., the physical/MAC and BS functions of

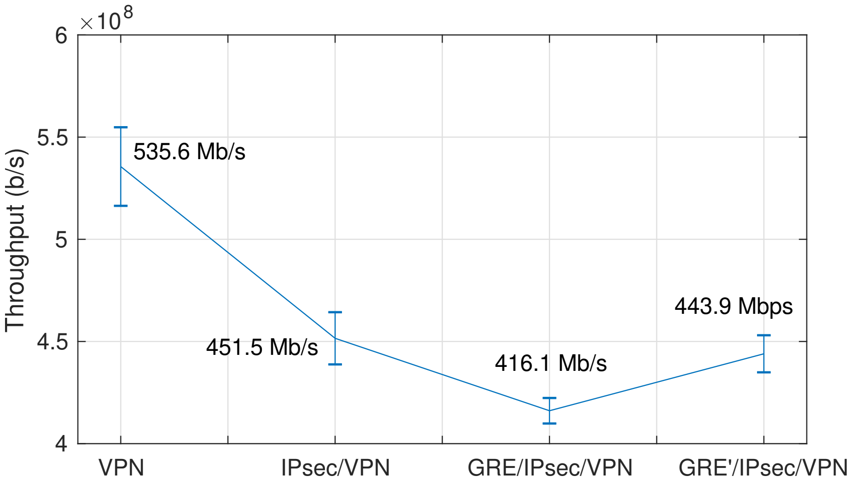


Figure 4.5: Performance evaluation of GRE/IPsec tunnel endpoints.

Figure 4.4).

Figure 4.5 presents a synoptic overview of the results of our evaluation. Our tests with the *iPerf* tool show a maximum average throughput of 443.97 Mb/s between the user and the external equipment (*GRE'/IPsec/VPN* case in the figure), with an observable increase of approximately 6.7% with respect to the case where the MTU on the GRE tunnel interface is set to its default value of 1500 bytes (*GRE/IPsec/VPN*). These performance figures suggest the capacity of the access router and the 5G core network prototypes to accommodate the requirements for all the experimentation scenarios considered in 5G-RANGE, given that the maximum throughput required in a 5G-RANGE access network segment is 100 Mb/s. The optimized MTU value of 1360 bytes was used at the GRE tunnel interfaces in all our subsequent experiments. The errors bars in Figure 4.5 correspond to the standard deviation.

To gain a better understanding of the performance overhead introduced by GRE/IPsec tunneling processes, Figure 4.5 also presents the maximum average throughput for the cases where: GRE is disabled and the tunnel endpoints are only supported by IPsec (*IPsec/VPN* in Figure 4.5); and GRE/IPsec are disabled, and the access router and the 5G core network behave as network routers (*VPN* case).

4.2.4. Validation of the experimentation scenario

After the deployment of the experimentation scenario, we evaluated the throughput that could be achieved between a network domain and the 5G core network component. Our measurements with the *iPerf* tool reveal an available throughput of 248 Mb/s between a VPN client and the 5G core network VNF (these measurements are similar in both the SUAV and residential domains). This value is obviously lower than the already commented 443.97 Mb/s (see Figure 4.5) and this is because this new value corresponds to a measurement in a real network. In any case, it is still considerably above the data rate considered in the design of the 5G-RANGE radio access, i.e., 100 Mb/s.

As expected, the available throughput on the SUAVs is lower than the available throughput in the VPN client as they communicate over a multi-hop ad hoc Wi-Fi network. The tests from the AP VNF SUAV result in average throughput of 22.7 Mb/s, while the tests from the Router/DNS VNF SUAV have come out with 53.6 Mb/s providing suitable values for subsequent experimentation activities.

On the other hand, tests with the Linux *Ping* command between the VPN client of the SUAV domain and the 5G core network VNF result in an average Round Trip Time (RTT) of 1.59 ms, providing a low value that is suitable for subsequent experimentation activities. We want to highlight that in scenarios spanning remote domains, the network path between a domain and the 5G core network would be established across the Internet, being subject to network congestion and potential bandwidth limitations and high end-to-end network delays. Although this is not necessarily a limiting factor to build distributed experimentation scenarios, it should be considered when designing the tests that will be performed on top of them.

4.3. Multimedia tests and final integration

4.3.1. Testing voice and data services

Once the viability and suitable operation of both the access network and the extension of the network formed by the SUAVs have been confirmed, different multimedia services have been trialed. These tests have been also replicated with the real radio equipment, once it was integrated in the laboratory for a project review demonstration. The experimentation scenario has been used to test a representative set of voice and data services that might be demanded by end users in remote areas. This has served to verify the appropriateness of our testbed to develop proof-of-concept activities in the context of a specific use case.

First, we established a voice call between a user in the residential domain and a user in the SUAV domain using a softphone based on the SIP, *Bria*[5], which was installed on one laptop in each domain. The softphones were configured to use the IP telephony service VNF of Figure 4.1. Figure 4.6 represents the transmission rate (*SIP TX*) and the received throughput (*SIP RX*) of the voice traffic, measured at the laptop in the residential domain. Around second 75, the video was turned on in both softphones, resulting in a traffic increment (from a few Kb/s up to approximately 2 Mb/s). The average *jitter* in both directions was 0.5 ms, resulting in appropriate interactive real-time

---

[5] Bria softphone, accessed Tuesday 15th November, 2022. Available: https://www.counterpath.com/bria-solo/

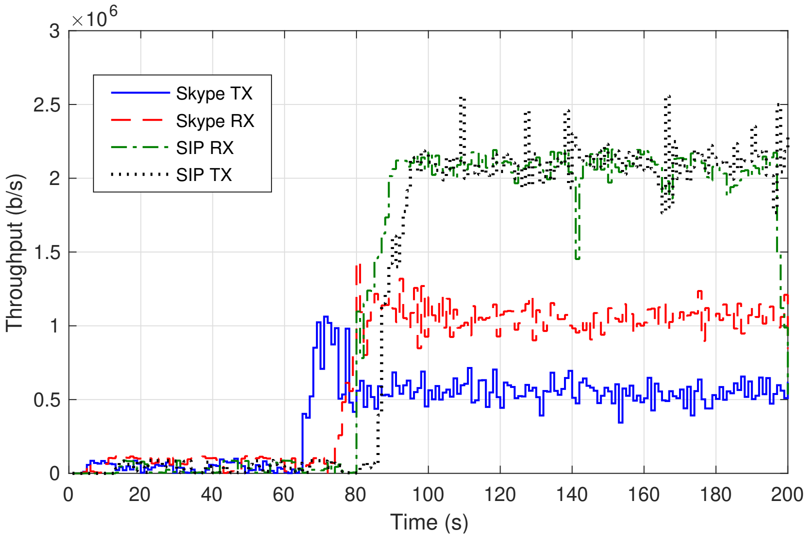


Figure 4.6: Data rates of SIP and Skype calls.

communications. The user experience during the call has been satisfactory, with no audio glitches nor skipped video frames.

In a second experiment, a group call was set up using *Skype*. The call involved the same users as in the previous experiment, with an additional third user connected to *Skype* through an external network (a commercial fixed access). Figure 4.6 shows the transmission rate and the received throughput of the voice traffic observed by the residential domain user. The video was activated approximately within 70 s, causing a consequent increase of the traffic. The received traffic nearly doubles the transmitted traffic because, in a *Skype* video conference, the video stream of each participant is routed to an external server cluster, which in turns forwards it to every other participant. That way, the received throughput corresponds to the video transmitted by the other two participants in the call.

Finally, the laptop at the residential domain was used to access a 4 K video from *YouTube*. Figure 4.7 shows the throughput of the received video at the laptop (labelled as *video throughput*). The video was continuously played out with no freezing nor skipped video frames.

4.3.2. Final integration and validation

In the previous section, we verified the appropriateness of our testbed to develop Proof-of-Concept (PoC) activities in the context of a specific use case. In this section, we present the methodology followed to integrate the prototypes of a 5G-RANGE BS and a CPE into the experimentation scenario. These prototypes are available at Inatel in Brazil, and were temporarily brought to 5TONIC for a demonstration.

The transceiver prototype [85] has been developed using the software-defined radio

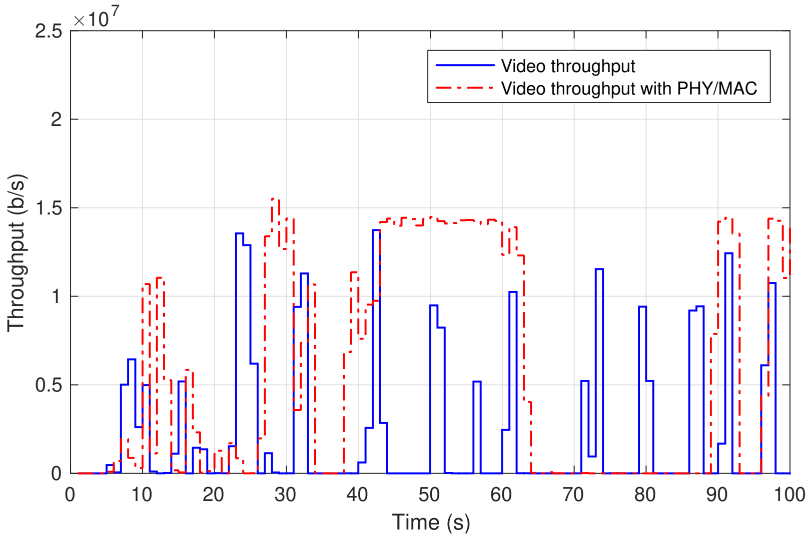


Figure 4.7: Data rates of video-on-demand service.

strategy, where the entire base band processing is implemented using C language over GNU Radio platform. This approach leads to the maximum flexibility, since the radio behavior can easily be adapted to the different channel conditions. In this prototype, data from the network layer is delivered to the MAC layer through an Ethernet connection. The MAC layer shares the physical resources for the different users according to their individual throughput demands and channel responses. Adaptive modulation and coding is used to guarantee a desired quality-of-service in terms of bit error rate. Modulation order and coding rate for each user is automatically defined based on the channel quality report provided by the mobile terminals. Once the data is mapped to the physical resources, the information is processed by the channel encoder and modulated using generalized frequency division multiplexing, an innovative waveform capable of providing robustness against doubly dispersive channels. In this prototype, a 2×2 MIMO system has been implemented, providing spatial multiplexing gain for users that are nearby the BS and diversity gain for those located far away from it. Once the data is processed by the base band unit, it is delivered to the digital-analog converter, coupled with the radio frequency head. In our implementation, universal software radio peripheral are used to receive the digital samples from the computer running the transceiver's MAC and physical cores, converting them to the radio frequency signal to be delivered to the transmit antennas. A power amplifier provides 6 watts per antenna, allowing the signal to be received up to 50 km from the BS, with data rates up to 100 Mb/s over a 24 MHz channel. Spectrum sensing is also implemented using the software-defined radio approach. It is performed by the mobile terminals and the collected measurements are periodically sent to the BS using the control channel. The MAC layer uses this information to decide which channels are available to be allocated to the users.

To facilitate integration activities, we leveraged the capacity of our NFV testbed to incorporate remote network domains. In particular, the residential domain of Figure 4.1 was initially replicated at Inatel, replacing the single-board, back-to-back connected computers by the BS and the customer premises equipment prototypes. The VPN client at Inatel was configured with the same security credentials as the VPN client of the 5TONIC residential domain. All the equipment and networks at the Inatel domain were configured with the same IP addresses as their correspondent equipment and networks at the 5TONIC residential domain. An access router function was also deployed. To ease the deployment of this function, the memory card of the device providing the access router at 5TONIC was cloned, being installed on a RPi 3 model B+ at Inatel.

We want to highlight that this methodology allowed addressing most of the integration aspects in advance, before the physical/MAC layer prototypes were brought to 5TONIC. Following this method, we have performed a straightforward final integration. Otherwise, following a traditional methodology, several configurations must be made after the integration into a single local demonstrator, such as *(i)* configure the addressing space, *(ii)* configure the GRE/IPsec tunnel, *(iii)* the integration of physical, MAC, and network layer components, or *(iv)* the overall validation with voices and data services. This work would be even more time-consuming in this case since the institutions are located on different continents. Thanks to the proposed methodology, all these assignments have been configured prior to the final integration. In addition, it enabled the realization of preliminary configuration and tests, verifying the proper interaction among the components at the Inatel residential domain and those available at 5TONIC. Taking advantage of the VPN service, these tests were conducted as if the Inatel residential environment was locally available at 5TONIC. Of course, the aforementioned tests were limited by the performance of the transoceanic network path between 5TONIC and Inatel, which supports an average throughput of 15.07 Mb/s in the 5TONIC to Brazil direction, 6.62 Mb/s in the reverse direction, and an average RTT of 232.46 ms (these values were obtained during a period of 20 days, taking measurements every hour). This information can be appreciated in Figure 4.8. The data is plotted following the standard boxplot shape, which represents the obtained measurements of the available throughput grouped in quartiles.

When the physical/MAC layer equipment was brought to 5TONIC, their integration into the experimental scenario only required the decommission of the two single-board, back-to-back connected computers of the 5TONIC residential environment, which were simply replaced by the physical/MAC layer prototypes. The whole process could be realized in a reduced time frame (less than half working day), making the whole experimentation scenario rapidly ready for the practical demonstration. As an example, Figure 4.7 also shows the throughput of a 4K video received from *YouTube* at the 5TONIC residential domain. The video was delivered through the BS and the customer premises

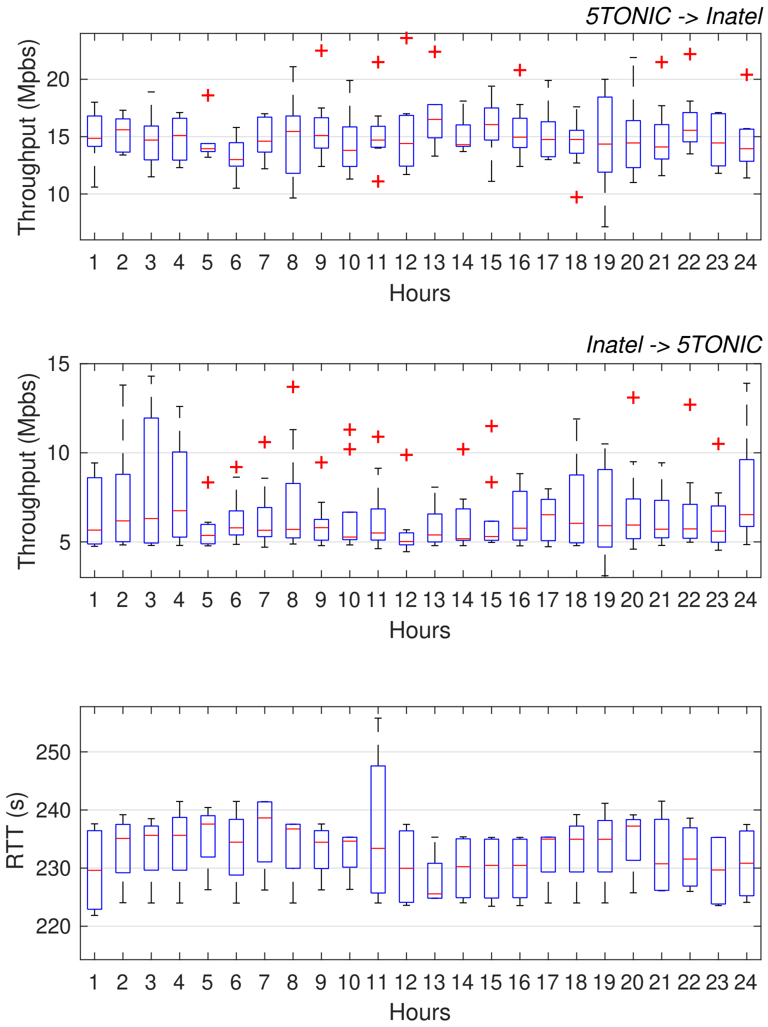


Figure 4.8: Transoceanic network path performance between 5TONIC and Inatel.

equipment prototypes, being uninterruptedly played out and with an appropriate user experience. The traffic pattern is similar to the one shown in the previous section, although not identical, since both experiments were conducted at distant moments in time, using different videos.

4.4. Discussion

This chapter presents an experience report after building a use case practical scenario to trial 5G network technologies for remote/underserved areas. The scenario was created using an existing NFV testbed that supports the flexible incorporation of network domains, this way easing integrating activities. Our experience suggests that the use of distributed network domains has the potential to reduce implementation and test cycles, providing realistic and moderately complex scenarios to stakeholders, who may test new developments along with other assets available at the different domains. In addition, it facilitates joint demonstration activities, as any remote domain can easily be redeployed at a central location (i.e., 5TONIC in case of the 5G-RANGE project) using the same security credentials. The design and development of the different parts of the experiment have been done remotely and later integrated together. This is possible thanks to the use of a VPN established from each site towards the 5TONIC laboratory, which holds the 5G core functionality, allowing the integration of the different network domains. This procedure facilitates the preliminary validation analyses to enable a fast and straightforward real integration procedure. Also, in this chapter, we have proved with commodity equipment that both *(i)* to comply with 3GPP/5G standards, and *(ii)* to use the VPN as an integration and deployment tool, does not limit the system performance.

Our future work will explore the potential of the 5G-RANGE testbed to develop experimentation scenarios for other innovative use cases in remote areas. In addition, we will work on the evolution of the testbed, considering new releases of its software base (i.e., OSM and OpenStack), as well as emergent open-source virtualization technologies for resource-constrained mobile nodes (e.g., Kubernetes and fog05).

4.5. Research impact/dissemination

The outcome of the work behind this chapter has resulted in a publication in the MDPI *Sensors* journal. MDPI *Sensors* is included in the Journal Citation Reports (JCR) with a current impact factor of 3.576 (Q1). In addition, this article has been cited by 4 research papers.

This work results from a collaboration with the Instituto Nacional de Telecomunicações (INATEL) from Brazil in the framework of the H2020 5G-RANGE project. This work was part of the project final review (March 2020 at 5TONIC premises). Moreover, some content is part of Deliverable 5.2 ("Final Version of the Network-level Architecture and Procedures") of the this project.

5

Cellular and virtualization technologies for UAVs: An experimental perspective

Unmanned Aerial Vehicle (UAV) operations have been exponentially growing over the last few years, not only for non-professional users (e.g., recreational flights, aerial photography, racing competitions), but also for commercial/professional purposes, providing a wide array of services in many types of environments, including both civil and military fields or even urban populated areas (e.g., packet delivery, medical supplies transportation, infrastructure surveillance, emergency response). Many attributes inherent to these aerial vehicles, such as fast mobility, flexibility, and adaptability to different conditions, enable promising wireless services and application deployments when combined with the appropriate communication facilities. UAVs have the possibility of, literally, becoming mobile terminals when connected to a conventional cellular network, which together with their capacity to carry a large variety of equipment as payloads (e.g., video cameras, sensors), allow them to potentially support several tasks such as real-time video streaming [97] or item delivery[1], creating promising business opportunities for stakeholders and cellular operators. These operations involve a certain degree of risk due to their complexity and nature in civilian environments. Although they have proven to be of great value to society in several circumstances, such as delivering time-critical medical supplies or evaluating disaster areas, there is a significant challenge of managing all the applications simultaneously and in the same airspace.

In this context, relevant institutions such as the Federal Aviation Administration (FAA), Google, Amazon, or NASA are working together on the UAS Traffic Management (UTM) project [98,99]. Like the internationally well-established Air Traffic Management (ATM) ecosystem, the UTM aim is to manage automated, safe, and efficient low altitude airspace operations in Unmanned Aircraft System (UAS) operations. Notwithstanding their similarities, which derive from tackling a common issue between them (controlling airspace), UTM and ATM have significant differences. The lower airspace (managed by

---

[1] Amazon Prime Air, accessed Tuesday 15th November, 2022. Available: https://www.amazon.com/b?node=8037720011

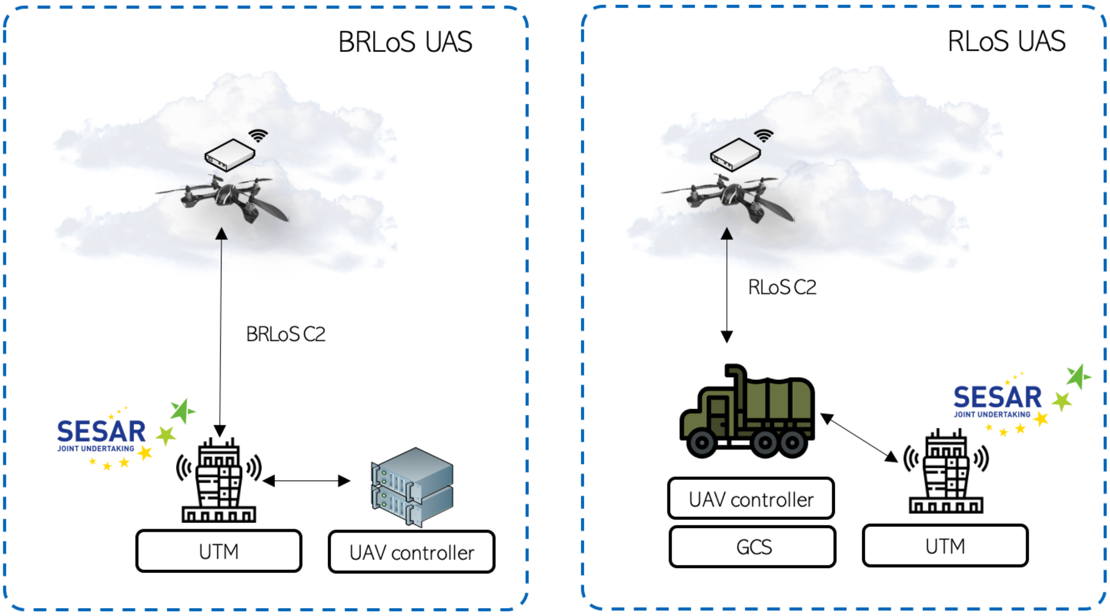


Figure 5.1: Different types of UAS following UTM: Beyond Radio LoS (BRLoS), and Radio LoS (RLoS).

UTM) is continuously subjected to several changes, and the situation on the ground must be continuously monitored. Besides, the trajectories calculation at lower airspace must take into account not only the collisions with other vehicles, but the ones with buildings and potential obstacles as well.

UAVs are commonly controlled using Radio Line-of-Sight (RLoS) from a Ground Control Station (GCS) (i.e., a direct link from the GCS to the UAVs). In particular, Small Unmanned Aerial Vehicle (SUAV) must operate in a relatively close area (depending on the selected RLoS technology) to the GCS to enable the connection. In contrast, the UTM project researches (among other topics) enabling Beyond Radio Line-of-Sight (BRLoS) UAV operations at low altitudes (below 120 m above ground level), facilitating and boosting near-future UAV commercial applications. Figure 5.1 presents the RLoS and BRLoS UAS considered above (UASs are normally understood to be both the UAV and the GCS working together). As it can be appreciated in both systems, it is essential to communicate the UAV controller with the UTM to enable emerging applications, since different UASs may coexist in the same airspace.

Many efforts over these research lines are also being invested in by the European Union by the Single European Sky ATM Research (SESAR). The SESAR initiative is working on the development of U-space [100, 101], a set of new services to uphold safe, effective, and secure access to European airspace for large numbers of UASs. In addition, many projects have been funded by the European Union to develop safe and efficient

UAS traffic management. These projects include 5D-AEROSAFE[2], DRONES4SAFETY[3], RAPID[4] or Labyrinth[5]. This research work has been done within the framework of this Labyrinth project.

In this context, the Labyrinth project aims at creating and validating new UAV applications to enhance the safety, security, and efficiency of civil system transportation and urban areas. Labyrinth will create a centralized system able to communicate with all the UAVs located in a defined area, processing their source and destination waypoints to compute their potential paths and avoid collisions (accomplishing this with the SESAR policy). Moreover, Labyrinth will build fully-operational use cases in four relevant transportation environments: *(i)* road (e.g., speed control, accident management, plate identification); *(ii)* waterborne (e.g., seaport facilities supervision); *(iii)* air (e.g., bird scaring, airport facilities supervision), and *(iv)* emergency (e.g., provisioning of medical supplies, discover escape routes). The Labyrinth consortium includes both research institutions and industrial partners, involving relevant key-adopters such as the Spanish General Directorate of Traffic (DGT), the Municipal Emergency Assistance and Rescue Service—Civil Protection (SAMUR—Protección Civil) from Madrid, and the Port Authority of the Eastern Ligurian Sea from Italy.

This chapter presents as the main result a functional testbed in the context of the Labyrinth project, focusing on its communications aspects by including all the required ingredients to complete a UAV mission following the U-space requirements. For this purpose, UAVs are connected to the cellular network to enable connectivity with the UAV controller (the entity in charge of computing flight trajectories and sending instructions to the UAVs) and other UAVs. Moreover, all the corresponding trajectories and instructions are sent from the UAV controller to the UAVs using an automation platform. This system runs on top of a lightweight virtualization platform to further test applications and functionalities beyond the Command and Control (C2). Besides, all this technology has been validated in an indoor flight set of experiments. This chapter also evaluates the 4G/LTE cellular network performance using commercial deployments with different communication prototypes and different network operators and the Fifth Generation of cellular network technology (5G) Stand Alone (SA) cellular network in a laboratory environment. The obtained results highlight that the cellular network properties are adequate for managing the UAVs. Although the 5G commercial deployment is still imminent and does not reach a large part of the population, it serves as a proof of concept to highlight the potential of 5G networks.

The rest of the chapter is organized as follows: section 5.1 reviews the related work and background regarding UAV technologies and communications, as well as the advances of

---

[2] 5D-AeroSafe, accessed Tuesday 15th November, 2022. Available: https://5d-aerosafe.eu/

[3] Drones4Safety, accessed Tuesday 15th November, 2022. Available: https://drones4safety.eu/

[4] RAPID, accessed Tuesday 15th November, 2022. Available: https://rapid2020.eu/

[5] Labyrinth, accessed Tuesday 15th November, 2022. Available: http://labyrinth2020.eu/

the combination of UAV technology with virtualization. Section 5.2 provides details about the testbed, describing in detail all of the technologies chosen for its design, highlighting its main characteristics, functionalities, and the role each one of them in the system. Section 5.3 presents a set of experiments to validate the feasibility of the testbed, including a performance evaluation achieved by the communications prototype, and an indoor flight mission to demonstrate the testbed functionality. Finally, section 5.4 presents the main conclusions and future work for the chapter.

5.1. Related work and background

5.1.1. Communications

Conventional research related to UAVs has typically been focused on navigation, optimal trajectories, and control challenges. Unfortunately, many articles underestimate the difficulties of maintaining stable and reliable communications between the aircrafts and the GCS, where connectivity might not be stable due to the unreliable nature of wireless communications. However, there has been a sharp increase of interest in this particular topic, as it can be seen in some examples that focus on UAS communications.

On the one hand, UAVs beyond their traditional usage as standalone mission platforms are more and more being proposed as flying wireless base stations [102] [103] in 5G networks and beyond (also known as UAV-assisted cellular communications) to support network connectivity in multi-UAV missions. UAVs include different application scenarios such as coverage and capacity enhancement. UAVs can provide on-demand connectivity in events with vast amounts of people, like sport events, or concerts [104]. The authors in [105] explore the utilization of low-altitude UAVs that are provided with Base Station (BS) to complement the terrestrial network. Aerial BSs can also assist terrestrial networks such as vehicular networks for information dissemination and connectivity improvement, thanks to their inherent mobility [106]. In [107], the authors present a UAV-assisted cellular network with optimized spectrum sharing and cyclical multiple access, which significantly improves spatial throughput over the conventional network, although the interference control complexity was higher. Other representative application is to use UAVs as BSs for public safety applications [108] [109] such as natural disasters (e.g., earthquakes, typhoons, floods) management, where the conventional cellular network might be unavailable, damaged, or insufficient. Through different simulations, these articles show that the optimized deployment of aerial BSs at different areas may improve throughput coverage and efficiency. Another example is SARDO [110], a UAV-based cellular search and rescue solution (i.e., a UAV equipped with a light-weight BS) to localize missing victims through mobile phones.

On the other hand, UAVs are also being proposed as aerial User Equipment (UE) (known as cellular-connected UAVs) [111] [103]. Cellular-connected UAVs make use of existing cellular infrastructures and should co-exist with existing ground-based user terminals (i.e., normal users of the cellular network). This paradigm has obtained significant interest to establish reliable communications among UAVs and different devices that may be used during a complete mission (e.g., UAV-controller, UTM). Besides, considering the recent appearance of 5G networks, short term bandwidth and latency improvements are expected to be introduced once these networks have properly been deployed into the cellular landscape, which may be beneficial for multi-UAV systems for efficient fleet control [112] [113]. However, the existing infrastructure is designed and optimized for terrestrial users needs, and there are still several challenges to be considered before relying on cellular-connected UAVs as a valid solution. For example, cellular-connected UAVs require high-speed uplink connectivity, while most applications for ground users demand high-speed downlink connectivity instead. Although the downlink is critical since the UAV trajectories are transmitted in this direction, most UAV applications generate information from the UAV upon the UAV-controller (e.g., video streaming, sensors), requiring higher data rates in the uplink. Moreover, UAVs might deal with significant interference introduced by the neighboring BSs because of the nonexistence of obstacles between the multiple BSs and the UAVs [114].

Finally, a pretty well established UAV paradigm for multi-UAV missions are Flying Adhoc Network (FANET) [115] [40]. These networks are scalable due to their facility to integrate new UAVs to the system with small deployment and maintenance costs. Another of their benefits is providing feasible communications between UAVs without any additional infrastructure, similarly to the well-known Mobile Adhoc Network (MANET) or Vehicular Adhoc Network (VANET). For this reason, FANETs have been the most considered solution by the research community to enable UAV communications during the last years. For example, Chapter 2 presents a FANET scenario where we evaluate a routing scheme using an emulation platform to characterize the communication channel (WiFi) and the network mobility models. Authors in [116] describe an innovative routing process based on ad hoc networks with better efficiency than traditional FANET algorithms. Another example is [117], where position-based routing algorithms are compared over a common scenario through a comprehensive comparative analysis.

5.1.2. Virtualization and UAVs

As it has been pointed out previously, UAV provide superior mobility in comparison with traditional vehicles, allowing them to implement a wide array of applications such as automatic forest fire detention mechanisms [118], warehouse transportation, or precision agriculture [119]. As a consequence, some parts of the research community have been focusing on softwarizing different applications and functionalities provided by UAVs in order to dynamically create and deploy networking services over aerial networks. Such effort will allow UASs to remove the need for specialized equipment to execute multiple services over the network, as well as reducing both development and maintenance costs. In this regard, one outstanding example of such effort can be seen in the synergetic junction of SUAV technology together with the Network Function Virtualization (NFV) paradigm; this paradigm has risen as one of the key enablers of 5G communications, with the main objective of softwarizing physical network functions for its deployment in generic hardware equipment that matches their compute, storage, and networking requirements instead of relying on specialized hardware, which can be both expensive and difficult to maintain in most cases. In the context of aerial networks, the types of payloads that UAVs would require to be on-board in order to provide a network service over an aerial network could be reduced by relying on generic hardware instead (which is usually easier to compact in comparison with specialized equipment), which combined with their increased mobility, can lead to the development of novel applications that would be difficult to provide with current solutions. Inside this equipment, the networking services are provided using Virtual Network Function (VNF), software implementations of network functionalities (e.g., a router or a traffic generator). In order to build a complex networking service over a network, it is necessary to interconnect several VNFs in order to create a Network Service (NS), and these VNFs usually expand over several nodes that have the necessary computational, networking, and storage resources for running the service over an infrastructure (in this case, partially or fully composed by the aircraft payloads), denominating it as the NFV Infrastructure (NFVI). Some prominent examples of this combination can be found in works like [120], where authors propose a video-surveillance system for big-poorly Internet covered areas using UAVs, whose mobility allows their distribution along the desired geographical regions, using several VNFs to transmit video through a network deployed inside the devices that provide the service (i.e., the node that is part of the NFVI, which can be either the own aircraft or its payload). Using this combination, authors in [121] propose a softwarization architecture for UAVs and wireless sensors networks, illustrating this architecture with an agricultural example (where several data types regarding temperature, humidity, etc. were obtained through sensors). Other example can be found in works like [122], where the authors present an airborne computing platform design to enhance UAV functionalities. Another example is shown in [7], where we deploy an Internet Protocol (IP) telephony service though a SUAV network using NFV in order to execute the necessary networking elements to generate and distribute data between users, which could be useful in emergency situations and search-and-rescue operations for poorly covered areas.

Although the prospect of this combination is promising, some other works have explored the possible limitations of such technologies over aerial networks, where the connectivity is intermittently available due to both the nature of the communications being performed (i.e., the unreliability of wireless communications) and UAVs battery dependency, whose constant depletion (and subsequent exchange) can disrupt the links established between them. In our previous work [18], we analyzed the different issues that NFV Management and Orchestration (MANO) platforms might face in resource-constrained aerial networks, showcasing that popular solutions like Openstack might not be an optimal solution for these networks due to its high technical requirements, transport-layer limitations and its centralized approach for the management communications. Openstack is an Open Source cloud operating system in charge of managing the computational, networking and storage resources from an NFVI for the deployment of VNFs, i.e., it is a Virtual Infrastructure Manager (VIM). In order to perform this resource control, it uses a set of services for each one of its functionalities (i.e., one service is in charge of the networking resources, other for allocating computational resources, etc.), controlling them from a central node, the controller. However, all these services are generally resource demanding, which is not suitable for resource-constrained environments or intermittently-available nodes, and due to its a cloud-oriented approach it focuses on supporting Virtual Machines, and such environments with limited computational resources might not be able to efficiently use this type of virtualization for its higher resource cost.

In consequence, there has been an effort on proposing more lightweight alternatives that could both be applied in systems with lower resources or cloud environments to increase their performance without compromising their functionalities: container technology. This approach does not isolate the Operating System (OS) of the host through hypervisor technology, but rather modifies the underlaying host OS in order to separate the instances through kernel namespaces and control groups. In consequence, containers provide better performance when compared with Virtural Machine (VM)s, as they require a lower amount of resources to perform the same functions [123] [124]. In this context, we can find the open-source solution Kubernetes [125], also known as K8s, as one of the most popular platforms in both industry and academia. K8s is an open-source solution to deploy and manage applications based on Docker[6] containers allowing the managing of workloads and services, as well as their automatic configuration and deployment. Thanks to its ever-growing support due to its large community of users and developers, some researchers have already proposed solutions combining K8s with UAV technologies, as can be seen in works like [126], where authors propose a cloud-based platform for a control system architecture in order to allow controlling several UAVs despite their physical location, Ref. [127] where authors propose a cloud-native NFV-driven Intrusion Detection System using Kubernetes, and Ref. [128], where authors present an architecture to support reconfigurable multimedia services for a practical emergency environment use case of rescue operations using UAVs. Unfortunately, K8s is a resource-demanding solution since it is still cloud-oriented (i.e., its application in large data centers with full connectivity where connectivity drops are infrequent), which

---

[6] Docker, accessed Tuesday 15th November, 2022. Available: https://www.docker.com/

limits its application in further resource-constrained environments such as Internet-of-Things (IoT) environments. Hence, a lightweight version of Kubernetes called K3s[7] is currently being developed as well, reducing the K8s footprint and size for its suitability in resource-constrained environments, as authors in [129] propose using this platform to incorporate IoT devices into a smart city.

Other approaches have been proposed to deal with MANO communications in resource-constrained environments by improving their performance over "cloud-oriented" solutions, particularly the Eclipse Fog05 Fog Infrastructure Manager (FIM)[8]. Fog05 is an open-source project with the objective of deploying a decentralized infrastructure to provide and manage all the compute, storage, communications, and I/O resources from a network. One of the main advantages of this platform is its decentralized approach to resource orchestration, as it eliminates the necessity of having a central controller unit to coordinate the management between nodes, effectively distributing the orchestration across the network itself. Moreover, Fog05 has a very reduced footprint and wire overhead, allowing its use on severely resource-constrained nodes (usually present in IoT environments), as it also relies on the Zenoh protocol[9] to unify data in motion, data in-use, data at rest, and computations in a network. This protocol is considered the evolution of the well-known Data Distribution Service (DDS) protocol [130], which has seen limited implementation in UAV-oriented scenarios [131] due to its reliance on multicast technologies, which present notorious performance problems under those conditions [132]. More details on DDS technology can be found in Appendix B. However, Zenoh improves this behavior by blending the publish/subscribe model with geo-distributed storage, allowing the nodes to retrieve this information at any moment from other nodes spread throughout the network, and not only when the information is being published. Nevertheless, this platform is currently under development, so it has seen limited support at the time of writing this publication, with few examples of its application such as the one seen in [133], where authors implemented a system to provide a 360-video streaming service end-to-end across three computing tiers (cloud, edge, and constrained fog).

5.2. Testbed components description

Until now, UAVs that have been deployed forming an aerial network are typically controlled using an RLoS link. Therefore, a GCS (in charge of UAV management) should be geographically adjacent to the UAVs in order to properly establish this link for its management. In consequence, as each GCS manages its own set of UAVs, the coexistence of several UAS in the same airspace is not feasible (i.e., if a UAS is operating in relative

---

[7] Lightweight Kubernetes, accessed Tuesday 15th November, 2022. Available: https://k3s.io/

[8] Fog05, accessed Tuesday 15th November, 2022. Available: https://fog05.io

[9] Zenoh, accessed Tuesday 15th November, 2022. Available: http://zenoh.io

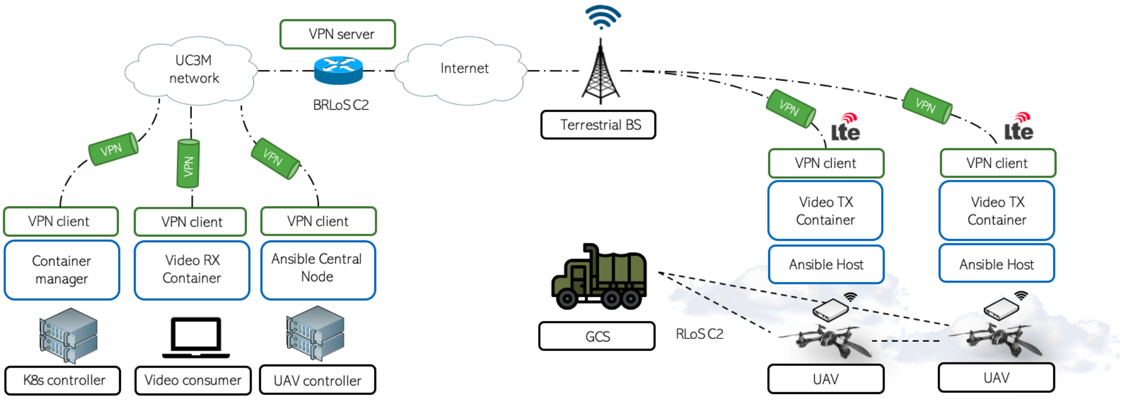


Figure 5.2: High overview of the proposed testbed.

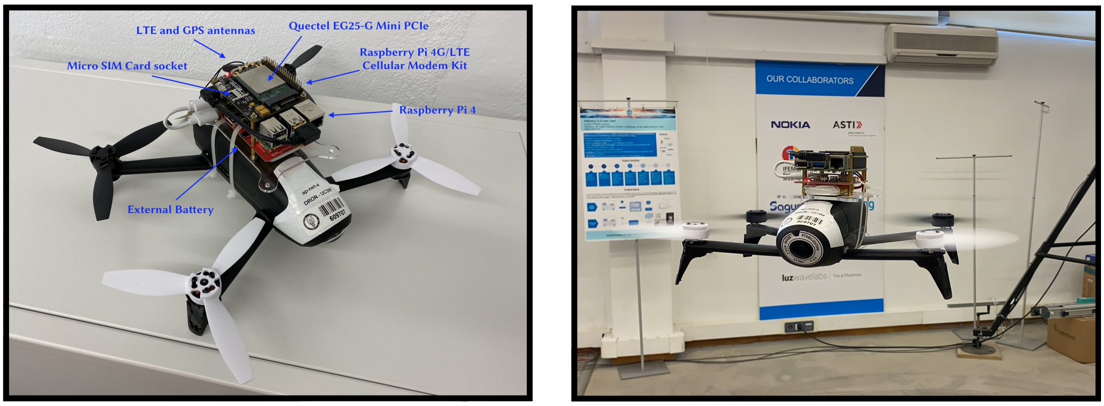
proximity to other UASs without considering the operations performed by the other UAS, the risk of having an accident is logically high).

For the correct establishment of next-future UAV applications, UASs must be connected to the UTM regardless of the technology chosen for this connection, which can be separated into three different alternatives: direct connectivity (e.g., packet network), 3rd Generation Partnership Project (3GPP) cellular connectivity, or satellite communications. This UTM manages the airspace, ensures safe operations, and computes the aircraft flying trajectories to avoid collision and accidents. The interconnection of all the elements encompassing a UAS with the UTM opens the BRLoS UAS management paradigm, producing a more automatic and agile deployment in comparison with legacy UAS scenarios and possibly allowing for an integrated airspace.

As can be appreciated in Figure 5.2, this chapter presents a testbed to experiment with new UAV services and applications managed by both BRLoS and RLoS models. However, all the experiments and efforts have been designed to properly establish the BRLoS management (RLoS models have been studied in our previous work [106] [4]). To this purpose, both the GCS and UAVs are connected to a UAV-controller (in this case, also acting as UTM) using the regular 4G/Long Term Evolution (LTE) cellular network. The following subsections describe the communication architecture, the automated UAV control system, and the virtualization platform, including the devices and technologies chosen for its design.

5.2.1. Cellular-assisted UAV communications

To support research activities, this publication proposes a novel communication model for this testbed that can be on-boarded as the payload of almost any UAV (we have used small UAVs to verify this fact). This prototype incorporates a Raspberry Pi (RPi)



(a) UAV payload. (b) UAV flying with onboarded payload at 5TONIC.

Figure 5.3: Details of UAV payload, and UAVs in flight.

4 Model B[10] Single Board Computer (SBC). The RPi can be equipped with one of these two alternatives to connect with the cellular network: *(i)* the Sixfab 3G/4G& LTE Base HAT[11], or (ii) a 5G/4G/3G HAT [12]. The RPi acts as the brain of the UAV, having the ability to host different network designs and configurations to enable its communications, i.e., this device allows the use of any protocol or communication solution to establish communications with other devices or/and the Internet, for example implementing a Virtual Private Network (VPN) tunnel. Moreover, the HATs provide the RPi with a simple interface bridge between mini PCIe cellular modems. Three different modems have been selected for this system: the Telit LE910C1 Mini PCIe LTE CAT1 Module[13], the Quectel EC25 Mini PCIe 4G/LTE Module[14] (both compatible with the Sixfab HAT), and the SIMCom SIM8200-M2 (compatible with the Waveshare HAT), allowing its straightforward configuration to start using cellular internet with the RPi. All equipment is powered with a portable battery of 3.7V and 3800 mAh. This equipment can be appreciated in Figure 5.3a.

Despite establishing connectivity with the commercial cellular network, communications cannot still be enabled between the UAV-controller and the device since the HATs assign a private IP address to the RPi, independently of the IP address that the PCIe modem uses to connect to the network (provided by the commercial operator). In consequence, all the communications between the RPi and the exterior (i.e., Internet)

---

[10] Raspberry Pi 4 Model B, accessed Tuesday 15th November, 2022, 2022. Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/

[11] Raspberry Pi 4G/LTE Cellular Modem Kit, accessed Tuesday 15th November, 2022. Available: https://sixfab.com/product/raspberry-pi-4g-lte-modem-kit/

[12] SIM8200EA-M2 5G HAT, accessed Tuesday 15th November, 2022. Available: https://www.waveshare.com/sim8200ea-m2-5g-hat.htm

[13] Telit LE910C1 Mini PCIe LTE CAT1 Module, accessed Tuesday 15th November, 2022. Available: https://sixfab.com/product/telit-le910c1-mini-pcie-cat1-lte-module/

[14] Quectel EC25 Mini PCIe 4G/LTE Module, accessed Tuesday 15th November, 2022. Available: https://sixfab.com/product/quectel-ec25-mini-pcie-4g-lte-module/

are performed using a Network Address Translators (NAT) (included in the HAT). By extension, the RPi can communicate with the UAV-controller (as the UC3M network provides public IP address), but not in the reverse direction, as private IP addresses are not directly reachable from the Internet (i.e., the NAT is preventing the UAV-controller from reaching the RPi). Unfortunately, the HATs do not allow port opening or port forwarding configuration.

In order to solve this issue, since the UAV-controller must be able to communicate with the RPi, the testbed will utilize a VPN service where all the UAS equipment is connected in order to enable a relay service. This VPN allows the communication between both elements thanks to the encapsulation, and subsequent routing, of the information that has to be sent. Furthermore, the VPN service provides additional advantages that are beneficial for the testbed infrastructure. For instance, since any UAV is able to change the BS (as a standard ground UE) due to the inherent UAV mobility, its IP address might change when this movement is performed. However, by using the address space provided by the VPN service, these handovers should be transparent for the UEs as long as the UAV has connectivity to the Internet, because the VPN will deal with the routing since the device will not modify its VPN IP address, even though its IP address (the one providing Internet to the RPi) might change. Moreover, using a VPN provides an additional layer of security, as all the communications between its elements will be encrypted to allow its transmission through the VPN tunnel, preventing man-in-the-middle attacks or the unauthorized attachment of malicious nodes to the network. Details on including data encryption, such as VPN service, can be found in our previous work (Chapter 4).

5.2.2. Automated UAV control system

In order to support UAV experimentation activities in flight, this chapter introduces and implements a system to control UAV trajectories automatically. It is essential to have a centralized system capable of simultaneously controlling several UAVs to facilitate flight operations and reducing system complexity. In the proposed solution, all UAVs can be operated from a GCS (RLoS) or from the UAV-controller (BRLoS). This procedure has been completely automated by transferring the trajectories from the UAV-controller (which is the entity in charge of the computation of such trajectories) into the RPi (in this case, located as the UAV payload) to be executed later in the UAVs. The UAV-controller is in charge of transferring the optimal trajectories to all UAVs in the scenario to avoid collisions with possible obstacles and other UAVs coexisting in the same airspace. A UTM development (meeting all the requirements mentioned above) will replace the UAV-controller functionality. Further research in this direction can be found in the literature [134] [135] [136].

The automation of this service is based on the Ansible platform[15]. Ansible is a software tool that automates software provisioning, configuration management, and application deployment. Moreover, it does not require third-party applications to establish communication channels between the central node (UAV-controller) and the hosts (UAVs), as it uses the well-known, and extensively used, Secure SHell (SSH) protocol.

On the other hand, UAVs commonly have an Application Programming Interface (API) to control them. This procedure is performed from the RPi (UAV payload) which has previously received the trajectories. One of the most popular examples is the MAVlink library [137]. MAVLink is a lightweight protocol to communicate not only with UAVs, but also with on-board UAV components. Another representative example is the Robot Operating System (ROS) [138], which is a set of building robot applications that are also employed to control UAVs. Using the specified protocols instead of proprietary protocols proffers the system huge flexibility and versatility since they have a big developer community, and in consequence, several commercial compatible UAVs become available. In our case, an open-source Python library called pyParrot[16] (compatible with a great variety of Parrot UAVs) is used to send the instructions from the RPi to the UAV. The instructions are simple commands already implemented in the library (e.g., take-off, landing, turn right, turn left, etc.) for these experiments. However, these instructions may become more complex if the received trajectories from the UAV-controller requires to. The instructions are sent to the UAV through its WiFi interface. These instructions can be transmitted using various alternatives, such as the serial port, the Universal Serial Bus (USB) port, or another communication channel. However, the available UAV in the testbed (Parrot Bebop 2) can only be commanded using WiFi.

5.2.3. Virtualization in UAVs: The power of containers in aerial networks

As explained in Section 5.2.1, UAVs will carry as their payload RPi devices in order to execute their movement instructions (sent from a remote GCS/controller), provide services to the users of the aerial network, such as those mentioned in Section 5.1 like rural monitoring, or execute specific applications depending on the mission type that has to be carried out. However, RPis are generally considered resource-constrained devices in the context of NFV environments, since the amount of computational resources that are available to the device are considerably lower as compared to other equipment types such as Mini-ITX or servers. In consequence, it is desirable to save as many resources as possible when providing services and/or running applications in order to increase the

---

[15] Ansible platform, accessed Tuesday 15th November, 2022. Available: https://www.ansible.com/

[16] pyParrot's documentation, accessed Tuesday 15th November, 2022. Available: https://pyparrot.readthedocs.io/en/latest/

available space to run new functionalities and/or scaling up operations without having to increase the number of devices in the network. Moreover, battery constrains are also a source of issues in aerial networks since RPis are forced to use a finite (and small) battery that will drain over time until its depletion, so consuming less resources can increase their operative life-time before requiring a substitution [5]. However, hypervisor-based solutions like VM (frequently used in NFV) are computationally intensive, as they need to imitate a full physical infrastructure to provide a single OS with its own kernel. Therefore, containerization can be regarded as a more adequate solution in these situations, since they run over the host Operating System to build its necessary components through kernel instructions, saving computational resources, as no hypervisor is required for the operation. Hence, aerial networks can benefit from container technology to save resources (and potentially battery life-time), making them good candidates for their implementation for service provisioning. Nevertheless, in order to build an NFV-driven (or close) solution, it is mandatory to have a control plane able to manage and orchestrate all of the containers needed for the softwarization of an application or network service, including their instantiation, life-cycle management, and logical connectivity. In this context, the Kubernetes tool (i.e., K8s) can be considered as an adequate solution for these tasks.

As it has been previously described in Section 5.1.2, Kubernetes is an open-source system for the automatic deployment, management, and scaling of containerized applications. This platform allows its users to easily deploy container images inside a cluster, which can be composed of several hosts that provide the infrastructure and resources to run the containerized applications. This container orchestrator provides several functionalities that assist the deployment of applications inside an infrastructure, including the automatic rollouts and rollbacks, ensuring that newer versions do not leave the application unavailable; self healing abilities, achieved by restoring or re-deploying containers that failed during the execution of an application; IPv4/IPv6 dual stack, horizontal scaling, which can even be automated based on CPU usage, among others. All these functionalities can be achieved thanks to its reliance on container technologies: K8s uses different container runtimes (particularly containerd[17], CRI-O[18], and Docker) to instantiate container-based applications, which are usually lighter than VMs since they are built on top of the kernel of an OS (contrary to hypervisor-based solutions where the software fully imitates the physical infrastructure where the VM is running).

In order to perform any kind of deployment inside a K8s infrastructure, it is mandatory that two components are present, and properly linked, between the hosts and/or the machines building the infrastructure, establishing a cluster. It is not mandatory for these machines to be in the same physical location and/or to have direct connectivity between

---

[17] Containerd, accessed Tuesday 15th November, 2022. Available: https://containerd.io

[18] CRI-O, accessed Tuesday 15th November, 2022. Available: https://cri-o.io

them through a single Local Area Network (LAN) or subnet. However, it is necessary to have full networking (i.e., layer 3) connectivity between all the components of the cluster, as it is specified by K8s documentation. The essential components of every K8s cluster are the following ones:

- Controller-plane: this component is in charge of the management and orchestration of the K8s cluster and the applications running inside. Its main tasks include the instantiation of pods (minimal unit where containers have to be deployed) & services, and react to cluster events such as scaling up/down a deployment, managing cluster errors, pod re-deployments, etc. This control-plane unit, commonly referred to as the “master” node, is usually deployed in one single host, although it can be deployed across multiple machines if necessary.

- Worker node: these nodes are in charge of providing the resources to the cluster. Inside these nodes, pods will be deployed to run the applications in the Kubernetes cluster, maintaining their functionality and reporting their status to the master node. In every cluster, there must be at least a single worker node, and they are not limited to a single host, i.e., inside a host there could be multiple workers (for example, multiple VMs).

Kubernetes is available for a wide array of Operating Systems such as Ubuntu 16.04+, CentOS 7+, or Fedora 25+, as well as different architectures like amd64 and arm64, this last one being typically used in mobile terminals and small devices such as RPis. Therefore, K8s is regarded as a flexible platform suitable for its use in multiple use cases that require an heterogeneous environment with multiple types of devices, including small equipment that can be on-boarded in most UAVs due to its small size and lightweight nature. This fact combined with the lightweight nature of containers compared to traditional virtualization solutions (i.e., VMs) assists in the development and deployment of network services on top of UAVs, including one of their most popular applications in their current form: video transmission.

All the aforementioned functionalities and advantages have been decisive for choosing this tool as the container manager for the implementation of the current testbed depicted in Figure 5.2. As a consequence, the RPis carried by the UAVs and the video-consumer will become the computing nodes of the infrastructure (using K8s’s terminology, worker nodes). A K8s controller (i.e., a master node) manages the instantiation, management, and termination of all the containers of an application deployed in the infrastructure. Communications between these components are entirely performed through the VPN service, ensuring both secure transmission (e.g., a man-on-the-middle attack) and preventing unauthorized users from joining the infrastructure (e.g., a malicious UAV posing as a legitimate computing node of the platform). In the particular case of the testbed, the UAVs will host a video transmission application (Video TX Container in Figure 5.2).

Video transmission has been one of the main use cases of UAVs since their inception. Thanks to their advanced mobility, these aircraft are able to on-board different types of cameras to take pictures and/or video footage. Moreover, some models are shipped with one, or several, integrated cameras by default as a consequence of its high demand in both industrial and commercial environments for a wide variety of applications such as video filming, wild life monitoring, surveillance, or search and rescue operations. In consequence, this testbed will introduce UAVs able to generate and send video traffic to a particular machine or client, either localized inside the testbed (video-consumer in Figure 5.2) or outside. This application will be deployed as a single container running in the worker nodes located at the RPi, where the video traffic will be generated. These nodes are represented as the UAVs in Figure 5.2. Afterwards, this video will be sent to a specific location, either to nodes located inside the cluster (e.g., the video-consumer) or located outside the cluster (e.g., a Personal Computer (PC) at a remote location). If the UAV goes offline at any moment in time, either due to a battery replacement cycle or connectivity issue, K8s will try to instantiate this application in a new UAV or, if no others are available, wait until one of them is available to re-deploy the container again, showing the potential of K8s to reduce service cut-offs in aerial networks.

5.3. Testbed experimentation

5.3.1. Communication performance evaluation

In this section, the UAVs communication performance has been evaluated to guarantee that it meets the requirements defined by the 3GPP in Release 15 (TR 36.777) [139]. These requirements are based on three network parameters: *(i)* data rate, which should be set between 60 and 100 Kb/s for command and control in both downlink and uplink; *(ii)* reliability, which should be below 10^{-3} of packet error rate in both downlink and uplink; and *(iii)* latency, which should be below 50 ms (100 ms of Round Trip Time (RTT)) in the downlink. It should be noted that these Key Performance Indicator (KPI)s are related to management assignments (C2). Each particular service will have its individual KPIs (for example, real-time video streaming KPIs are quite different).

We have first tested the quality of the Internet connection with the well-known Ookla speed test[19]. The speed test uses public servers to perform a quick and realistic diagnosis of Internet connectivity. In this context, the network performance was also tested from the UAV upon the UAV-controller since, in practical experimentation, the minimum values have to be met against this entity. Other limiting factors may arise in the second

---

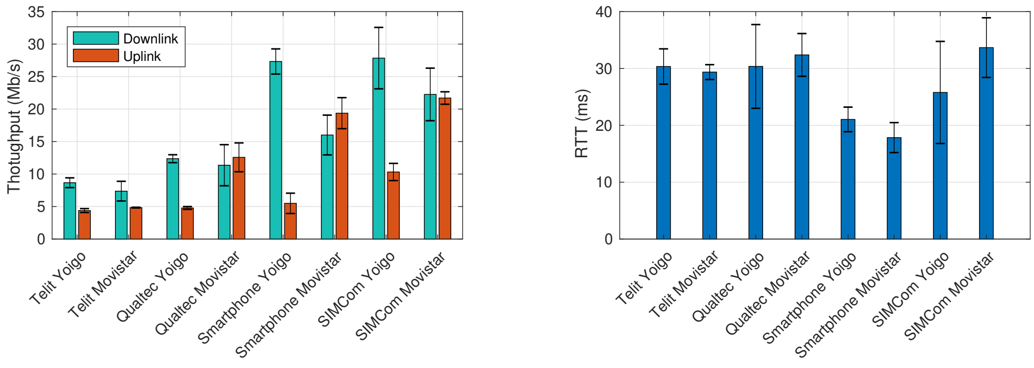
[19] Ookla speed test, accessed Tuesday 15th November, 2022. Available: https://www.speedtest.net/

experiment, such as using a VPN service (since all the traffic should traverse the VPN server).

Experiments have been replicated with the three available mini PCIe and a commodity smartphone (Xiaomi Redmi Note 9S). The purpose of repeating the same experiments employing different devices is to determine if the potential limiting factors are procured from either the cellular network characteristics, e.g., the status of the network, the antennas used, the strength of the signal at certain points of time, or the devices themselves (i.e., checking if the devices could be limiting the quality of the connection due to a hardware/software limitation). Likewise, experiments have been conducted using two different Internet Service Providers (ISP) to check if connectivity performance varies depending on the network operator. The selected ISPs are Movistar and Yoigo (the first and the fourth operator in Spain ranked by the number of users). Twenty repetitions of the experiment have been conducted at different time intervals randomly distributed throughout the day to ensure that the external effects are not reflected (or at least mitigating their effect as much as possible) in the results. The main purpose of these measurements is not to characterize the network, as this would require an extensive set of measurements with varied coverage areas and having an insight of the network infrastructures involved to rule out any other factors having an effect on the results achieved. The main purpose instead is to verify whether the results fulfill the 3GPP KPIs.

As it can be appreciated in Figure 5.4a both downlink and uplink are considerably limited for the Telit mini PCIe due to its technical characteristics, which define a nominal rate of 10 Mb/s in downlink and 5 Mb/s in the uplink, regardless of the selected ISP. On the other side, experiments with Qualtec mini PCIe should not be so limited since its nominal rate is 150 Mb/s and 50 Mb/s in downlink and uplink, respectively. However, after comparing with the smartphone and SIMCom PCIe results, both the downlink and uplink appear to be limited despite its technical characteristics (i.e., it has been demonstrated with the SIMCom results that the performance of the network is superior to that obtained by Qualtec PCIe). Finally, the SIMCom PCIe results improve the results obtained with the smartphone, which can be considered the benchmark. It should be noted that the Waveshark HAT, in combination with the SIMCom modem, has a notably higher price than the rest of the selected equipment, so we can infer that the performance is better because it belongs to an upmarket product. On the other hand, RTT experiments (Figure 5.4b) reveal that all devices and ISPs fulfill 3GPP KPI requirements. Please, take note that the error bars correspond to the standard deviation.

This experiment also reveals that the Movistar network has balanced downlink and uplink data rates; meanwhile, Yoigo prioritizes downlink data rates. For selecting a final prototype, it is important to consider that UAVs potential applications will overload mostly the uplink (i.e., the highest data rate will be generated on the UAV and sent in



(a) Throughput (b) RTT

Figure 5.4: Performance analysis using Ookla speed test.

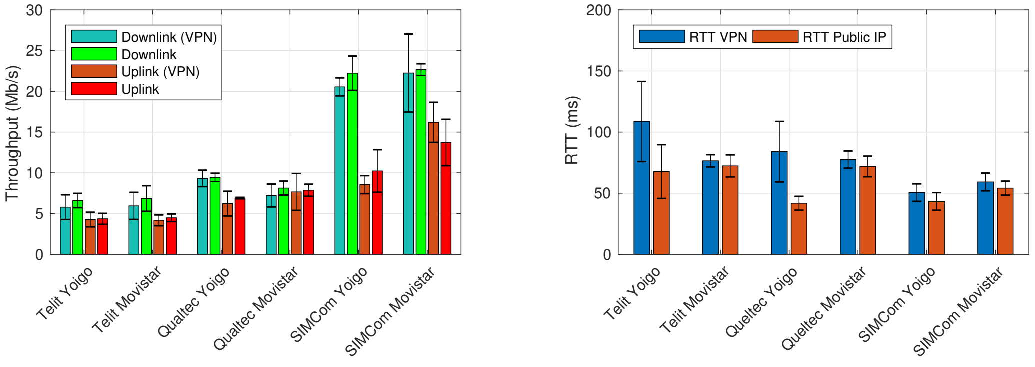
the uplink). However, all the combinations (mini PCIe and ISPs) comply with the 3GPP KPIs.

In the second set of experiments, we have measured the performance from the RPi upon the UAV-controller, enabling and disabling the VPN service to quantify the performance penalty that the VPN may introduce. Throughput results are similar in the second experiment to those obtained with the Ookla speed test as represented in Figure 5.5a. As expected, there is a slight decrease in the performance since data packets are not only traversing the VPN server but also have to include the VPN headers (decreasing the data payload size). However, the values are still much higher than those required for C2. On the other hand, the RTT results (Figure 5.5b) are considerably worse, although they are within the requirements set by the 3GPP KPIs. This phenomenon occurs because of the VPN encryption and decryption process. The VPN relay is not greatly affecting the performance since the VPN server is located on the same network as the UAV-controller. More details about the performance decrease of introducing a VPN service can be found in chapter 4. Reliability tests were also performed by sending a 200 Kb/s stream as required in the KPIs (both downlink and uplink), and no packets were lost during this process.

It can be concluded that all the devices fulfill the management requirements. Other services can be provided using this communication solution, such as transmitting periodic High-Definition (HD) images or real-time video streaming (HD 720p). Still, some challenges may arise when using this technology for complex multimedia applications or multiple services simultaneously.

5.3.2. 5G Standalone benchmark

By December 2020, Ericsson has activated 5G SA technology at 5G Telefonica Open Network Innovation Centre laboratory (5TONIC) [140]. 5TONIC is a leading



(a) Throughput (b) RTT

Figure 5.5: Performance analysis upon the UAV-controller.

research laboratory focusing on 5G technologies based in Madrid, where members from both industry and academia work together in specific research and innovation 5G experimentation projects. The deployment of 5G SA in this laboratory is a crucial technological milestone for validating the solutions developed by 5TONIC verticals partners allowing promising capabilities enabled by 5G technology demonstrations. This deployment includes the integration of the Ericsson Radio Access Network (RAN) [141] and the Core solutions for 5G New Radio (NR) standalone[20] in the laboratory facilities. With this achievement, 5TONIC becomes one of the first open research and innovation laboratories in the world to offer 5G SA capabilities to validate vertical applications.

Due to the lack of commercial deployment of this technology, the 5TONIC 5G SA network has been used to benchmark the network performance offered by a real 5G SA deployment. Although these measurements have been conducted in a laboratory environment with ideal conditions (compared to Section 5.3.1 measurements), they demonstrate the upcoming 5G SA technology potential.

The NETGEAR Customer Premises Equipment (CPE)[21] was used to perform these experiments. This device connects through the Ethernet interface with the RPi to provide connectivity. Thanks to its compact size (10.5 cm × 10.5 cm × 2.15 cm) and low weight (240 g), this device can be on-boarded on almost any UAV, as shown in Figure 5.3b. Moreover, it includes a rechargeable battery promising battery life-times for one day at total operation. As in the previous section, the Ookla speed test has been used to diagnose Internet quality. We obtained an average value of 387.98 (±90) Mb/s in the downlink and an average value of 52.3925 (±5) Mb/s in the uplink. RTT results are 11.94 (±3.5) ms. These results vastly improve the ones obtained by a commercial

---

[20] Ericsson's dual-mode 5g core, accessed Tuesday 15th November, 2022. Available: https://www.ericsson.com/en/core-network/5g-core

[21] Nighthawk M5 5G WiFi 6 Mobile Router, accessed Tuesday 15th November, 2022. Available: https://www.netgear.com/home/mobile-wifi/hotspots/mr5200/

4G/LTE network. Accordingly, this technology has to be contemplated for the near future and can be considered as the enabling technology to deploy the full potential of UAV networks. However, an exhaustive performance evaluation will have to be carried out with commercial deployment.

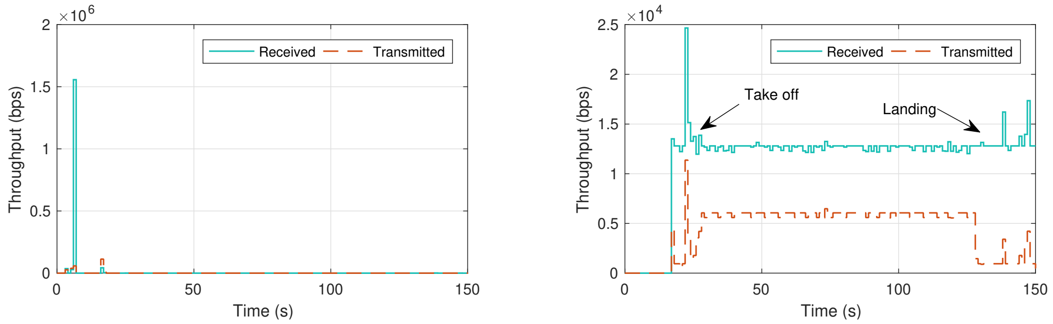
5.3.3. Indoor flight validation

Indoor flight tests have been carried out to validate the payload as an operation onboarded system (no outdoor flight has been done due to regulatory restrictions). These results serve as a Proof-of-Concept (PoC) demonstrating the functionality of the proposed testbed. For this purpose, indoor tests have been carried in the 5TONIC laboratory. On the other hand, the K8s controller, Video-consumer, and UAV-controller elements (see Figure 5.2) are located at UC3M. In this way, an inter-site experiment (as introduced in chapter 4) is performed to demonstrate that the BRLoS management operations run correctly. As it can be appreciated in Figure 5.3a,b, the communication prototype (RPi + Qualtec PCIe) has been on-boarded as an UAV payload, using a compact battery to power up the prototype. The overall dimensions of this prototype are $6 \times 11 \times 4$ centimeters, with a total weight of 200 g including the battery. It is essential to consider lightweight and compact equipment since the power required by the UAVs to carry all its load can significantly impact the life-time of the UAV main engines battery.

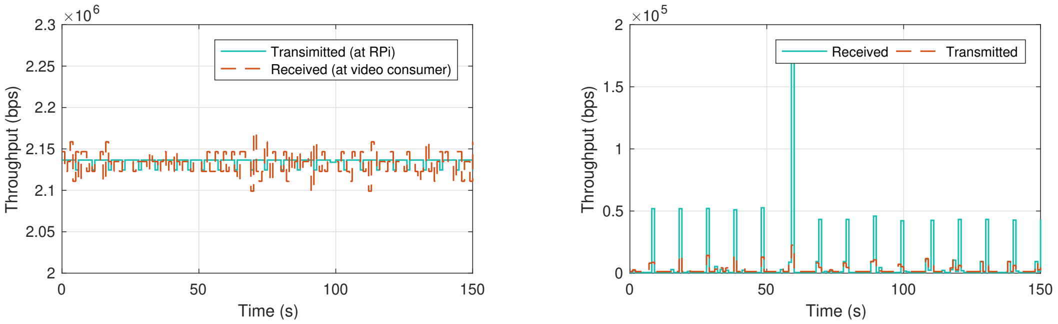
It is essential to consider that Multi-UAV operations entail certain risks, not only in colliding with aerial traffic (i.e., other UAVs, human-crewed aircraft in the proximity of airports), and accidents including people and vehicles on the ground in populated areas. Authors in [142] proposed a complete risk assessment model for UAV operation in urban environments with three main risk categories (people, vehicles, and human-crewed aircraft). The mission planner generally contemplates the operations and factors involving the most significant risk. However, unbounded exploration can lead to undesirable scenarios. To solve that issue, authors in [143] propose a framework that evaluates the risk of the potential actions. In case of actions deemed too risky, they are replaced with another action of lower risk.

C2 information has been transmitted from the UAV-controller to the RPi (UAV payload) using the Ansible automation platform explained in Section 5.2.2. Once the payload has received the trajectory, it replicates the instructions to the UAV using its WiFi interface (also explained in Section 5.2.2). This traffic is represented in Figure 5.6a,b. Figure 5.6a shows a traffic peak corresponding to the trajectory transfer. On the other hand, Figure 5.6b displays the transmitted and received traffic between the RPi and the UAV. The two traffic peaks that can be appreciated correspond to the take-off and landing procedures.

On the other hand, the testbed virtualization platform has been used to test multimedia services viability. A docker container deployed on the RPi (Video TX



(a) Ansible C2 at UAV (RPi)
(b) pyParrot C2 at UAV (RPi)



(c) Video from UAV (Docker at RPi)
(d) K8s management traffic

Figure 5.6: Measurements in indoor flight.

container in Figure 5.2) transmits a 2Mb/s User Datagram Protocol (UDP) traffic flow to a docker container deployed on the video-consumer. This stream emulates an HD video transmission (HD 720p [144]), a target rate that most HD cameras are able to generate. All this traffic has been sent using the VPN tunnel described in previous sections, since it provides an extra security layer to the information being transmitted (i.e., preventing third-party attacks that could compromise the video feed in real scenarios). Figure 5.6c reveals that the video-consumer has correctly received the flow.

Finally, the traffic transmitted and received for the correct management of the virtualization platform is shown in Figure 5.6d. This traffic should not be neglected as it shares the same channel as other critical communications such as C2, and its presence can disrupt other traffic if it requires a high amount of bandwidth for its proper functionality. However, it is relatively small in comparison to the available bandwidth of the channel, with an average of 2.71 Kb/s for transmitted management traffic and 6.52 Kb/s for received management traffic (representing less than 1% of the available throughput), so it is unlikely that this traffic will interfere with C2 communications and other applications embedded in the UAV payload.

5.3.4. Indoor flight with connectivity loss

Once the feasibility of the testbed was demonstrated in the previous Section 5.3.3, the experiment is repeated with the introduction of a connectivity loss between the UAV and the UAV-controller. UAVs must have some degree of autonomy when communications are lost with the UTM in order to avoid accidents or collisions with other aircrafts, since losing connectivity with this element could be disastrous for the UAV if no alternative method to calculate its flying trajectory is available, and it is important to have in mind that communications depend on an external network that can introduce errors and temporary disconnections to the UTM. In order to reproduce this situation, the interface that provides connectivity to the RPi is switched off. Figure 5.7c,d shows that the connectivity is lost around second 110.

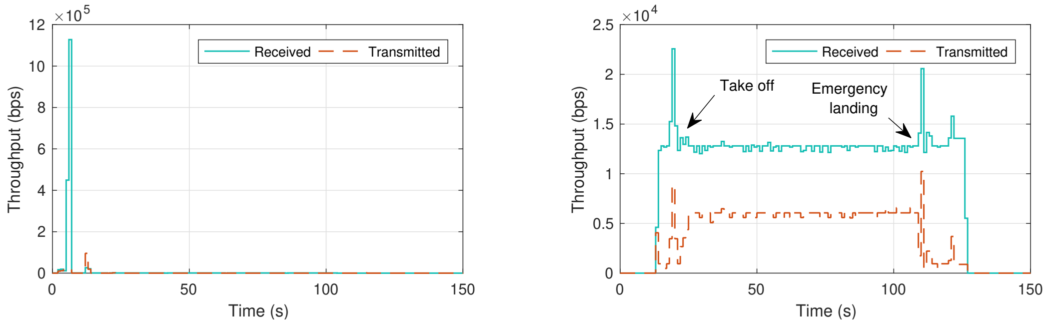
In the meantime, the RPi is continuously monitoring the connectivity with the UAV controller. After detecting a connectivity loss, it waits for a predetermined time (set through a programmable parameter) to check if the communications have been recovered. As it is not the case, the UAV performs an emergency landing routine, as shown in Figure 5.7b. This instruction can be modified by any other procedure, such as a Return To Home (RTH) instruction, or remain static (hovering) in a certain position waiting for new communications from the UTM. However, it is essential to point out that the UAV must have some autonomy in this decision-making process, ensuring that all these instructions can be safely performed and, in addition, using detect and avoid mechanisms to evade obstacles while executing these instructions.

5.4. Discussion

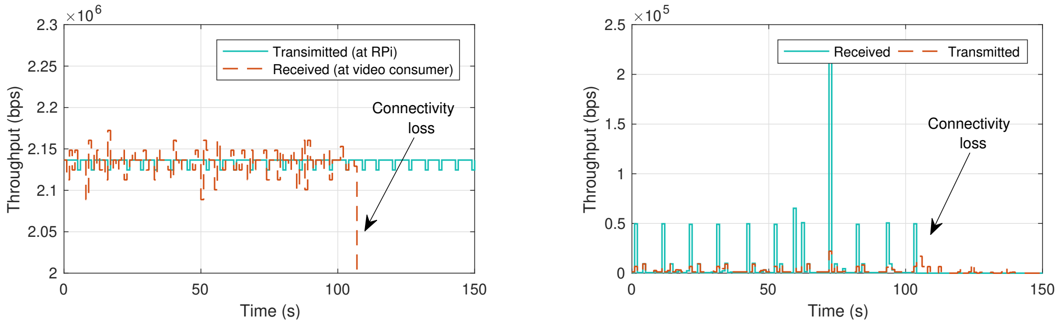
This chapter presents as the main contribution a practical cellular-assisted UAV to enable BRLoS traffic management in the context of the Labyrinth project. This testbed will be employed to conduct validation analyses within the Labyrinth project.

Different experimentation using this testbed has also been presented: on the one hand, it has been demonstrated that the selected communication equipment and the testbed design fulfill their purpose and meet the KPIs set by 3GPP for a proper BRLoS operation. However, the results reveal that the 4G/LTE commercial network may have some limitations for applications beyond C2, while the conclusions obtained from the 5G SA benchmark reveal promising results in terms of bandwidth and latency. On the other hand, indoor flight experiments have been carried out to guarantee the platform's viability and functionality.

Besides, the testbed incorporates a virtualization platform to deploy a wide variety of experiments over the network. This platform improves on traditional virtualization platforms due to its use of containerization technologies, a more lightweight solution for virtualized environments, making its implementation suitable in an environment with



(a) Ansible C2 at UAV (RPi)
(b) pyParrot C2 at UAV (RPi)



(c) Video from UAV (Docker at RPi)
(d) K8s management traffic

Figure 5.7: Measurements in indoor flight with connectivity loss.

resource-constrained devices and/or intermittent connectivity.

This work serves as a starting point for several lines of work and research. First, it would be essential to conduct performance measurements using 5G cellular networks once the commercial deployments are extended. Although 5G characteristics are promising, it will be necessary to exhaustively analyze its output performance to illustrate its potential strengths and weaknesses. On the other hand, it would be interesting to include energy consumption measurements about the communications prototypes. In resource-constrained environments like UAVs, where a battery supplies power to the on-board equipment, it is crucial to properly quantify power consumption in order to calculate its impact on battery life-time (which is very low due to the high energy consumption demanded by flight engines).

5.5. Research impact/dissemination

The outcome of the work behind this chapter has resulted in a publication in the MDPI *Sensors* journal. MDPI *Sensors* has an impact factor of 3.576, and it is ranked as Q1 in the Journal Citation Reports (JCR). This article has a citation from another research article.

This work results from a collaboration with the Telefonica R&D department in the framework of the H2020 Labyrinth European project. Some content of the chapter has been included in the Deliverable 4.1 ("Communications Platform Basic Analysis") of the same project.

PART IV

CONTRIBUTIONS TO CONNECTED UAVS

Having analyzed some critical challenges that must be solved in the Unmanned Aerial Vehicle (UAV) area, especially in multi-UAV scenarios where the complexity of the solutions is exponentially exacerbated, this part proposes different solutions to various challenges.

Chapter 6 explains the problem of the limited autonomy of UAV batteries (around 20 minutes in commercial UAVs) when they are used to provide long-duration services (in the order of hours) and proposes a UAV replacement algorithm that, on the one hand, maximizes the service time while, on the other hand, minimize the number of UAVs required to provide such service.

Besides, chapters 7 and 8 present the Interface Manager (IM) and the Communication Infrastructure Manager (CIM), software entities that allow communication between the different domains encompassing an Unmanned Aircraft System (UAS) (e.g., UAV, Ground Control Station (GCS), UAS Traffic Management (UTM)) using the most appropriate communication alternative according to the instantaneous state of the network. An Software Defined Networking (SDN) approach is used to achieve this process.

6

Energy-aware management in multi-UAV deployments: Modelling and strategies

The unstoppable growth of the Unmanned Aerial Vehicle (UAV) (commonly known as drones) ecosystem during these last years, has been proven to be just the beginning of a near-future global phenomenon. The US Federal Aviation Administration predicts[1] that UAVs providing commercial services will triple over the next five years, and will overtake consumer off-the-shelf UAVs by the year 2024. UAVs will grow eightfold over the next decade and will become the largest segment of the civilian market.

Fifth Generation of cellular network technology (5G) UAV missions, e.g., to complement existing cellular networks in high-density environments, deliver network coverage in hard to reach rural areas (Remote Access Networks), or in Internet-of-Things (IoT) scenarios, may require the management of moderately large UAV fleets. UAVs mainly act as aerial communication platforms such as *(i)* aerial Base Station (BS) (to support existing 5G infrastructure in high traffic demand) [145], or *(ii)* aerial WiFi Access Point (AP) forming a Flying Adhoc Network (FANET) (to create new networks) [146]. So that UAV research area aims to extend the 5G network (where it has no range) or support the existing 5G network (when it is not enough) using radio solutions as payload [147], e.g., 5G or Long Term Evolution (LTE) microcells, multi-hop solutions based on commodity WiFi. When compared to terrestrial antennas, aerial units may have some advantages since they can change their altitude with the possibility of avoiding obstacles, including no geographical restriction on the antenna location. However, these advantages turn into crucial design challenges, such as the optimal positioning, the limited flight time, or the optimal trajectories calculation and network planning [37].

In particular, multi-UAV environments may give rise to long-endurance missions that require uninterrupted service provisioning (performing UAV replacements) that are not achievable using a single UAV due to battery capacity constraints (at most around 20 min flight[2]). A UAV replacement means that a UAV that is waiting in the Ground Control

---

[1] F. A. Administration, FAA Aerospace Forecast, 2019, accessed Tuesday 15th November, 2022. Available: https://www.faa.gov/

[2] Parrot Bebop 2, accessed Tuesday 15th November, 2022. Available: https://www.parrot.com/es/

Station (GCS) becomes active and goes into the scenario to substitute one of the UAVs that are on service, so as to provide the same functionality. This is only possible by providing a fleet exceeding the number of UAVs that have to be active on service at the same time, i.e., there is a reasonable number of fresh UAVs for replacement.

Nevertheless, developing an appropriate replacement strategy of UAVs, is one of the critical hurdles that have not yet been properly addressed by the research community. The replacement strategy enables to optimize the cost in terms of required aerial infrastructure resources while keeping the provided level of service. To guarantee that long-term (beyond battery life) services can be deployed, some of the UAVs that are at the GCS must be able to successfully replace the UAVs that provide the actual service on the stage whenever necessary (for example, when a UAV has low battery or fails). However, the economic cost of oversizing the fleet is enormous, as these devices commonly have high prices. For this reason, it is necessary to develop a resource optimization mechanism in order to allow intelligent and autonomous UAV systems to be managed with the lowest possible number of UAVs.

Figure 6.1 illustrates a representative use case of UAVs delivering network coverage. As it can be appreciated, some UAVs provide connectivity to several end-users. A Controller entity, located in the GCS, is in charge of scheduling when UAV replacements will take place. Once the replacement procedure is started for a certain UAV, that UAV directly goes back to the GCS to change its battery ①, while another one comes back in its place ②. As soon as it has a charged battery installed, it is available again in the replacement pool for other UAVs to be changed when required. By following this methodology, an uninterrupted service may be provided. Reducing the UAV fleet while ensuring a reasonable quality of service is not a straightforward procedure (Further details about the methodology depicted in Figure 6.1 can be found in Section 6.3, Sections 6.3.1 and 6.3.4).

This chapter presents the practical UAV replacement problem and analyses its complexity. Next, different scenarios, as well as the methodology to evaluate the service performance, are presented. A sub-optimal heuristic algorithm is proposed that guarantees the proper modeling and control of a fleet of UAVs that are used to provide Internet connectivity minimizing the fleet size. This algorithm is validated comparing it with the optimal solution, using an improved version of the brute-force search combinatorial algorithm developed in our previous work [5]. Finally, the practical limitations of the proposal are analyzed, while the possible alternatives solutions are considered.

The rest of the chapter is organized as follows: the related work and background are reviewed in section 6.1. Section 6.2 states the proposed problem and analyses its complexity. Section 6.3 describes the methodology to operate with large multi-UAV

---

drones/

fleets. Then, section 6.4 details the suggested scenarios and remarks the results obtained from the simulation. Finally, section 6.5 concludes the chapter and depicts some future research lines.

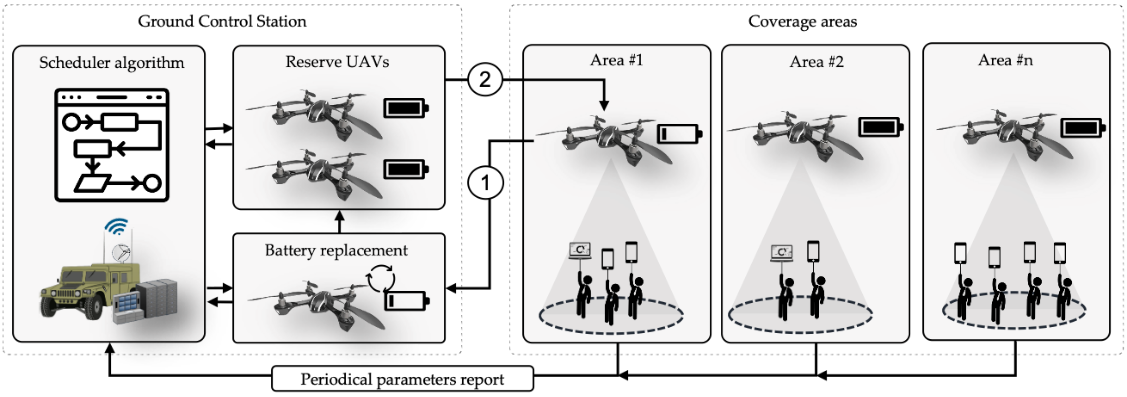


Figure 6.1: Typical UAV use case using the proposed methodology.

6.1. Related work and background

Due to the versatility of UAVs, these devices are used in a wide variety of fields. The following section introduces the evolution of management strategies used to overcome battery limitations and algorithmic solutions for UAV battery replacements which are the main topics of this chapter. It also explains how UAVs may complement 5G networks, and wireless communication solutions in large geographical areas using UAV swarms.

Ubiquitous connectivity is one of the current challenges of 5G networks and beyond 5G [37]. UAVs have appeared as a promising solution to provide reliable and flexible wireless communication services for ground users in a wide variety of scenarios [145]. The usage of UAVs promises to provide cost-effective wireless connectivity for devices without infrastructure coverage. Concretely, UAVs are considered as flying BSs for coverage extension and capacity enhancement of the existing 5G cellular networks. In this paper [109], authors explore the use of UAV-BSs to provide coverage during natural disasters. In this work [147], an Evolved Packet Core (EPC) inside a UAV is introduced, to orchestrate the LTE Radio Access Network (RAN) in the presence of multiple BSs. This EPC can also interoperate with commercial BSs as well as commodity user equipment. In [148], the authors provide an overview of UAV-aided networks, introducing the underlying architecture and wireless channel characteristics.

One of the most critical design challenges in multi-UAV systems is the achievement of the all-to-all communication between UAVs, which is necessary for cooperation and collaboration [149] [150]. If every UAV is connected to existing network infrastructure such as a GCS, satellite network or base stations, swarm communications can be delivered via this infrastructure. This type of network scheme simplifies some problems that may be associated with UAVs ad hoc networks alternatives, like routing protocols or the distributed control of the network. However, it also brings as a consequence certain limitations such as the expensive equipment (long-range or satellite antennas) and obviously less flexibility since the deployment is fixed to existing infrastructure. An alternative solution is the usage of FANETs. In this type of system, UAVs have several roles, not only as functional devices to provide coverage, gathering sensor data, or video dissemination but also to be used as network relays to connect all UAVs through the UAV network itself. Commonly, only one (or a few) UAV (also known as backbone UAV) are required to be connected to the fixed infrastructure (GCS). The backbone UAV is generally equipped with two radios: *(i)* low-power radio (WiFi or Bluetooth, for instance) is used for communication between the UAVs and *(ii)* high power long-range radio to communicate with the GCS [151]. It is common to find quite a few examples of research works that use FANETs to support 5G networks [40]. For instance, Reference. [146] extends a 5G network slice for video monitoring with a FANET composed of small low-altitude UAVs with Multi-access Edge Computing (MEC) facilities to allow high-speed transmission.

Although the development of UAV networks is receiving significant attention from the research community, some challenges must be solved before their proper deployment and consolidation. One of them is their limited battery capacity since normally a UAV source power mainly depends on small batteries (we are considering in this chapter small rotary-wing UAVs and not big fixed-wing UAVs with fuel engines). Consequently, these Small Unmanned Aerial Vehicle (SUAV) are hardware-constrained devices that cannot be too heavy or carry heavy payloads. Besides, to the power consumption of the flight engines, it is essential to consider the additional energy required by onboarded computers, that may not be carrying their own external batteries and in case they were, extra weight would be added to the system. As a consequence, we find that the useful lifetime of a UAV system is undoubtedly limited by these restrictions. Different research works propose solutions to provide uninterrupted service on long endurance missions and overcome the reduced-battery challenge. For instance, Ref. [152] presents an algorithm to offer continuous structural inspection services using UAVs not only through simulation results, but also by using an implementation. In this solution, authors replace a UAV unit before its battery is drained. The replacement algorithm employed in this article will be used later in this chapter to compare and contextualize the proposed solution. The modification of their methodology (since in this particular case, authors work with a single-UAV system while the proposed scenarios require moderately large fleets of UAVs) is explained and detailed in Section 6.4.2.2. In [153], authors consider UAV replacement (among other possible alternatives, such as refueling [154] or recharging) to maintain total surveillance of an area perimeter. Additionally, some articles propose an automatic battery replacement [155] [156] [157]. They offer a GCS capable of swapping

UAV batteries without human interaction. Ground task automation not only reduces human interaction but also increases the multi-UAV system operation area, improving the coverage and enabling operation in hazardous environments. This trend makes us choose battery replacement as the preferred option in the solution proposed in this chapter. Battery price is considerably lower than the cost of a UAV, and the time to replace the battery is remarkably shorter than the time to recharge it. Moreover, thanks to these studies and their practical experimentation, we use these results as input for our scheduling algorithms to provide accuracy to the design of UAV replacement strategies. Diverse works attempt to solve the limited battery life problem which is inherent to current SUAVs by proposing diverse alternatives. In chapter 2, it is considered that the UAVs land to provide service (if possible and secure operation). The work in [158] summarizes different techniques to prolong the UAV operation time from Battery dumping [159] to Photovoltaic arrays [160] [161]. Some other additional techniques have been proposed like wireless charging using lasers is in [162].

The optimization field, to improve the restricted communication performance of UAV networks while using the minimum amount of physical resources, is also an actual discussion topic in state of the art. In [163], the effective use of flight-time constrained devices is investigated, maximizing the average data service to ground users following a fair resource allocation policy. The solution of the cooperative allocation problem proposed in [164] significantly improves the performance of several network parameters. In [165], the authors try to minimize the number of vehicle-mounted BSs required to guarantee wireless coverage for a group of distributed ground users. Similar work in [166] proposes a placement algorithm for vehicle-mounted BSs that maximizes the number of covered ground users using the minimum UAVs. In [167], authors investigate the UAV coverage problem and propose a multi-UAV coverage model based on energy-efficient communication. The work in [168] [169] focuses on the application of a multi-layout multi-subpopulation genetic algorithm achieving significantly better performance results than the other meta-heuristic algorithms also considered to improve the coverage deployment of multi-UAV networks. An explicit definition of the minimum-energy paths between a predefined initial and final configuration of a quadrotor by solving an optimal control problem concerning the angular accelerations of rotors is detailed in [170]. Their solution yielded minimum-energy and fixed-energy paths for the aerial vehicle.

6.2. Problem statement

As it has just been mentioned, one of the main challenges in multi-UAV systems is to keep all the target geographic areas covered overtime by UAVs, since their battery life is limited (minutes) as compared to the typical mission timelines (hours). In order to face this problem, our approach is to use a fleet with the number of UAVs that are

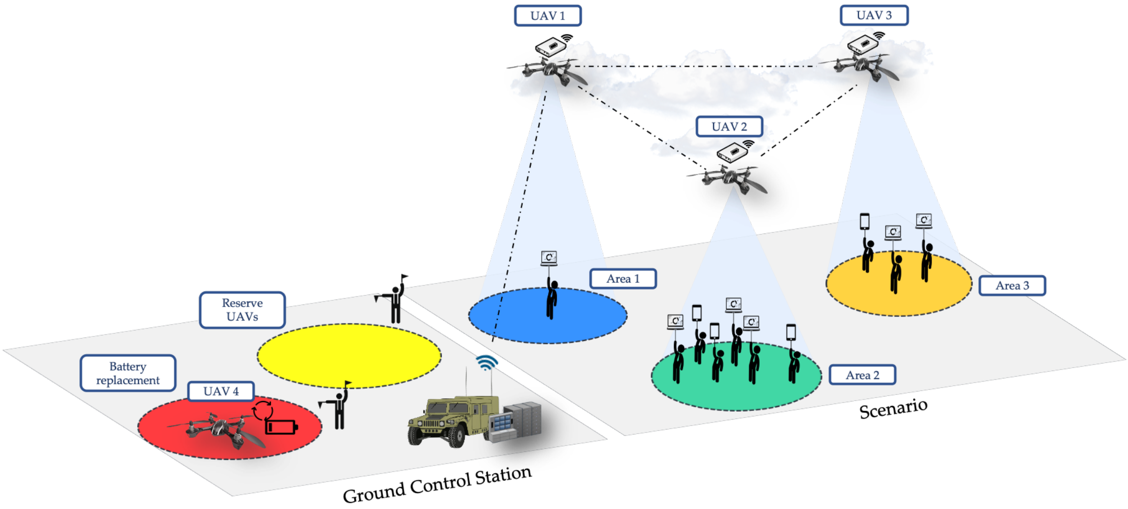


Figure 6.2: Multi-UAV system during a mission for three target areas ($j = 3$) and four UAVs ($i = 4$).

required to cover the whole scenario, and then maintain extra UAVs in a backup pool to serve as replacement units (as it can be seen in Figure 6.2). Once a replacement has been scheduled by the GCS, a fully recharged UAV enters the scenario while the replaced UAV goes back home to substitute its empty battery and be therefore ready to be changed by the next active UAV that requires a replacement. However, this procedure of identifying the minimum number of necessary (extra) UAVs and scheduling UAV replacements in the appropriate moment (to guarantee a minimum level of service availability) resembles a sophisticated approach and is the main problem that is treated in this chapter.

Figure 6.2 depicts the reference scenario considered in our analysis. In this scenario, different colors are used to represent different geographic areas, which encompass the target areas where UAVs are intended to provide network coverage to end-users, the geographic location where UAVs are directed for a battery replacement, and the specific area where the backup pool of UAVs is kept for subsequent use. These color patterns have been reproduced in Figure 6.3 to classify not only what task the UAVs are doing at a certain moment but also to show which UAVs are covering the target areas at any given time. The following subsection faces the practical UAV replacement problem from an optimization viewpoint stating a simplified and manageable procedure, checking its complexity, and solving it through different approaches (optimal brute force algorithm, heuristic algorithm).

6.2.1. Complexity analysis

In this subsection, we prove that a simplified version of the proposed problem maps to an NP-hard problem (bin-packing problem [171] in this particular case) so that we are

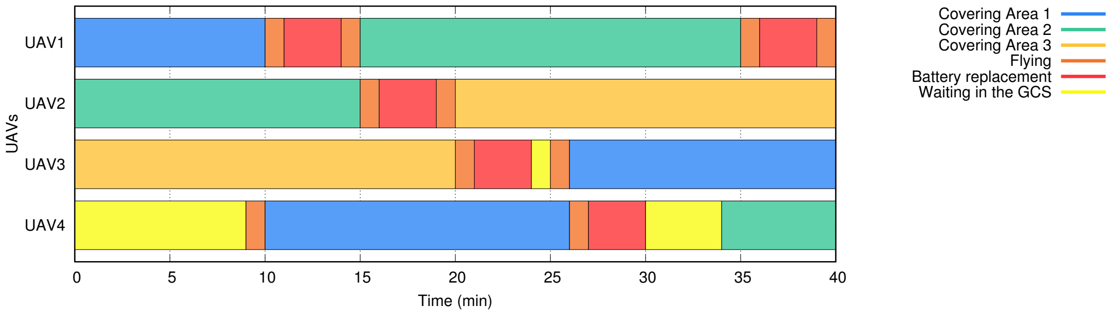


Figure 6.3: Multi-UAV system states during a mission for three target areas ($j = 3$) and four UAVs ($i = 4$).

able to state its complexity. We denote a UAV using the index i and the target areas using the index j. Each UAV i has $C_{B_i}(t)$ battery level at instant t and the UAV might be in four different states: *(i)* battery replacement state (landed in the GCS), *(ii)* flying state (towards the GCS, or towards a region where it is intended to provide a network service), *(iii)* covering a region, or *(iv)* waiting in the reserve UAVs area to replace an active UAV. These four states can be appreciated in Figure 6.3. This diagram represents a hypothetical scenario with three regions ($j = 3$) and 4 UAVs ($i = 4$), also indicating when the replacements take place to guarantee system availability over time. Note that a region/area indicates where a UAV has to fly. In the case (e.g., the number of users, a high volume of traffic) that two UAVs have to be geographically near, we consider two different areas.

This problem statement must guarantee each region j is always covered by a UAV, i.e.,

$$\sum_i x_{i,j}(t) = 1, \quad \forall j, \forall t \tag{6.1}$$

with $x_{i,j}(t) = 1$ whenever UAV i covers region j. Moreover, at any time t, the battery level of UAV i has to stay above a safe threshold a_j (e.g., 20%) for each region, so as to ensure the flight back to the GCS:

$$\sum_j a_j x_{i,j}(t) \leq C_{B_i}(t) y_i(t), \quad \forall i,\ \forall t, \tag{6.2}$$

here $y_i(t) = 1$ whenever UAV i is not in the GCS.

Additionally, battery levels keep on decreasing while UAV is covering a region. Otherwise, we consider its battery levels is set to 100% once it has returned to the GCS, and the operator has replaced the battery:

$$C_{B_i}(t+1) = \left(C_{B_i}(t) - c\right) y_i(t) + R_{T_d,T_r}\left(1 - y_i(t)\right), \quad \forall i,\ \forall t. \tag{6.3}$$

with c the battery consumption, and R_{T_d,T_r} the average battery charge ratio during the time spent in returning to the GCS T_d, and the operator replacement task T_r.

T_d remains constant in this simplified version of the problem, no matter how far a UAV i is from the region j it was covering, to the GCS.

The main goal of this problem is to minimize the number of active UAVs over time:

$$\min \sum_{i,t} y_i(t). \tag{6.4}$$

This optimization problem with objective function (6.4), and constraints (6.1) and (6.2), maps to the bin-packing problem. Notice that this simplified problem has as bins the drones and as items the areas. Battery levels $C_{B_i}(t)$ are just the bin capacities (Note that having time-dependent variables corresponds to having t repeated in such variables multiple times, i.e., with $t = \{1, 2\}$, $y_i(t)$ is expressed as two different variables $y_{i,1}$ and $y_{i,2}$.), and the battery threshold of each region j becomes the items' weights. Thus, constraint (6.2) is just the bin-packing restriction that prevents exceeding bin capacities. Furthermore, constraint (6.1) imposes that all items (our regions j) are fitted inside a bin.

Without considering (6.2), we already have an instance of the bin-packing problem. Since this makes some instances of our problem being NP-hard, our reduced problem automatically becomes NP-hard. Then, the next step is to generate a heuristic algorithm that will provide a sub-optimal solution. At the same time, it is required to develop a methodology that will enable the algorithm evaluation.

6.3. Methodology

This section describes the different elements depicted in Figure 6.1 and explains the steps to be followed by the mission planner to provide uninterrupted network services. It first describes the parameters that UAVs must report to the GCS in order to serve as input for the scheduler algorithms. Then, it details the diverse assumptions taken for system modeling, that enable simulations to evaluate the preliminary proposals. Later, it presents the metrics to assess the performance of the proposed solutions. Finally, it describes the different strategies used in this chpater to schedule UAVs replacements.

6.3.1. Reported parameters

Current UAV systems regularly report to their control station their location (Global Positioning System (GPS) coordinates if the UAV incorporates this type of navigation) and the remaining battery. However, this knowledge may not be enough to have a holistic view of the UAV network which enables the scheduler algorithm to satisfy the objective function (6.4) of minimizing the number of required UAVs to provide guaranteed service availability. This is the list of parameters periodically reported by the UAVs to the GCS that enable the calculation of essential inputs for the scheduler algorithms that will be defined later to make the appropriate replacement scheduling:

- GPS coordinates: longitude, latitude, and altitude enable the calculation of the distance between each UAV and the GCS. Consequently, taking into account the cruising speed of the UAV, it is possible to estimate the time, and also the required battery, needed to complete the replacement procedure.

- Remaining battery: the current value of the available battery, in combination with the historical battery values (last n values), allows calculating the average energy consumption. With these values, an approximation of the UAVs' lifetime can be determined.

- Network neighbors: the neighboring nodes enable us to generate a graph that represents the UAV network. With the GPS position and the theoretical wireless range, an overview of the network topology can be obtained. However, in certain circumstances, such as several packet collisions or high interferences, having a UAV nearby does not guarantee to have a proper communication channel established. Assuming that communications are bidirectional, for a network link to exist between two UAVs, both have to report each other to the controller, i.e., UAV_A reports UAV_B and UAV_B reports UAV_A. This functionality is deployed in the UAV payload equipment.

- Number of connected users: if a UAV acts as a BS or AP (it can also act as relay, video transmitter, telemetry/sensor transmitter), it must report the number of users that it is serving. This way, we can better determine the impact that a failure (disconnection) of this particular UAV causes in the network.

More details on how to disseminate the data needed to compute the algorithm (reported parameters) can be found in Appendix B.

6.3.2. Assumptions

Using a discrete-time model and in order to provide a reasonable implementation of the UAV replacement strategies, it is required to make some assumptions (simplifications):

1. The UAVs in the fleet are all the same model. It also implies that all batteries have the same dimensions and therefore have the same capacity/duration. This approach is reasonable since handling UAVs that use the same battery model reduces the number of these in the GCS and also simplifies the battery exchange procedure.

2. As long as there is a topological path existing between two nodes, it is assumed that the network route is possible and it is configured, i.e., no time is needed to configure

different routes when topology modifications happen. Unquestionably, the routing protocol used in the network may eventually affect the system but under normal circumstances, the convergence time is negligible [4].

3. The chosen path between two network nodes (UAVs or network users) is the shortest path based on the number of hops. A priori, this decision makes sense since taking the shortest path minimizes the delay (and actually reducing the delay is one of the main objectives of 5G). However, if the network has several users distributed heterogeneously, using different paths may be interesting to balance the network load and avoid packet collisions.

4. When a UAV is in flight, any non-flight related energy consumption (for instance due to wireless transmissions) is negligible [172]. Furthermore, UAVs usually incorporate two batteries [7]. The primary battery is in charge of supplying the flying engines while the secondary supplies the payload equipment. The secondary battery enables a static mode of operation and, in particular situations, UAVs may land to extend the life of the provided network service, stopping the flying engines while keeping the payload powered with its own battery (as introduced in chaper 2). Our laboratory experiments with the secondary battery (3.7 V and 3800 mAh) result in more than 2 h of duration, so the battery limiting the UAV operation is the primary one in any case. Moreover, the battery consumption model is linear and the same in all the UAVs (it does not depend on the flight conditions).

5. Because the price of batteries is remarkably lower than the cost of UAVs, it is assumed that the number of available batteries cells is huge. This way, the GCS never runs out of charged batteries. Batteries can also be recharged during the mission, and some of them could be reused.

6. UAV payloads have enough computing capacity, consequently not saturated under any conditions. In our previous work [7], we have carried out experiments using Raspberry Pi (RPi) 3B single board computers to prove their correct functioning.

Although all these assumptions may affect the results of the simulations, the primary purpose of this chapter is not to achieve accurate results but to verify that it is worth using a replacement scheduler algorithm to manage moderately large UAV fleets. Once this hypothesis is demonstrated, the progressive replacement of each simplification opens interesting future work for the evaluation of more realistic results.

6.3.3. Performance metric

The average number of users connected (over time) is used as the metric to evaluate the performance of UAV replacement strategies. Each sampling period (e.g., 5 s in our simulations), this metric is examined in order to calculate the percentage of end-users connected to the GCS, which is in charge of providing Internet connectivity (i.e., the number of end-users that through a path established across the UAVs network are actually connected to the GCS). The average value of all the partial results during the simulation time will be used as performance metric.

6.3.4. Scheduler algorithm proposals

The following subsection outlines the strategies that have been taken into account when performing the simulations. Obtaining the optimal solution and defining a heuristic algorithm is part of the optimization process. The optimal solution will not predictably serve for large scenarios (in a reasonable time), but it will validate the heuristic algorithm in small scenarios for its future application in real environments. A summary of the parameters that describe the proposed UAV replacement strategies is shown in Table 6.1.

6.3.4.1. Optimal algorithm

To find the optimal UAV scheduling strategy that minimizes the number of UAVs used to cover a certain analysis region and a given number of users, the a brute-force algorithm has been proposed. This algorithm is an evolution of the strategy developed by us and presented in [5], which has incorporated positioning information, a number of users, and specific parameters related to the displacement between the GCS and the regions to be covered (i.e., landing time, take off time, and cruising speed of UAVs). In this regard, the proposed algorithm can be seen as an evolution of the approach addressed in [5]. The UAV scheduling strategy is explained in Algorithm 1, and its operation can be summarized in the following three stages.

- Computation of parameters for UAV allocation. This procedure consists of calculating the lifetime of each UAV (i.e., the battery lifetime to exclusively provide the service in the designated region) and its corresponding replacement time, considering the information about the locations of the GCS and the service regions (GPS coordinates) as well as the parameters v, T_o and T_l. In this step, a priority level or ranking is also assigned to each region according to the number of users that can be affected (disconnected) directly or indirectly if the UAV allocated to that region, and acting as an AP, suffers a failure. Thus, a higher priority level corresponds to a region that, if is not covered (no UAV allocated), it produces a higher number of disconnected users directly or indirectly (AP in that location with a link or links to other locations). This information is used in the process of allocation of UAVs to each region (next step), and ensures that fewer users are affected if $i < j$ at a given time.

Algorithm 1: Optimal UAV battery replacement strategy.

```
Input: GPS GCS and regions, connected users, $j$, $i$, $C_B$, $d$, $L$, $v$, $T$, $T_l$, $T_w$, $T_s$
Output: UAV ID, replacement time
Compute parameters for UAV allocation: $T_B$, $T_r$
Compute Region ranking  // From first to last, based on the number of possible disconnected users
while $Analysis\ time < T_w$ do
    if Topological change then
        Compute Region ranking
                                            // Combinatorial analysis
    foreach $UAV \in i$ do
        foreach $Region \in j$ do
            Create $pair(UAV, Region)$
    $SetPairsUAVRegion$
    foreach $pairUAVRegion \in SetPairsUAVRegion$ do
        Combinations of $j$ elements between $pairUAVRegion$ with other pairs $\in SetPairsUAVRegion$
        Compute service availability
    $SetCombinationsUAVRegion$
    Sorting based on a decreasing service availability level
    Selection of the best combination: $UAVpreallocated$  // Analysis of battery charge levels
    foreach $UAV \in UAVpreallocated$ do
        foreach $Battery\ level \in \left\lceil \frac{T_B}{T_s} \right\rceil$ do
            Create $infoUAVBattLevel$  // Vector with battery levels for each UAV
            Compute service availability
            Computer number of UAVs for next allocation process
    $SetInfoUAVBattLevel$
    Sorting based on a decreasing service availability and number of UAVs in reserve
    Selection of the best battery levels for each UAV
    Perform UAV allocation to designated regions  // UAV allocation process
    Update available UAVs in reserve for replacements  // Unused UAVs and UAVs to which the battery has been replaced
```

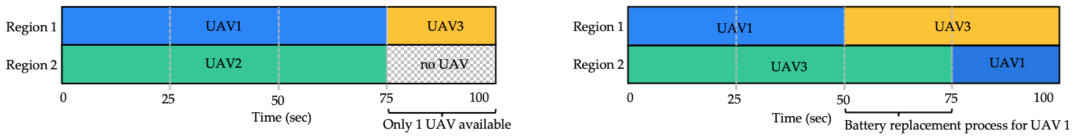
- Optimal distribution of UAVs to cover the service regions. Through a brute-force analysis, all possible combinations of available UAVs to cover the different service regions are explored. The best distribution (combination) of UAVs, which consists of those whose characteristics (battery duration) allow for the highest service availability time, is systematically selected. If necessary, a detailed example of the combinatorial analysis of UAVs to cover services is described in Table 4 in [5].

- Analysis of the percentage of battery charge to perform the replacement. With the information from the previous step (UAV allocation per service region), the algorithm analyses the optimal charge level for each UAV in which the corresponding replacement must be performed. This procedure is carried out by means of an exhaustive exploration of each level of charge for every UAV, and seeks to guarantee the highest service availability time and efficient use of the available resources (minimization of the number of UAVs for replacements). In a traditional approach, as shown in Figure 6.4a, replacement is performed when the battery capacity reaches its minimum threshold ($T_B = 75$ s in the example). Although this procedure allows the full capacity of the battery to be used, the simultaneous discharge of several or all UAVs may cause a greater demand of resources (UAVs) for the subsequent allocations (in the worst case $i = j$) and unavailability of one or several regions if there are no UAVs available for replacements. On the contrary, a desynchronization in the replacement time, as shown in Figure 6.4b, allows not only a greater availability of services but also minimization of the number of UAVs in the system. In the example presented in Figure 6.4a, four UAVs are required (two UAVs in services and two in the reserve) to guarantee a service availability equal to 100%, whereas in Figure 6.4b, only three UAVs are necessary to reach the same availability level. Once the algorithm has determined the charge levels for a replacement that allow the maximum service availability and the maximum number of UAVs available for the next allocation, these UAVs are allocated to their corresponding regions. The allocation process continues iteratively (i.e., execution of step two and step three) until reaching the maximum time horizon T_w, as shown in the example in Figure 6.4b with $T_w = 100$ s.

The proposed optimal strategy is an offline exhaustive search mechanism whose complexity is given by (6.5)

$$f(i, j, T_B, T_s) = C(i \times j\,, j) + \left(\left\lceil \frac{T_B}{T_s} \right\rceil\right)^j \tag{6.5}$$

where the first term represents the combinatorial analysis for the allocation of UAVs and the second term corresponds to the analysis of the charge levels for replacements. Both terms in (6.5) are non-polynomial, the first term is the dominant and, according to the



(a) UAV replacement considering all battery consumption

(b) UAV replacement to offer the maximum service availability and the highest number of UAVs for the next allocation

Figure 6.4: Differences between the analysed scheduling procedures. Example for $j = 2$ and $i = 3$ (2 UAVs in services and 1 UAV for replacement).

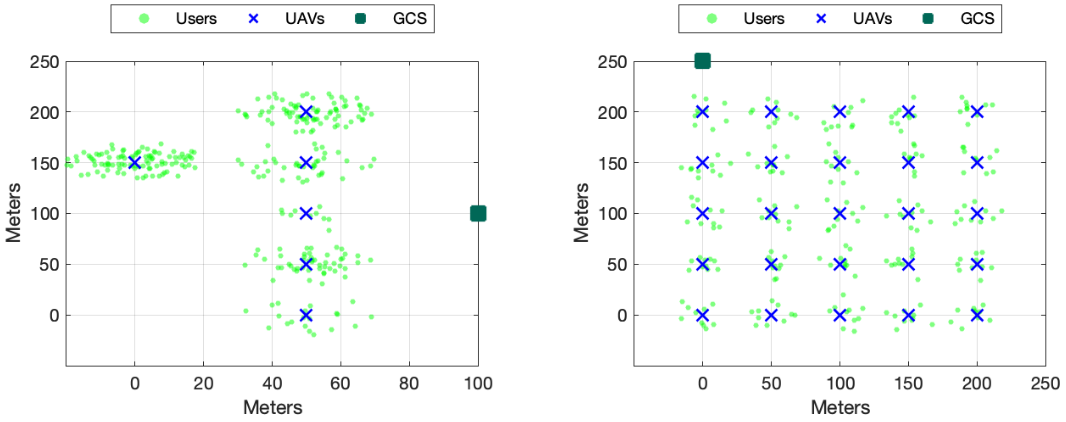
Big-O classification [173], the order of growth of the algorithm is $O(C(ixj, j))$, i.e., non-polynomial. Based on preliminary tests we can report that if the UAVs have the same characteristics (i.e., equal battery capacity) the analysis in step three (exploration of the charge levels) is only necessary for the first allocation process, because desynchronization is maintained all other allocations, as shown in Figure 6.5. While this mechanism can partially reduce the complexity of the algorithm, obtaining an optimal solution using exhaustive search limits it to real-time applications and reveals the drawbacks in selecting the number of regions and UAVs (at most $i = 12$ and $j = 6$). In this regard, this strategy can be used in planning stages to estimate the number of UAVs needed for a mission, such as an emergency or rescue scenario. However, in these cases a suitable alternative is a strategy described in [5], because it is a more generic and less complex approach.

Therefore, the hardness of the problem analysed in Section 6.2.1 (NP-Hard) and the complexity of the optimal solution shown in (6.5) (exponential) demonstrate the need for less complex heuristic mechanisms that can be used in real-time implementations. These strategies are described in the following sections and represent the major contributions of this chapter.

6.3.4.2. Optimal algorithm validation

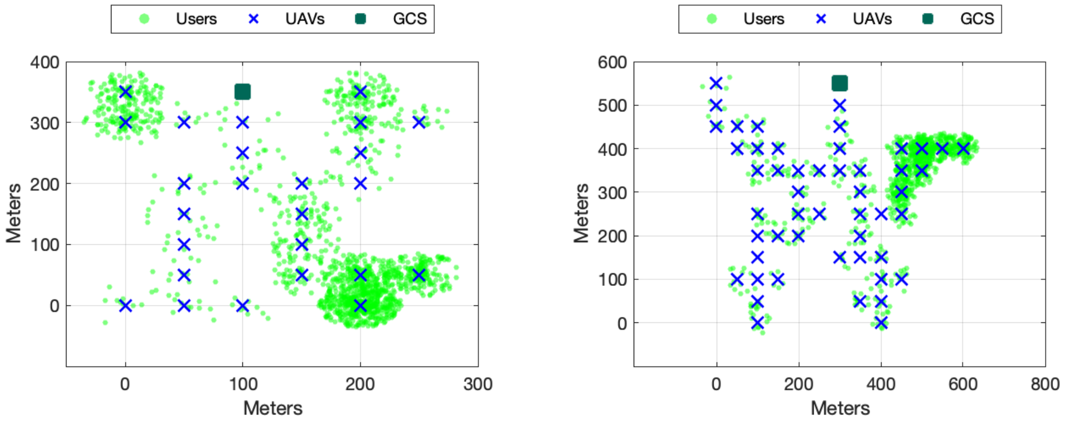
In order to validate the optimal algorithm (section 6.3.4.1), it is necessary to provide as input a rough estimation of the power demanded by each UAV. However, in a realistic environment, in addition to the user traffic (video, telemetry, etc.), there is unpredictable network traffic which needs to be taken into account to measure energy consumption, such as packets re-transmission, WiFi management packets, routing messages, etc.

The power consumption was directly measured using a real RPi and a specific power meter. In order to do so, it is required that the RPi resembles the real conditions as stated in the scenario definition in terms of traffic, CPU load and consider the necessary hardware to enable wireless communication since the consumption depends heavily on these parameters. To calculate all these values, a simulation using ns-3 network simulator has been performed (recreating a real scenario). With this information, we have used the



(a) Scenario II: UAVs as APs

(b) Scenario II: UAVs as BSs



(c) Scenario III: UAVs as APs

(d) Scenario III: UAVs as BSs

Figure 6.5: Proposed scenarios for algorithm performance evaluation.

Iperf tool to generate the UDP traffic flows representing the traffic coming from the ns-3 simulation.

Power consumption was measured with the Monsoon FTA22D Power Meter[3] which provides a robust power measurement solution for mobile devices with high accuracy (±1 %). The selected equipment for UAV payload is the RPi 3B Single Board Computer. The RPi is powered by the Monsoon (Vout voltage of 4.2 V) main channel.

More details on simulation and energy consumption values can be found in our work [5]. The main findings of this analysis are that power consumption does not vary with similar traffic, but it varies by the use of external/additional hardware.

6.3.4.3. BETA: Betweenness Centrality Heuristic Algorithm

Heuristic algorithms are employed to solve optimization problems that are out of scope in reasonable times by optimal algorithms. In this particular case, it is also essential that this heuristic algorithm has a fast execution time because it must be run in real-time. Betweenness Centrality Heuristic Algorithm (BETA) schedules the replacements based on the relevance of each participant within the network. To determine the relevance of an area in a network scenario, we apply graph theory fundamentals. Each area/UAV (an area is covered by a UAV) would correspond to the graph vertices (also called nodes), while the links among UAVs correspond to the graph edges (also called links or lines). One of the most well-known metrics to identify which are the most significant vertices in a graph is centrality, more specifically the betweenness centrality, which resembles the number of times a vertice acts as a connection along the shortest path between two other nodes. However, in the proposed multi UAV networks, nodes do not communicate with other nodes randomly since they do it with those that have Internet connectivity to the public network (either the GCS or a GCS-enabled UAV), as this provides the ground users with Internet connectivity.

To formulate this custom metric, we have divided the graph into two sub-graphs: *(i)* sub-graph which is composed by those UAVs that do not have Internet connectivity and *(ii)* sub-graph which is formed not only by the GCS, but also by the UAVs that may eventually have connectivity to the core network. Therefore, due to these specifications, the centrality metric has been calculated in the following way:

$$g'(v) = \sum_{\substack{s \neq v \\ s \in A \\ t \in B}} \left(\frac{\sigma_{st}(v)}{\sigma_{st}} U_s\right), v \in A \tag{6.6}$$

where $\sigma_{st}(v)$ is the number of shortest paths from UAV s to UAV t that traverse UAV v, and σ_{st} the total number of shortest paths from UAV s to UAV t. U_s is the number

---

[3] Monsoon FTA22D Power Meter, accessed Tuesday 15th November, 2022. Available: https://www.msoon.com/

of users connected to UAV U_s. The amount of users is crucial since if there are no users connected to UAV U_s, there is no impact on the network. This statement (6.6) (which is quite versatile) despite being designed for FANETs is also suitable for BS scenarios (where UAVs are directly connected to the core network).

A ranking was computed using the $g'(v)$ metric as input. In case two UAVs have the same value $g'(v)$ the one closer to the GCS will be above the other in the ranking since this will minimize the total replacement procedure time, and the replaced UAV will be active sooner to perform another UAV replacement. Now it is possible to assume which scheduling strategy to follow. BETA attends the following strategy (it can be appreciated in Algorithm 2): *(i)* if there is any topological change or the ground users move around the scenario, the algorithm must compute the ranking again; *(ii)* whenever there is a UAV available in the reserve, the algorithm schedules a replacement to the UAV with less remaining battery. However, this replacement takes place only if it does not affect UAVs that have a higher position in the ranking, i.e., that means that the remaining lifetime of the top UAVs is shorter than the time needed to make a UAV available again (after flying towards the GCS and battery replacement). In the case that this UAV replacement cannot be performed, the same analysis is repeated for the next UAV with the lower battery until the algorithm finds a UAV to make the replacement. For this algorithm to work correctly, it has to be executed periodically. In our case BETA runs every 5 seconds which coincides with the sampling period.

Algorithm 2: UAV replacement methodology.

```
Input: GPS, remaining battery, neighbors, connected users
Output: UAV ID, replacement time
Update UAV lifetime              // based on remaining batteries and GPS
Update battery ranking           // based on remaining batteries
if Topological change then
    Compute UAV ranking          // based on neighbors and connected users
for UAVs battery ranking do
                                 // From lowest to higher
    if UAVs in reserve > 0 then
        time = time to replaced UAV to be available again
        warnings = find (time > lifetime)
        if warnings < reserve UAVs then
            Schedule replacement
```

6.4. Simulation details and results

In order to validate these algorithms we have used different scenarios with different properties that will be discussed in this section. The following subsections detail *(i)* the simulation parameters and the justification of their selection, *(ii)* the ground-truth solutions with which the BETA algorithm is also compared, *(iii)* the simulation setup, and finally *(iv)* the simulated scenarios in combination with achieved results.

6.4.1. Simulation parameters

This subsection details the parameters that have been taken into account to carry out the simulations and the selection criteria. This data, together with the related notation, can be seen in Tables 6.1, 6.2, and 6.3.

Table 6.1: System parameters.

Parameter	Notation	Units/Coments
Number of regions	j	Integer number
Number of UAVs	i	Integer number
Location GCS	P_{GCS}	x,y coordinates
Location UAV$_i$	P_{UAV_i}	x,y coordinates
Number of users per region	u_j	Integer number
Total number of users	U	Integer number
Battery replacement time	T_r	Time units, e.g., seconds
Battery capacity	C_B	Electric current per time units, e.g., mAh
Device consumption	d	Electric current, e.g., mA
Link distance	L	Length units, e.g., meters
UAV cruising speed	v	Speed units, e.g., meters/seconds
Take-off time	T_o	Time units, e.g., seconds
Landing time	T_l	Time units, e.g., seconds
Simulation time	T_w	Time units, e.g., seconds
Sampling time	T_s	Time units, e.g., seconds

The time needed to perform a battery replacement is based on [152]. The battery capacity is based on the Parrot Bebop 2 specifications (It is chosen because we have

Table 6.2: Simulation parameters A.

Scenario	**Parameters**		
	$\boldsymbol{i}$	$\boldsymbol{j}$	$\boldsymbol{u}$
I	6	6-12	300
II	25	25-50	250
III	25	25-50	300
IV	50	25-50	500

Table 6.3: Simulation parameters B.

Parameters	**Value**
$\boldsymbol{T_r}$	180 s
$\boldsymbol{C_B}$	2700 mAh
$\boldsymbol{d}$	5670 mA
$\boldsymbol{L}$	70 m
$\boldsymbol{v}$	5 m/s
$\boldsymbol{T_o}$	60 s
$\boldsymbol{T_l}$	60 s
$\boldsymbol{T_w}$	3600 s
$\boldsymbol{T_s}$	5 s

performed several tests using this model, and it is the selected unit in the technical validations we have worked previously[4] [5] [7], since it has demonstrated that it is able to carry a single board computer onboard like a RPi for a reasonable time without problems and a reasonable cost). To calculate the device consumption, we have assumed that the UAV flies for 20 min (also specified in the technical characteristics). For WiFi range and although the standards state that the range is quite large, in practice, we have found that the WiFi range is relatively short for an acceptable received signal level [14]. The cruising speed has been calculated based on its maximum speed (also on technical specifications). Meanwhile, takeoff and landing times have also been calculated by our own measurements

---

[4] Telemadrid - La Universidad Carlos III junto al Instituto Imdea Networks desarrolla un dron antincendios: shorturl.at/bpGJ7

[5] TVE - Zoom Net - 5GRange: shorturl.at/cjozW

since we have not found accurate information. The simulations are iterated assuming a fixed number of areas to be served (and obviously one UAV per area) and then increasing the number of replacement UAVs (starting by 0 and increasing until the number of UAVs in reserve equals the number of UAVs in the scenario, which would mean doubling the size of the fleet).

6.4.2. Ground-Truth solutions

To provide context to the BETA and optimal algorithms performance, they will both be compared with two alternative solutions (with smaller complexity). The primary purpose of the chapter is not to measure how far the heuristic solution is from the optimal but to highlight that the use of this type of solution is worthwhile and under which conditions and in which scenarios. In order to do that, the four scheduling techniques will be compared (BETA, optimal, baseline and simple scheduling) and different conclusions will be obtained

6.4.2.1. Baseline

This is the simplest strategy. UAVs are assumed to periodically send their current battery level and GPS position, however, no further calculations are made from the GCS. When an active UAV reaches a minimum battery threshold, i.e., only the required battery to return to the GCS plus a safety threshold, e.g., 20%, a replacement is scheduled (if UAVs are available), i.e., the drained UAV flights to the GCS, and at that moment (when the drained UAV starts flying to the GCS), a fresh UAV takes off and flies to the uncovered target area to provide the service. If no fresh UAVs are available, there will be no service in that area until a UAV is ready to go and cover it again. The lack of intelligence in this baseline solution prevents us from reaching 100% service provisioning in any case because even with infinite UAVs to serve as fresh replacements there will always be a gap without network service corresponding to the time that passes since the drained UAV leaves the stage towards the GCS until the moment the new UAV enters the stage and starts operating.

6.4.2.2. Simple scheduling

This strategy is inspired by UAVs are also assumed to send their current battery level and GPS periodically. However, in this case, the controller is required to estimate a battery threshold (based on battery reports and other parameters such as the UAV speed and the takeoff and landing times) that includes not only the minimum battery needed to return to GCS but also includes the time needed for the fresh UAV (in case there are available units) to reach the target area. That way the new UAV will start serving the area just after the old one leaves and predictably, if there are enough UAVs in reserve, all the areas can be covered for the whole mission time (or at least a high percentage of the time). In case the active UAV reaches the threshold and there is no fresh UAV to perform the replacement, the UAV can still continue providing service until it reaches the battery level needed to reach the GCS enlarging the service time.

6.4.3. Simulation setup

A *Matlab (Matlab R2017b)* event-based simulator achieves all the chapter results. To calculate the BETA, simple scheduling, and baseline solutions in all the scenarios, a computer equipped with a 2.6 GHz Intel Core i5 processor and 8 GB Random Access Memory (RAM) was used. Meanwhile, the optimal algorithm has been run on a computer equipped with a 3.33 GHz x 12 cores Intel Core i7 Extreme processor and 24 GB RAM. If the reader is interested in reproducing the experiment, the code is available in this repository [174].

6.4.4. Validation scenarios

6.4.4.1. Scenario I: Proof of Concept

To start the analysis, we have defined a basic scenario (it can be appreciated in Figure 6.5a) as a proof of concept. In this stage, there are a total of 6 coverage areas and 300 ground users heterogeneously distributed. In a scene with these reduced dimensions, it is possible (in terms of reasonable computation time) to run the algorithm that provides the optimal solution so it will be possible to compare all the alternatives. Figure 6.6a depicts the average connected users for the four algorithms when the UAVs act as a FANET which means that UAVs onboard commodity WiFi equipment and use the created UAV WiFi ad hoc network itself to connect to the GCS (which in turn provides the Internet connectivity). Therefore if one of the UAVs that are geographically closer to the GCS (and hence connecting part of the topology to the GCS) runs out of battery and there is no possible UAV replacement, some parts of the network may get disconnected even though the rest of UAVs may be successfully covering other target areas. Following this logic, whenever there is a failure in the backbone UAV, the system gets completely divided. On the other hand, Figure 6.6b depicts the average connected users for the four algorithms when UAVs act as BSs (they are directly connected to the public network without the need of a hop by hop network like a FANET). These scenarios are usually employed in massified events where the existing cellular network is operating correctly, but may be insufficient. As expected, these results are better than the FANETs results since each UAV is only responsible for its own end users. However these on-boarded BS solutions are usually more expensive and it is not always viable (when the infrastructure does not exist or is temporarily damaged for instance).

Figure 6.6a shows that both BETA and the simple scheduling strategies perform similarly and are close to the optimal solution. To reach 100% of connected users with the simple scheduling approach, it is required to double the UAV fleet (12 UAVs) but in any case in reduced scenarios, the simple scheduling solution is enough to provide an adequate service. On the other hand, the baseline algorithm provides erratic and unintuitive results considering that the performance decreases as the fleet increases. This phenomenon happens because although the time that UAVs are covering the target areas is higher, the network is disconnected for longer, i.e., having more UAVs does not guarantee overall connectivity if the backbone UAV is not working. If there are no reserve UAVs in reserve (the fleet size is equal to the number of target areas), the return and battery replacement process (of all the UAVs in the scenario) is almost synchronized (and operate simultaneously). However, if there are some UAVs in reserve, this process may be unsynchronized. For this reason, the baseline results decrease and, in consequence, are worse and inconstant.

The results in Figure 6.6b (UAVs acting as BSs) are better as we commented and again BETA and simple scheduling strategies are close to the optimal solution. The baseline solution performance improves in this case, as the size of the fleet increases. All the strategies in fact stabilize with a fleet of 8 UAVs, two in reserve (fleet 25% oversize), and both BETA and simple scheduling achieve acceptable values.

In scenarios with reduced dimensions this 25% of fleet oversize (having two UAVs in reserve) seems quite reasonable. However, in a scenario with numerous areas, e.g., 25 areas, 50 areas, this oversize may imply a rather expensive operation. It is then important to validate the solutions in much bigger scenarios and see the performance of the algorithms there.

6.4.4.2. Scenario II: Grid

Figure 6.5b shows a scenario with 25 coverage areas and 250 ground users homogeneously distributed, i.e., ten ground users per area. We have selected a grid topology which is fail-tolerant since there are multiple alternative paths to reach the GCS from each area. Moreover, all the areas have the same number of users, which makes the difference in the UAV ranking insignificant in the FANET scenario and almost nonexistent in the BSs scenario.

Figure 6.6c shows that the performance of the heuristic strategy is better than the simple scheduling solution (when the fleet is formed by 30 UAVs, 5 UAVs in reserve, the results improve by more than 10%). Both strategies reach acceptable levels from 35 UAVs fleet. The heuristic solution reaches 100% of users connected with a 38 UAVs fleet while the simple scheduling solution, as in the first scenario, needs to double the fleet size to reach 100% of connected users. On the other hand, the baseline solution has similar behavior to the previous scenario. This outcome highlights that if no strategy (however simple) is used to schedule the UAV replacements, the results can be harmful, and even over-dimensioning the resources does not guarantee favorable results.

Figure 6.6d presents the results of UAVs acting as BSs. In this case, the heuristic algorithm behaves similarly to the simple scheduling solution. The heuristic algorithm schedules the UAV replacements based on $g'(v)$ metric (6.6), which is determined using graph theory. In this scenario, the nodes representing the UAV network have the same $g'(v)$ since they are all directly connected to the infrastructure and provide connectivity to the same ground users; therefore, all UAVs connect the same number of users to the network. For this reason, scheduling the UAV replacements using the heuristic strategy has no advantage other than that they are performed as soon as there is an available fresh UAV. This phenomenon reveals that the heuristic solution makes the difference in scenarios where UAVs have different relevance within the network.

6.4.4.3. Scenario III and scenario IV: Tree

Finally, we have designed two tree type scenarios with the users distributed very heterogeneously. This type of scheme makes some UAVs much more relevant, and scheduling replacements effectively seems to have a substantial impact on the final performance. The first scenario has 25 coverage areas and 300 ground users. The second scenario has 50 coverage areas, 500 ground users. The areas and user distribution can be appreciated in Figure 6.5c,d.

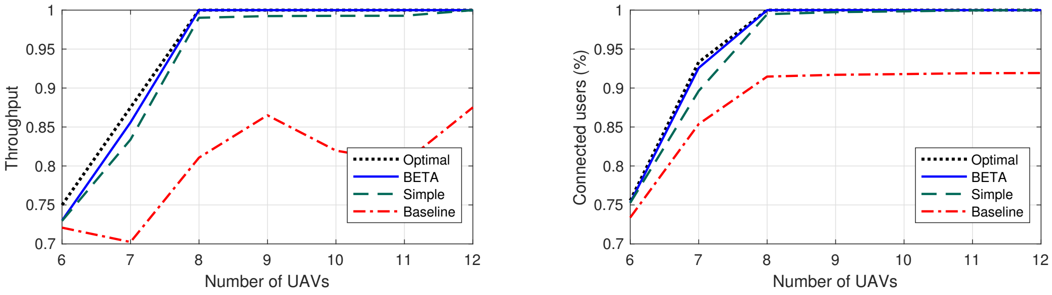
Figure 6.6e reveals that the difference between the BETA solution and the simple scheduling solution is significant in these scenarios. For a 30 UAVs fleet (5 UAVs in reserve), we achieve a 20% improvement by using the heuristic scheduler, which is an important variation when providing a network service. The heuristic strategy obtains 100% of users connected from 36 UAVs. As in previous scenarios, the baseline solution produces insufficient results. It is interesting to observe that as the UAV fleet increases (which have high economic cost), the users are not connected longer.

Similarly, when UAVs act as BSs, Figure 6.6f, we obtain better results using the heuristic strategy. This variation is because, in this scenario, the ground users are heterogeneously distributed, and consequently, UAVs have different $g'(v)$ since they connect diverse numbers of ground users, which implies that performing the correct replacement has more impact.

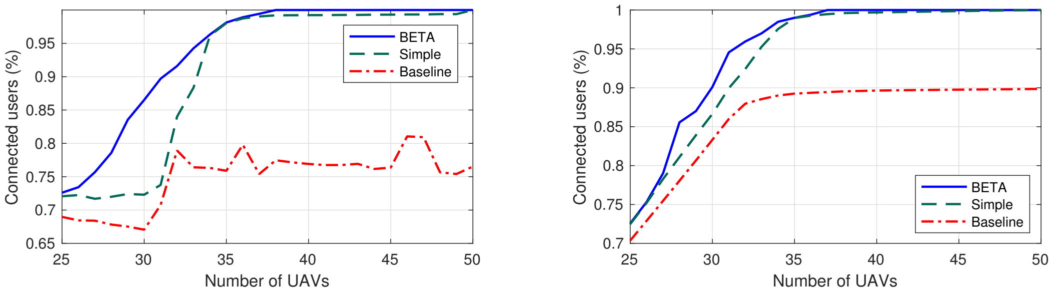
The conclusions of scenario IV are similar to the ones of scenario III, although the performance (Figure 6.6g,h) is worse because of the greater complexity of the UAV network topology and the greater failure possibility.

6.4.5. Comparison of the UAV replacement strategies

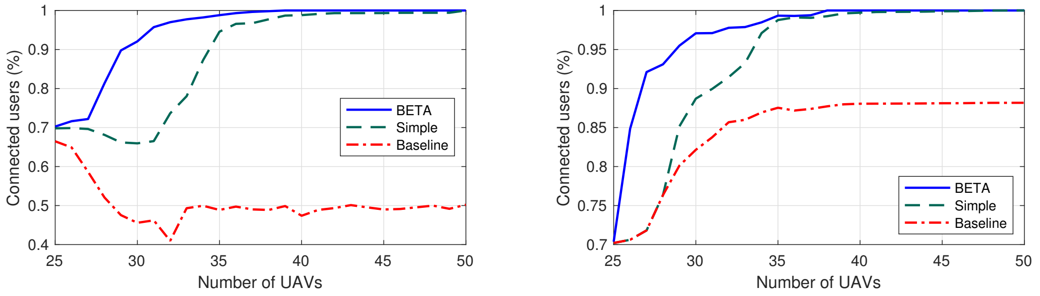
Once the results have been obtained (Figure 6.6) and discussed (Section 6.4.4), in this section we will perform a comparison of the results by computing: *(i)* for scenario I,



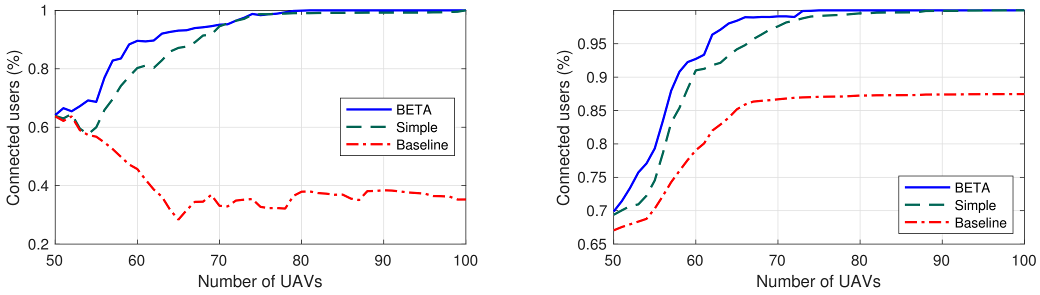
(a) Scenario I: UAVs as APs (b) Scenario I: UAVs as BSs



(c) Scenario II: UAVs as APs (d) Scenario II: UAVs as BSs



(e) Scenario III: UAVs as APs (f) Scenario III: UAVs as BSs



(g) Scenario IV: UAVs as APs (h) Scenario IV: UAVs as BSs

Figure 6.6: Average throughput in different scenarios increasing the fleet size.

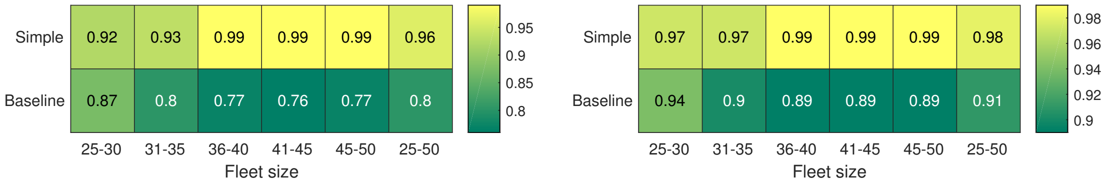
the distance from the optimal solution (*Opt*) to the suboptimal or approximate solutions (*SubOpt*), in order to verify the quality of the results, and *(ii)* for the succeeding scenarios, the difference between the simple scheduling and baseline approaches against the Heuristic strategy. To this end, the criterion of approximation ratio (ρ) has been used [159]. This parameter, in the context of the proposal, estimates how many times lower the approximate solution is as compared to the exact or optimal result; it is defined as the average value of the ratio between the suboptimal and optimal solutions, as shown in

$$\rho = \frac{1}{i} \sum_{i} \frac{SubOpt_i}{Opt_i}, \tag{6.7}$$

where *SubOpti* and *Opti* are the results for all the variation of UAVs in reserve (from zero to fleet size) for the optimal and suboptimal strategies, respectively. The ρ factor ranges from 0 (whether *Opt* and *SubOpt* have completely different values) to 1 (if *Opt* and *SubOpt* produce the same solution); an intermediate value (i.e., $0 < \rho < 1$) represents the similarity or closeness factor to the optimal solution ($SubOpt = \rho x Opt$) [159]. For a better understanding of the calculation of this parameter, (6.8) presents an example for the simple scheduling approach of Scenario I when UAVs act as APs (Figure 6.6a).

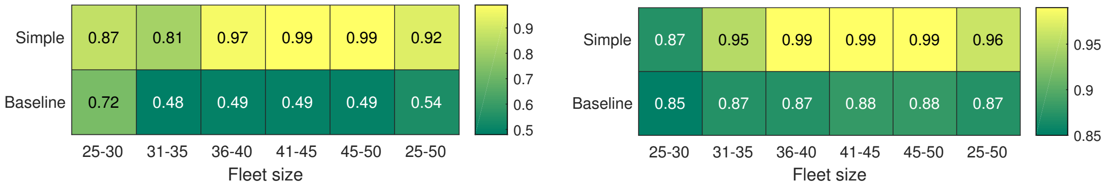
$$\rho = \frac{1}{7} \times \left(\frac{72.95}{74.99} + \frac{83.38}{87.51} + \frac{99.02}{100} + \frac{99.24}{100} + \frac{99.27}{100} + \frac{99.28}{100} + \frac{100}{100} \right) = 0.98. \tag{6.8}$$

The result of (6.8) shows that the suboptimal solution (simple scheduling approach) is similar to the optimal solution in a factor equal to 0.98 (98% similarity between solutions). The rest of the ρ factors for Scenario I (Figure 6.6a,b) are summarized in Figure 6.7, while the ρ values for other scenarios are presented in Figure 6.8. The comparison between the optimal solution and the approximate solutions in Scenario I, based on ρ factor, reveals that all the proposed strategies produce not only near-optimal solutions, but also a stable performance (i.e., high-quality feasible solutions). In all cases, as illustrated in Figure 6.6a,b and then corroborated in Figure 6.7, the approximate algorithms (BETA, simple scheduling and baseline) generate solutions very close to the optimal, even with 99% similarity to the exact result (1% of error), which is achieved by the BETA approach. Then, this strategy is used as a baseline to evaluate the performance of the other strategies (simple scheduling and baseline) for Scenario II, Scenario III, and Scenario IV. In summary, from the information analyzed in Section 6.4.4, the average number of connected users allows us to appreciate where one algorithm improves another, while the distance to the optimal solution computed in this section allows us to quantify this variation. To provide a higher level of detail, ρ has been analyzed by ranges depending on the fleet size (all the analysis has been carried out by increasing the number of UAVs gradually). In addition, this value has also been computed at a global level. The main reason is to analyze in which areas the solution improves and quantify it. Since from a



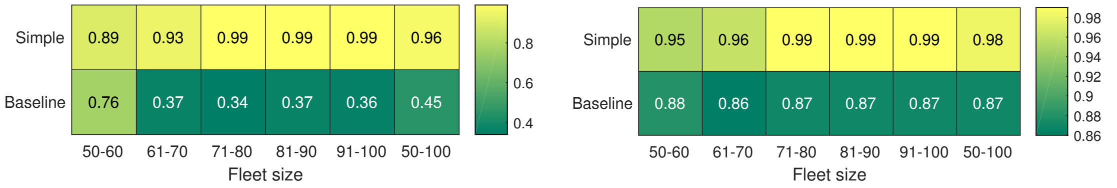
(a) Scenario II: UAVs as APs

(b) Scenario II: UAVs as BSs



(c) Scenario III: UAVs as APs

(d) Scenario III: UAVs as BSs

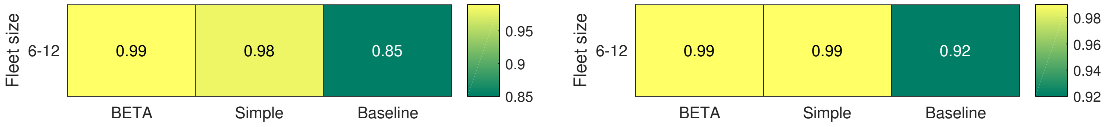


(e) Scenario IV: UAVs as APs

(f) Scenario IV: UAVs as base stations (BSs)

Figure 6.8: Approximation ratio ρ: betweenness centrality heuristic algorithm (BETA) vs. other suboptimal strategies.

particular value, the solutions provide similar results, computing this metric at a global level makes it difficult to recognize the areas of improvement.



(a) Scenario I: UAVs as APs

(b) Scenario I: UAVs as BSs

Figure 6.7: Approximation ratio ρ: optimal strategy vs. heuristic strategies for Scenario I.

It can be seen in Figure 6.8 that the central area of improvement is in the first two ranges, i.e., from 25 to 35 UAVs. This result is positive because, as expected, if there is a reasonable amount of UAVs (with their corresponding cost), a typical solution can perform adequately. However, in scenarios with limited resources using the heuristic strategy improves in all cases the simple scheduling solution.

Analyzing the above metrics, we can conclude that using strategies to make replacements is worthwhile. However, the heuristic strategy designed in this chapter is considerably aggressive since it schedules a replacement whenever UAVs are available. This strategy can result in the number of replacements skyrocketing over time, as well as

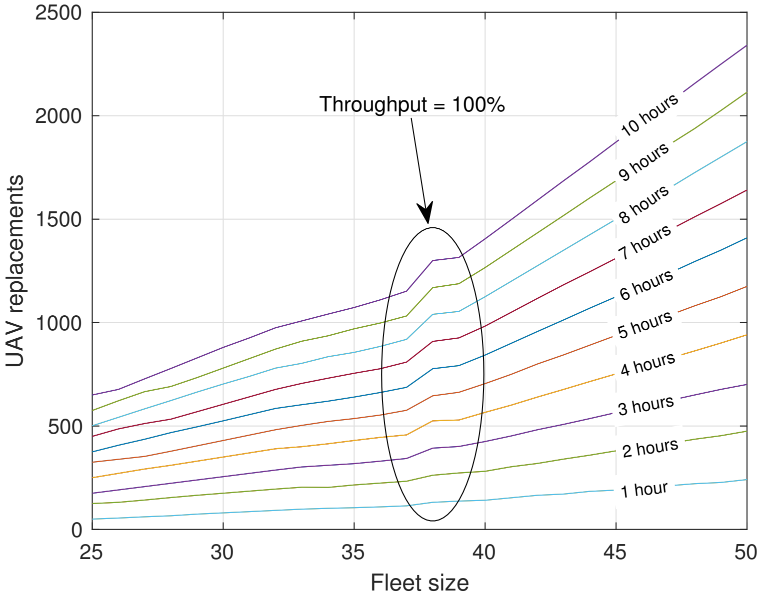


Figure 6.9: Number of UAV replacements using BETA in scenario III.

the number of batteries to be used, which would bring a high economic cost. It should be noted that the price of a battery is much lower than that of a UAV but in any case it is not negligible.

Figure 6.9 displays the number of UAV replacements as a function of the number of UAVs forming the fleet and the mission time for scenario III using the BETA strategy. Approximately 1200 replacements are needed to provide a service of 10 h and obtain 100% of users connected. Furthermore, due to the aggressive nature of BETA, the replacement grows exponentially after reaching a fleet size that guarantees 100% of connected users. In addition, it should be mentioned that the main advantage of small UAVs is that deployments are generally done very quickly and very flexibly, but as we have seen it is both economically and logistically difficult, to achieve reasonable solutions when service time largely exceeds battery lifetime. Other alternatives should be used in these cases like bigger UAVs with more battery capacity (or even using fuel), land the UAVs on the ground to improve their autonomy, or even deploying of a fixed infrastructure if service is expected to be maintained for a long.

6.5. Discussion

This chapter states the practical UAV replacement problem, where a multi-UAV network is expected to provide long-endurance network services (in the order of hours) using constrained devices with limited autonomy (in the order of minutes). It is verified that the optimal UAV scheduling to minimize the number of UAVs for replacements while providing a guaranteed service availability, is NP-hard and that its optimal solution has exponential complexity. In this regard, some heuristics approaches have been analyzed and evaluated.

Secondly, the chapter details a methodology, including the simulation environment and the parametrization required to perform a preliminary evaluation of these heuristic strategies. The simulator code is available in [174] to reproduce the experiment and evaluate upcoming future strategies.

The chapter also introduces BETA, a heuristic replacement strategy that performs the replacements as soon as possible based on the relevance of each UAV within the network. BETA is presented as an example in order to verify if it is worthwhile using a heuristic replacement approach or not. BETA is capable of running in real-time with a 99% similarity with the optimal solution in some simple scenarios (scenario I). In heterogeneous scenarios, BETA improves the basic solutions, achieving the most significant improvement in instances where the scenarios are heterogeneous, and the resources are limited. Furthermore, we conclude that it is far better to have a replacement strategy (no matter how simple it is) than having no strategy at all. BETA is compared with the optimal algorithm in order to evaluate the distance whenever possible and with other alternatives in some other scenarios and it has been possible to see that in some situations the advantages are not so relevant as in other ones.

The chapter opens several lines of future research, such as to be able to provide priority of replacements for UAVs serving users in emergency/disaster scenarios. The application of replacement strategies in disaster scenarios includes an uncertainty degree caused by several factors caused by moving UAVs while they are operating (that may change the topology), or extreme conditions that may force the engines and battery consumption. Some other futures lines include *(i)* the combination of FANETs and BSs in the same scenario, *(ii)* to test UAVs models with different battery capacity, *(iii)* to model the energy consumption according to more realistic consumption patterns based on experimentation, or *(iv)* to schedule UAV replacements without making them through the GCS.

6.6. Research impact/dissemination

The outcome of the work behind this chapter has resulted in two publications. The first publication was in the Hindawi *Wireless Communications and Mobile Computing* journal. This journal has a current impact of 2.336, and it is ranked as Q3. The second publication was in the MDPI *Sensors* journal. The MDPI *Sensors* has a current impact of 3.576 in the Journal Citation Reports (JCR), and it is ranked as Q1. A total of 26 research papers has cited both articles.

This work has been carried out in collaboration with the Polytechnic University of Catalonia under the umbrella of the 5G City and TRUE5G Spanish projects.

7

Multi-interface network framework for UAV management and data communication

The same way we can find rules, normative, and agreements on procedures around the world to manage different aspects of ground or aerial traffic, several efforts during the last years have been addressed to organize and regulate the civil usage of Unmanned Aerial Vehicle (UAV) at a Very Low Level (VLL) airspace (below 500ft above ground level) [175].

One of these efforts recently resulted into U-space [70], the Concept of Operations (CONOPS) to harmonize the UAS Traffic Management (UTM) [176] in Europe by Single European Sky ATM Research (SESAR) [177]. This first high-level concept [100] will be progressively complemented with the ongoing and future definitions of requirements to make UAV flights safe, avoiding the adverse side effects these could have on citizens and the environment.

To test the U-space design and identify gaps and needs, the European Union, with its Horizon 2020 research and innovation program, has endorsed research projects that implement different aspects of the U-space system. One of those projects is Labyrinth, in which a U-space UTM system is being built to investigate and experiment with aspects such as route optimization, timely monitoring and separation of the traffic, security, or capacity of the communications [178]. This last area focuses on the performance of the different communications alternatives used in a UTM system, such as WiFi, Radio, 4G/LTE, Fifth Generation of cellular network technology (5G), or satellite communications (Satcom). Also, these links can coexist and coordinate when more than one is used to provide the UAV with a backup link or to provide the possibility to switch and choose the link with better Quality of Service (QoS).

This chapter presents some of the goals of the work package in Labyrinth dedicated to communications. The following section describes the information exchanges expected in a UTM system and how these will be implemented in the Labyrinth prototype to overview their role in the system. Section 7.2 presents the communication framework. Section

7.3 presents a preliminary experimentation of the proposed communication prototype. Finally, section 7.4 concludes the chapter and depicts some future research lines.

7.1. Labyrinth details and requirements

In conventional aviation, each center where Air Traffic Controller (ATC) perform their assignments is called an Air Traffic Services Unit (ATSU). Similarly, the UTM server will act as an automated ATSU for the UAVs. Like in an ATSU, the UTM server tries to keep a global picture of the traffic in the area (present and future). It will also perform relevant tasks such as: *(i)* assist the operators to design a feasible and optimized trajectory, free of collisions; *(ii)* monitor the traffic and send instructions during the flight to the pilots to guarantee the distance between the UAVs; *(iii)* rearrange the traffic to give way to priority specific flights; *(iv)* apply measures and warn pilots in case there is a problem with a UAV out of control; and *(v)* broadcast information of interest for the pilots.

In manned aviation, ATCs provide that help using tools or sources of information either on-site or from external locations. Also, in our case, what in the U-space jargon are called services, will be sources of information or applications that support the UTM server to perform its tasks. Those services can be inside the UTM server or another external server provided by a different entity or company. In particular, in Labyrinth, the path planner responsible for calculating the optimal path from origin to destination is provided as an external service [178].

U-space defines different types of airspace volumes, based on the risk of the operations. Those volume types determine the requirements so as to be allowed to fly in each of them. In the case of Labyrinth, where we are testing the UTM with different use cases (road transport, air transport, emergency, and waterbone transport), the type of volume for these use cases require the UAVs to constantly report their position to the UTM server. This report must be either done by the Ground Control Station (GCS), forwarding the reported position by the UAV, and by the UAV itself directly sending the position to the UTM server. To complete the reports from the UAV, it is essential to have a cellular or satellite connection. Ideally, the cellular connection will be the main alternative chosen to make these reports; however, the satellite connection provides a backup link that can complement the lack of cellular coverage.

In manned aviation, when ATCs need to change the trajectory of a certain flight, they send a text message with the instruction to the pilots or directly speak it to them using the radio. Therefore, ATCs issue the instruction but it is the pilot, after replying with a WILCO (message confirming that the pilot has received the instruction and will comply with it), who finally introduces the command and executes the instruction. In U-space, UAV pilots will receive the instructions from the UTM server and they must accept and load the command in the UAV, also during the flight. For the instructions, warnings or information, to reach to the UAV pilot, the GCS must be constantly connected to the

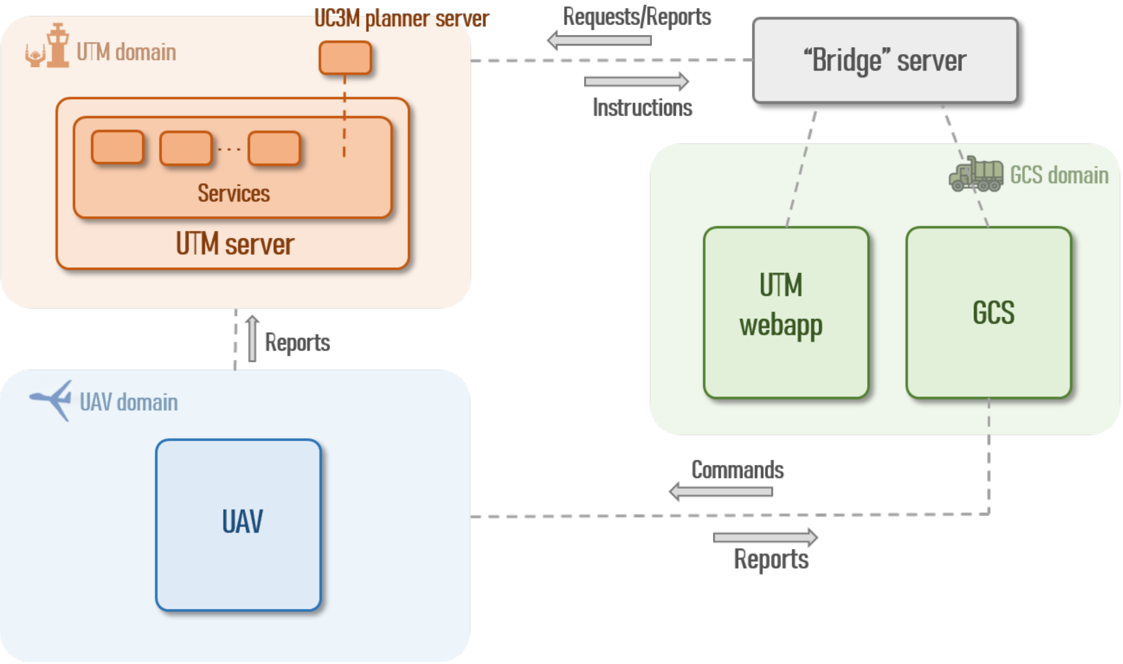


Figure 7.1: Elements and communication links in the Labyrinth U-space UTM system.

UTM server and, of course, the UAV with the GCS, to be able to load and apply the commands.

To communicate with the UTM server we defined an Application Programming Interface (API), a protocol that describes how to send requests or reports to the UTM and the format of the replies that will be received. For the pilots to easily communicate with the UTM, they need some dialogues in a graphic user interface. These can be embedded in the GCS software, using the API to make those calls. But unfortunately, that is not always possible, since there may not be room left in the display for the dialogues, or the code of the GCS may not be modified for security reasons, or it might imply a considerable cost for the flight operator. To ease things in this sense, we are working on a web application to contain those dialogues and properly display the received information and warnings.

As previously said, the UTM sends instructions to the UAV pilots. Some of them can be simple, like a change of altitude or speed, with only one numerical value implied. But others could imply a long list of hundreds of complex values defining a new trajectory and in this case, we cannot expect the pilot to manually introduce them in the GCS. The problem is that, even if we had the web app running in a browser in the same computer as the GCS, for security reasons, is not possible to get from a browser the level of interaction with the operative system that we would need for the web app to interact with the GCS (otherwise, it would be very easy to harm our computers while visiting malicious or infected webs). The solution here has been to use another server as a bridge

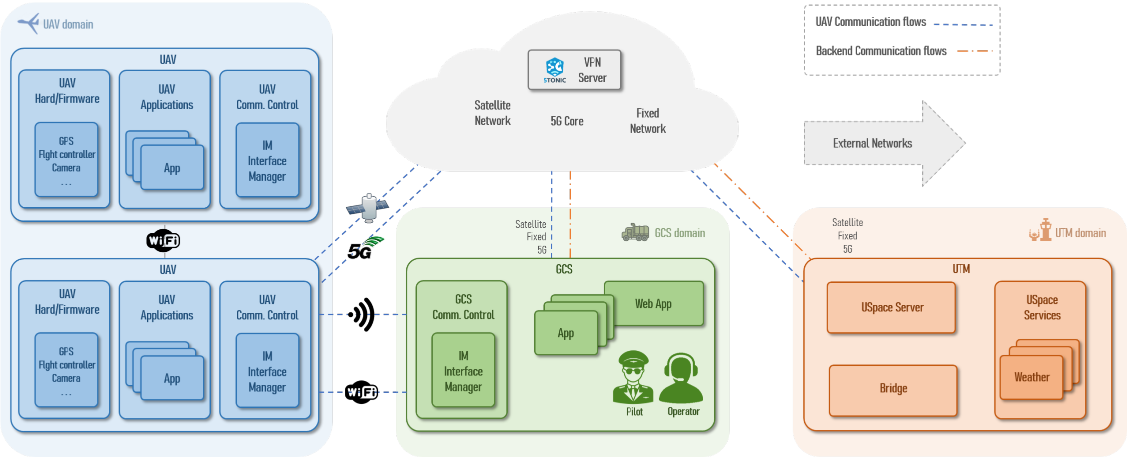


Figure 7.2: Labyrinth communications framework.

to communicate the web app and the GCS. This server will also forward the position reports from the GCS to the UTM, and the requests of the pilot from the web app to the UTM. In the opposite sense, the bridge can also deliver the UTM messages to the web app, and the instructions to the GCS to be converted into commands and loaded in the UAV once accepted by the pilot.

The last type of message exchange, this one without the UTM or the GCS involved, appears in the case of swarms of UAVs. Inside the swarm, the UAVs can share between them their position or intentions to keep themselves separated. They can also use this UAV-to-UAV communication to coordinate their tasks during the mission.

A summary of these messages and the operation of the UTM can be seen in Figure 7.1.

7.2. Labyrinth network framework

As discussed in the previous section, the principal actors in a mission are within three main domains: the UAV domain, the GCS domain, and the UTM domain. The communication between these domains has to be reliable and remain stable during the mission for proper operation.

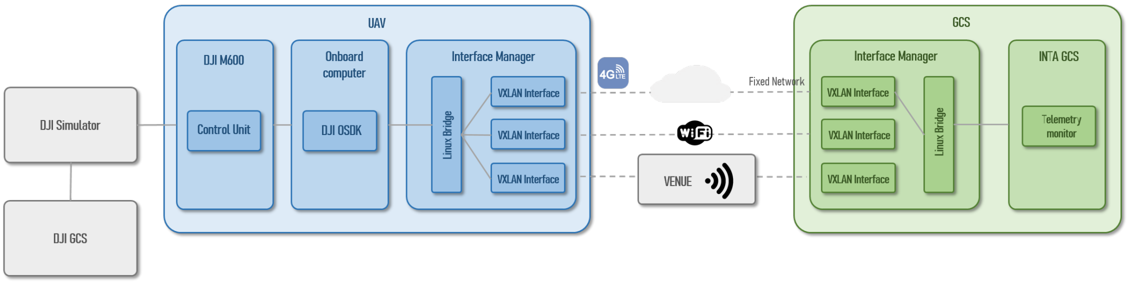
7.2.1. Inter-domain communication alternatives

Communication between UAV and GCS domains can be implemented using different alternatives, from the traditional approach through a direct link using a radio modem to novel paradigms such as the use of the public cellular network, which would allow a GCS-UAV communication beyond the line of sight as long as the two domains have Internet connectivity. In particular, in the Labyrinth project communication architecture (see Figure 7.2), four types of communications are considered: *(i)* a radio modem link;

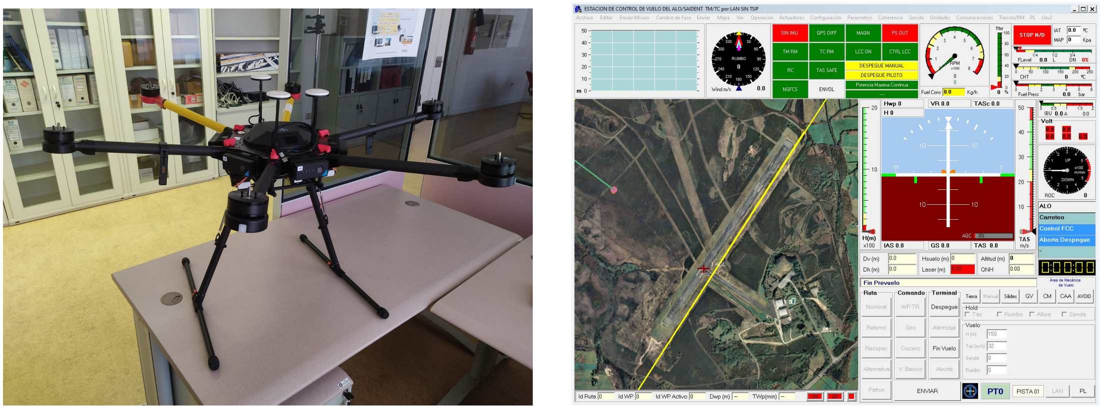
(ii) satellite, which is limited by a small bandwidth and a higher delay (this point to be checked in the project), but with available coverage in open environments; *(iii)* WiFi, not only to communicate UAV and GCS domains but also to enable the previously mentioned UAV-to-UAV communications if they are flying close together. Another representative use due to the small range of WiFi, is when the UAV is on the ground, the UAV operator has to access the equipment onboard the UAV if any configuration or adjust is needed; *(iv)* public cellular network (5G/Long Term Evolution (LTE)/4G), which enables the UAV not only to communicate with the GCS but also to report directly its position to the UTM server. It should be noted that the GCS can be connected to the public network (which enables connectivity with the UAV) using the fixed network or the cellular network interchangeably as long as one of them is available. The radio modem link is the most common in UAVs, especially in residential and recreational environments. In this project, and in particular, in this chapter, the focus is on cellular communication for several reasons: *(i)* It enables direct reporting of specific parameters directly from the UAV to the UTM; *(ii)* It does not limit the coverage range to connect UAV and GCS as both domains will be connected as long as both have connectivity to the public network, thus enabling the possibility of performing missions Beyond visual Line of Sight (BvLoS); *(iii)* With the imminent arrival of commercial 5G deployments, the cellular network performance will improve exponentially regarding the delay and available bandwidth, enabling innovative applications that require those values.

For security reasons and as a general rule, when user equipment connects the public network (Internet), it does not do so with a public Internet Protocol (IP) address, but it accesses the network through a Network Address Translators (NAT) provided by the router (in the case of the fixed network) or the network operator (in the case of the cellular network). Therefore, the mentioned device is not reachable from externals networks and direct communication is not achievable. Although it is possible to connect to the Internet using public IP addresses, it has been decided to enable connectivity between the different devices using a Virtual Private Network (VPN) to make this framework solution more generic and approachable to the potential users but also more secure. The VPN server, which is located at 5G Telefonica Open Network Innovation Centre laboratory (5TONIC) [140] (a leading research laboratory focusing on 5G technologies based in Madrid), will relay the information/packets between all the devices connected to the VPN, thus facilitating communication. In our previous work (chapter 5), we have examined the performance impact of using a VPN service with cellular-enabled UAVs, proving that it does not penalize communication performance to a great extent and provides more benefits than disadvantages.

In order to connect the GCS and UTM domains, it is just required that the GCS is connected to the public network either using the fixed network, the cellular network, or even the satellite. In this case, it is unnecessary to use a VPN service since the interaction between the GCS and the UTM domains is performed through a web application, as explained in the previous section and shown in Figure 7.1.



(a) Validation scenario.



(b) DJI Matrice 600. (c) INTA Ground Control Station display.

Figure 7.3: Validation scenario and components.

7.2.2. Interface Manager

As detailed above, there are several alternatives for inter-domain communication. It is mandatory to maintain stable communication over time in such an environment, but specific requirements regarding the network performance must be met to carry out the mission under the established parameters. For this reason, it is necessary to have an application in each of the connected devices (UAVs and GCS) that is responsible for selecting the best communication alternative depending on the needs of the moment. This application corresponds to the Interface Manager (IM) shown in Figure 7.2.

The IM should automatically select among the different alternatives based on different network parameters such as delay, bandwidth, or packet losses. For this purpose, it is necessary to constantly monitor the status of each of the available links. At the same time, this information about the state of the links must be available in the GCS at the operator's disposal, who can manually select the preferred alternative or let the IM to automatically choose.

This application must assume that the selected links to be used may eventually be asymmetric, i.e., the UAV may use one certain link to communicate with the GCS and the GCS may use another different link instead to communicate back with the UAV. In the same way, it would also be interesting if different types of traffic could use a different link. Thus, it would be possible to isolate the types of traffic by channel, physically separating the most critical information such as command and control (tele-measurement and tele-comand) from other less critical information such as video.

Finally, it should be noted that all UAVs have the ability to include different payloads to carry out the application aspired by the user. In particular, UAVs will incorporate virtualization enabled equipment since virtualization technologies have positively shown many advantages enabling the deployment of communication services over aerial networks (see chapter 5).

7.3. Preliminary validation

In order to perform a preliminary validation of the Labyrinth communications framework (final trials have been scheduled by the end of year 2022), we have conducted some experiments in a laboratory environment. These experiments have been carried out at the Instituto Nacional de Técnica Aeroespacial (INTA), a research organization associated with the Spanish Ministry of Defense that participates in the Labyrinth project.

7.3.1. Validation details

As represented in Figure 7.3(a), the validation involves a UAV and a GCS. The selected UAV is the DJI Matrice 600 (see Figure 7.3(b)). The aircraft carries as payload an onboard computer, responsible for collecting relevant data from the UAV (e.g., velocity, position, altitude) and sending it to the GCS. This data is represented at the GCS (see Figure 7.3(c)).

Both the GCS and the UAV use a communication prototype (that runs the IM) with the basic features of the presented architecture. Two Raspberry Pi (RPi) 4 B have been selected to implement the communication prototypes. The selection of these devices has been motivated by the large community of users and the support generated. In addition, these devices have the possibility of being powered by small batteries, and the prototype can be onboarded on the UAV for preliminary validation. This equipment has been previously used in real flight tests demonstrating its ability to onboarded as the UAV payload (see chapter 5).

Among this equipment, three alternatives are considered, as depicted in Figure 7.3(a): *(i)* public network (Cellular network in the case of the UAV, and fixed network in the case of the GCS), *(ii)* WiFi, and *(iii)* radio modem. However, due to the limitations of radiating in a laboratory environment and the current state of Labyrinth user developments, the radio modem link is emulated using the Virtualized Environment for

Multi-UAV Network Emulation (VENUE) emulator (chapter 2, a UAV-oriented emulation platform. A 4G/LTE HAT is used to connect the UAV RPi to the public cellular network. As explained in the previous section, if devices connect to the Internet using private IP addresses, it is necessary to use a VPN service. This service has been deployed in the 5TONIC laboratory [140]. The VPN credentials (required to create the VPN tunnel) generated for the devices connecting to the VPN are unique and always assign the same IP addresses so that the configurations made on both devices are persistent. On the other hand, the WiFi link between the two devices is a 2.4 GHz ad hoc link.

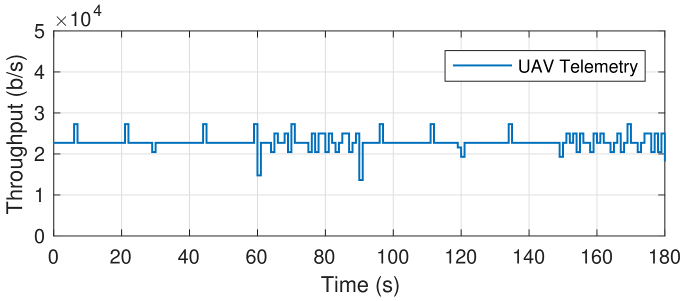
One Virtual Extensible Local Area Network (VXLAN) [68] has been created for each of the communication links. A VXLAN is a network overlay protocol designed to transport data link layer traffic over the network layer. The essential operation of a VXLAN is that it encapsulates traffic from one Local Area Network (LAN) and transports it over an IP network to a different physical LAN. This way, it is possible for devices on both networks to communicate with each other in the same way as if they were on the same LAN.

VXLANs, on the one hand, enable UAV and GCS equipment (Onboard computer and Telemetry monitor in Figure 7.3(a)) to act as if they were on the same LAN regardless of which link they are using to communicate. Therefore, changes made in the IP network (the one that transports packets to the other end of the VXLAN) are transparent for UAV and GCS computers. In addition, the fact that the UAV and GCS equipment work as if they were on the same LAN simplifies the development process for Labyrinth users, who do not have to worry about IP addressing and routing. As a result, we obtain a programmable device that allows implementation, flexibly, any policy to select links, both upstream and downstream, independently.

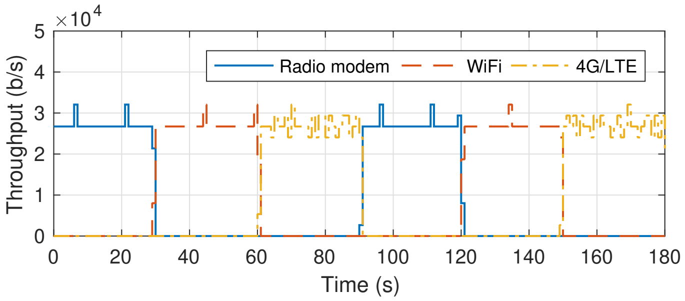
Each of the VXLANs that have been created (one per link) are attached to a Linux Bridge (see Figure 7.3(a)). On the other hand, this Linux Bridge is attached to the Ethernet interface of the RPi, which will provide connectivity to the UAV equipment. The procedure is similar in the GCS; however, only one VXLAN is attached (one per link) to the Linux Bridge to select the link to enable communications. For example, a UAV operator can switch from the public network to the WiFi at a given moment, and this procedure is done by detaching the VXLAN corresponding to the public network and attaching the VXLAN corresponding to the WiFi to the Linux Bridge. More details about VXLAN configuration can be found in our previous work [4].

7.3.2. Experiments

The UAV, using DJI's flight simulator, performs a virtual flight mission (the DJI simulator sends the flying parameters to the UAV as if it were flying) at Rozas airfield, Lugo, Spain. The Rozas Airborne Research Center (CIAR) is located at this aerodrome, where real flight campaigns will be carried out once the developments are in the validation stage (end 2022).



(a) Traffic received at Telemetry Monitor.



(b) Traffic received in the Interface Manager.

Figure 7.4: Telemetry data sent from the onboard computer (UAV) to the Telemetry Monitor (GCS).

The onboard computer collects interest data from the main computer (Control Unit in Figure 7.3(a)) of the UAV via serial link and sends them to the INTA control station in order to monitor the UAV status, such as Global Positioning System (GPS), Initial Measurement Unit (IMU), and battery load; it packages them in the corresponding format, and transmits them to the station. These packets are sent with a frequency of 10 Hz and have a predefined size of 241 bytes.

As a first approach and proof of concept, this information will be transmitted by the communication prototype. During this process, an operator at the GCS will be in charge of manually changing the links.

The traffic transmitted from the onboard computer (UAV) to the station and received at the Telemetry Monitor (GCS) can be seen in Figure 7.4(a). The traffic frames are captured in the GCS communications equipment. In Figure 7.4(b), the same traffic can be seen but captured on each one of the available interfaces. The graph shows that although the selected link has been changing during the mission (approximately every 30 seconds), the data flow has remained stable, and these changes have been transparent to the Telemetry monitor. The graph also shows that the traffic received on the interfaces is bigger than the traffic received in the station application. This is due to the overhead introduced by the use of VXLANs. This effect is more noticeable since the packets sent are quite small (only 241 bytes), and the header introduced by the VXLAN (20 bytes) although very small in general, here accounts for a significant percentage of the total size (8.2%). Finally, it can be appreciated that the telemetry flow is less stable when using the public network (4G/LTE) due to external factors.

7.4. Discussion

This chapter presents some of the main objectives of the European Labyrinth project focusing on the communication architecture. This chapter introduces the different domains required to carry out a UAV mission under the parameters established by the U-space UTM. In addition, it presents the communication architecture details.

With all these considerations in mind, initial validation is performed in a laboratory environment at the Instituto Nacional de Técnica Aeroespacial (INTA), where a proposed communication prototype connects the UAV with the GCS using different links (e.g., public network, WiFi, and radio) sending real telemetry data. The results prove that the proposed prototype operates correctly and serves as a starting point to reach the final development where all the architecture requirements will be accomplished.

This work serves as a starting point for developing the Labyrinth project communications architecture. Likewise, it opens several working and research lines that will be discussed in this section.

First, integration with the UTM server has to be carried out, both from the UAV and the GCS. It is necessary to define different details such as the report message format or the required transmitting frequency. In the same way, the messages/instructions received from the UTM server must be translated into the GCS to incorporate them into the UAV.

Other representative working line would be to integrate new communication solutions into the proposed communications prototype, such as a Radio modem or Satellite. Once we have all the available alternatives configured, it is necessary to characterize each link taking into account not only network parameters such as the available bandwidth or delay, but also other parameters such as the maximum coverage range or the energy consumption. All this information is required as input to decide which link to use in each specific case. In addition, this information must be available and constantly monitored in the GCS at the disposal of the UAV operator, who will be able to make decisions based on these parameters.

On the other hand, it would be of great interest to perform an asymmetric link selection. This means that the UAV and the GCS can use different links to communicate each other (i.e., the UAV may use the radio link while the GCS may use the public network). Likewise, different types of traffic should go through different links (at the same time), which will allow to physically isolate data that is highly critical (e.g., telemetry, commands) from traffic that is not critical for the security mission (e.g., video, payload data).

The IM can migrate towards a Software Defined Networking (SDN) paradigm [179], where the selection of links will be decided automatically from an SDN Controller (that can be located in the GCS, the UAV, or an external domain). Similarly, this SDN controller should monitor the current state of each link and decide which alternative to use. However, each UAV should have some autonomy for decision-making to avoid accidents in case of loss of connectivity.

Finally, all these developments must be tested in real flight campaigns tests to demonstrate their feasibility and functionality before carrying out the use cases proposed in the Labyrinth project.

7.5. Research impact/dissemination

The outcome of the work behind this chapter is the result of the second part of an internship at Instituto Nacional de Técnica Aeroespacial (INTA) from the Spanish Ministry of Defense (more details can be found in Appendix A).

The work in this chapter has been published in the Workshop on *Cellular UAV and Satellite communications*, located at the IEEE *Global Communications Conference (GLOBECOM)*. This article has also been presented in a seminar organized by IMDEA Networks Institute and the Telematics Engineering Department of the Universidad Carlos III de Madrid, and in a webinar organized by Labyrinth project: "Innovative applications of drones for ensuring safety in transport".

8

Communication Infrastructure Manager for fully connected UAVs

Over the past few years, the implementation of more complex Unmanned Aerial Vehicle (UAV) services (integrated into the urban environment) or even based on multiple UAVs operating collaboratively has been profusely investigated and it has led to new scenarios that are not yet adequately deployed (e.g., package delivery, monitoring of sports events or crowds such as concerts or demonstrations, communications coverage extension, support to emergency services in cities like firefighters, police, hospitals). There are in fact still several challenges to face before we can see these applications really integrated into our daily lives.

These challenges have to do on the one hand with the regulation field. For example, the European Union has recently released a new single European regulation, which affects all UAVs regardless of their weight and size and whether they are used for professional or recreational purposes. Some significant rules to comply with the new European regulation are *(i)* registering as an Unmanned Aircraft System (UAS) operator, *(ii)* accrediting UAV pilot training, or *(iii)* acquainting the different restrictions/regulations for each possible flight area. On the other hand, the challenges to be solved also have to do with the technological field.

In this chapter, we will address the latter perspective (technological field), although its relationship with air traffic management will also be discussed. In particular, this chapter handles one of the challenges that UAVs will have to face in the hyper-connected environment already present in the current Fifth Generation of cellular network technology (5G) communications ecosystem and even more so in the upcoming Sixth Generation of cellular network technology (6G) scene (all-to-all connectivity as depicted in Figure 8.1).

Moreover, new regulations will imply constant Command and Control (C2) traffic and periodical reports to the UAS Traffic Management (UTM) [180] in non-segregated airspace. In this line, authors collaborate in the framework of the Labyrinth Project, an European funding project that encompasses relevant entities in the aviation sector such as the National Institute of Aerospace Technologies (INTA), the German Aerospace Center

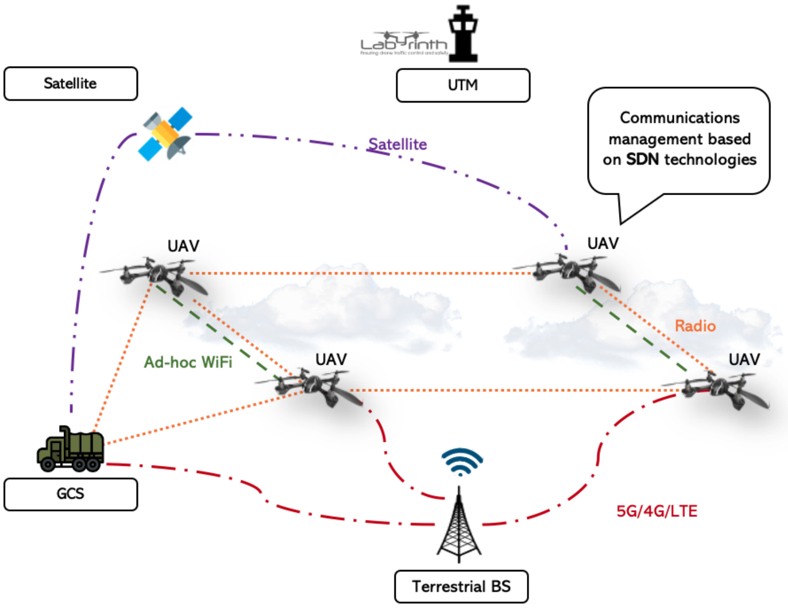


Figure 8.1: Motivating scenario.

(DLR), or EUROCONTROL, as well as outstanding users of the developed services such as the Municipal Emergency Assistance and Rescue Service and Civil Protection (SAMUR-Protección Civil) from Madrid, the General Directorate of Traffic (DGT), or the Port Authority of the Eastern Ligurian Sea (among others).

In the first place, UAVs will need to transmit many more information streams. In addition to the traditional video stream, UAVs may need to transmit *(i)* data related to aeronautical telemetry (e.g., pressure, temperature, speed, Global Positioning System (GPS) position), *(ii)* telemetry from different onboarded sensors (e.g., air quality, smoke detection), *(iii)* control information for USpace UTM, *(iv)* control information for their Ground Control Station (GCS), *(v)* control information between UAVs for swarm management, or *(vi)* information from new applications they may support.

In the second place, UAVs are going to have a considerable amount of alternatives for data transmission, including traditional radio frequency links for C2 and payload (usually in separate bands), to which will be added different backup links that may eventually be considered as primary links based on cellular technology (either 3G, 4G, 5G), satellite technology, proprietary line-of-sight links, millimeter wave, visible light communications, WiFi links, Bluetooth links, etc. In general, these technologies are not interchangeable, and each one has its field of application (e.g., long-distance point-to-point communications, networking, low energy consumption). However, given that they are increasingly being developed in lighter and smaller devices, it is not out of the question to assume that a UAV could use several simultaneously, thus multiplying its versatility.

Finally, the consequence of a highly volatile and dynamic environment such as the

UAVs operations is also relevant, both from the point of view of the intermittent availability of the transmissions themselves (e.g., coverage, obstacles, interference) and the intermittent availability of the transmission devices (e.g., transmission failures, UAV batteries running out, new devices, whether UAVs or GCSs that dynamically join or leave the communications network) which continuously forces communication networks topology reconfigurations. This heterogeneity of such a broad spectrum, but also so focused on the discontinuity of the service, raises multiple challenges in the provision of these services, which in many cases are expected to reach, beyond reliability (correct operation), extensive levels of robustness, reliability, and resilience (3R services), insofar as they are services that are very useful in all kinds of emergency operations, operations in challenging locations, or expensive operations that are difficult to repeat.

This chapter presents the design and validation of the Communication Infrastructure Manager (CIM) whose main objective is to manage all the complexity related to the assignment of flows to interfaces. This task will involve a series of rules that establish this relationship since, in a 3R environment, it will be essential to continuously monitor the links to establish the metrics to be applied in each circumstance dynamically and adaptive. Moreover, the solution to this challenge also involves establishing a network infrastructure that allows all those implicated in the communication to interconnect, extending as far as possible the number of paths available to a given destination. For the CIM development, we started from a prior simple version (chapter 7). The CIM has been upgraded to operate with Software Defined Networking (SDN) technology (more details can be found in section 8.2).

The rest of the chapter is organized as follows: Section 8.1 reviews the state of the art and background. Section 8.2 presents the system design. Section 8.3 present three proof-of-concept experiments that showcase the potential of the CIM in multi-UAV scenarios. Finally, section 8.4 concludes the chapter.

8.1. Related work and background

Proper communication is one of the most crucial factors for reliable UAV operations. However, we encounter poor links from some UAVs to the control station on numerous occasions because of the multi-UAV mission's essence. This challenge is exponentially multiplied in the military domain operating with tactical UAVs, potentially encountering unplanned/unexpected conditions. The authors in [181] present a tactical data link that guarantees control reliability in emergencies and introduces tactical data links UAVs as communication relays. The authors in [182] propose using other UAVs that are part of the system as communication relays to enable two-hop connectivity and enhance extensive reliability.

Nowadays, UAVs can carry as payload much more than simple communication gadgets thanks to the rapid evolution in the miniaturization of devices. This fact allows UAVs to become powerful Internet-of-Things (IoT) components, offering not only sensing, but also communication service, and onboard data analysis. For example, authors in [183] employ LoRa, WiFi, and LTE networks to provide the UAV system's with broadband and cellular wireless network support. This multi-interface concept is increasingly present in the multi-UAV environment and will be treated in this paper. Authors in [184] propose a solution for UAVs in which 4G and 5G cellular technologies are combined to maximize the operating range in areas where commercial 5G is not yet deployed. Their system, therefore, uses the best available alternative based on two radio parameters such as the Reference Signal Received Power (RSRP) and Reference Signal Received Quality (RSRQ). In fact, the utilization of cellular technologies in UAVs is becoming more and more relevant, literally turning the UAVs into flying base stations or mobile end users.

The adoption of softwarization technologies has been recently raised in the UAS research area. The recent arrival of the 5G ecosystem has not only brought new radio technology as commonly thought but has also incorporated key software technologies such as Network Function Virtualization (NFV) and SDN. Our previous work has explored the use of NFV [19], analyzing its strengths and weakness. Additionally, SDN has opened up new opportunities to automate the management of telecommunication services and verticals and facilitate achieving the performance requirements imposed on 5G and beyond. For this reason, this chapter includes some of the SDN contributions to the UAS research area.

SDN brings significant advantages to multi-UAV environments, such as a centralized global network view or maintaining the status of the whole network, allowing for a flexible network reconfiguration based on different metrics. Therefore, forwarding decisions in each device may not be based just on local conclusions but on what happens in the rest of the network. SDN can help to addressing several inherent challenges of multi-UAV networks [40]. SDN can reconfigure the whole network (due to the global view of the SDN controller) under any topological changes, which are quite frequent due to the intrinsic mobility of UAVs. SDN can also use different strategies for packet forwarding beyond the destination Internet Protocol (IP)/MAC address, taking into account different parameters such as payload or packet types. SDN networks can also help to avoid network interference since it enables path/channel selection at any given moment. Furthermore, SDN may select which channels are more reliable, not only based on instant parameters (e.g., signal level) but also based on other representative information that can not be considered in traditional routers (e.g., UAV mission planning input, historical data).

These advantages mentioned above are just a slight representation of what the SDN paradigm can bring to multi-UAV networks. However, the design of legacy SDN technology was not done to fit into such a specific environment, like multi-UAV networks with so many particularities. Consequently, several challenges/issues must still be solved in order to involve SDN in this framework; in this regard, both academia [185] [186] and industry have increased their interest in this topic during the last years: authors in [187] propose a SDN and MQTT hybrid system for battlefield multi-UAV environments. Their proposal adapts to the periodic swarm topological changes, supports flexible data transmission among payloads, and improves swarm security. Authors in [188] design a SDN framework for the UAV backbone network. This framework monitors the UAV network to manage and analyze the available information effectively. Authors in [189] investigate the deployment of a SDN UAV network providing a communication service. This article considers the placement of the SDN controller and its implications in terms of communication overhead and end-to-end delay. As the envisioned development of this chapter, the last references employ the SDN ecosystem as a key enabler technology for the correct operation of their systems.

8.2. Communication Infrastructure Management

In order to maintain control over all the interfaces available in the devices (both the UAV and GCS) as well as the establishment and configuration of the communication networks, we developed the Communication Infrastructure Manager (CIM) (the evolution of the Interface Manager (IM) developed in Chapter 7). The CIM employs SDN with the primary objective of keeping a global vision of the network that allows it to select the most suitable communication alternative at any given moment based on different metrics. To study the feasibility and functionality of the CIM in multi-UAV environments, we propose a reference scenario (depicted in Figure 8.2). The following subsections describe the steps to bring the system up, covering all the implementation details and explaining the scenario components in-depth.

8.2.1. Reference scenario

Figure 8.2 shows the reference scenario presented to analyze the performance of the CIM. The reference scenario includes the three domains present in the Labyrinth project (UAV, GCS, and UTM) for a correct operation. Additionally, it includes the external services domain required for enabling communications using the public Internet.

As can be appreciated, the CIM entity (shadowed in grey) is located within the UAV and GCS domains since both domains may include several communication alternatives. The figure indicates that each domain includes four communication alternatives in this particular case. However, the number of interfaces may vary depending on each case. The figure also shows the SDN controller (located in the GCS) that acts as the network's brain.

The following subsections detail the required step to configure the reference scenario for further experimentation properly.

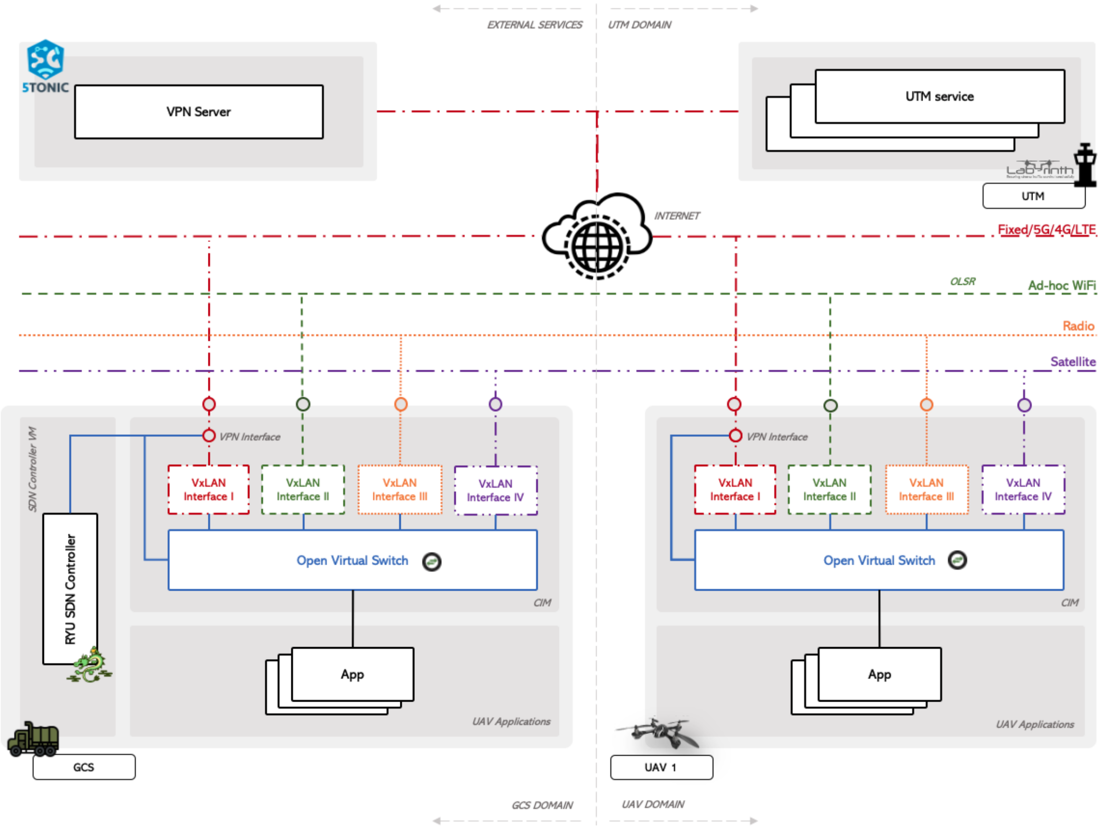


Figure 8.2: Reference scenario.

8.2.2. Network layer connectivity

The first step is to enable connectivity at the network level (IP level) between the UAV and GCS domains. This procedure may be straightforward in some cases; however, it may require additional configuration in others. Each device may have different interfaces (e.g., 5G, proprietary radio, WiFi, satellite) and must be appropriately configured.

For example, to enable multi-hop communication in WiFi (ad hoc), it is required to utilize a routing protocol. This way, when two devices are not in the WiFi range but there is a path using other intermediate nodes as a relay, communication at the network level will be enabled anyway. In our case, Optimized Link State Routing (OLSR) [190] protocol has been used. OLSR is a well-known legacy MANET protocol that enables multi-hop connectivity, which has been exemplified in Chapter 2 performing acceptably for the proposed scenarios.

Another representative example is cellular communication, which in our case is basically the end to end communication between two devices using the public Internet. Typically, devices connect to the Internet using a private IP address (configured by the Internet provider) and a Network Address Translator (NAT). Therefore, these devices can browse the Internet but are not accessible from external machines. In order to solve this mentioned problem, a Virtual Private Network (VPN) server has been configured. Therefore, all devices connect to the VPN server, which acts as a relay to enable device -to-device connectivity. The performance does not drop significantly when using the VPN server as a relay (this effect has previously been studied in chapter 4). The VPN server has been deployed at the 5G Telefonica Open Network Innovation Centre laboratory (5TONIC). This server (that can be seen in Figure 8.2) is considered an external service. This means that they are services that are not in the GCS, the UAV, or the UTM domain but are required for everything to work as expected.

8.2.3. Data link layer connectivity (on top of Network Layer)

Once the device-to-device connectivity has been appropriately configured for all the available interfaces (section 8.2.2), the goal is now to transmit data link layer traffic over the network layer (i.e., Ethernet traffic over an IP network) using overlay networks. The reason is because overlay networks provide an isolated domain, i.e., the traffic transmitted over the overlay is isolated from any other traffic going over the network (e.g., VPN, OLSR, Network monitoring app). Furthermore, creating an overlay requires no additional configuration of the intermediate nodes.

In our case, we use Virtual Extensible Local Area Network (VXLAN) [68], a network overlay protocol designed for this purpose. In this way, traffic entering a VXLAN tunnel exits at the other end(s) of the tunnel as if both devices were on the same Local Area Network (LAN). Additional details of VXLANs configuration can be found in our previous work [4].

8.2.4. SDN switches

At this point, to enable inter-domain communications (from GCS and UAV apps in Figure 8.2), it is necessary to deploy a programmable switch to forward traffic from the domains (GCS/UAV apps in Figure 8.2) over the overlay networks (VXLANs in Figure 8.2). This programmable switch has different ports to which the applications and VXLANs are connected. Therefore, specific rules should be installed on this switch to forward incoming and outgoing traffic to the desired port.

Open Virtual Switch (OvS) has been selected for this purpose. OvS is one of the most popular implementations of virtual programmable switch agreeing with OpenFlow. The basic functionality can also be realized with Linux Bridges, as seen in the previous version detailed in Chapter 7. However, the operation of OvS, which can be through as a SDN controller (explained in the following subsection 8.2.5) or standlone, provides unquestionable advantages. Firstly, by using OvS and a SDN Controller, it is feasible to react dynamically to the high rate of changes in the network (very common in multi-UAV environments) by modifying the network configuration. On the other hand, OvS offers great granularity by adding different rules that operate simultaneously.

8.2.5. SDN controller

A SDN controller in a SDN network takes the role of the brain of the SDN network. As we have advanced in section 8.2.4, the rules installed in the OvS for traffic to be forwarded between the different ports can be standalone or through a SDN controller. The main advantage of using a SDN controller is that these forwarding rules can be modified as needed when significant changes in the network that invalidate previous configurations occur.

The controller and the OvSs communicate via the OpenFlow protocol (control plane). In our case, we have decided to deploy the SDN controller in a virtual machine in the GCS. This controller is in the GCS because there is no space limitation in the equipment used. It is then possible to have equipment with better resources that allow more developments to be deployed. The SDN controller selected is RYU. RYU is an open-source SDN controller that provides software components and Application Programming Interface (API) that facilitate the development of SDN management and control.

We have decided to use the cellular interface to communicate the SDN controller and the different OvS. This is because cellular connectivity will always be available during the missions in the Labyrinth project (where the correct functioning of this scenario will be tested). Given this assumption, it is logical to ask why not always use the cellular network since it is always available in our scenarios. First, if we always use this technology, we can consume all the available resources, resulting in losing critical C2 information. Secondly, the delay of this network is higher than other technologies considered (e.g., WiFi, Radio Frequency (RF)). Thirdly, it relies on the public network to send data/video that may be private/sensitive in some circumstances.

8.2.6. Network monitoring application

As explained in section 8.1, legacy SDN technologies such as OpenFlow have not been designed with the special considerations/characteristics of multi-UAV environments in mind. For example, SDN switches (section 8.2.4) react to different events in a wired network. When one of these events occurs, the corresponding switch reports the SDN controller (using the OpenFlow protocol), which in turn, with this information and those of the other switches in the network, can make the appropriate decisions. However, these events do not operate correctly in decentralized wireless networks, such as the ad hoc WiFi networks that we often find in multi-UAV environments. For example, due to the nature of ad hoc wireless networks, it is impossible to determine when a network participant is reachable or not [191], except with an external application such as the one we need to implement.

To solve this problem, we have implemented a *network monitoring application* that collects different metrics not only about the network status but also from the status of the device that hosts the switch. This information is periodically sent to the controller, whether there are changes or not. With this information the SDN controller acquires real-time knowledge about what is ensuing in the network. In this case, each network participant (UAVs and GCS) has a built-in Python script that collects the different metrics. Since the *network monitoring application* is custom built-in Python, it can collect many metrics (as many as programmed). Some examples are the available bandwidth, the jitter, or the resilience. For this study, we consider the following metrics:

- Connectivity [Boolean]: the connectivity metric indicates whether connectivity exists between two network participants (for all the available interfaces)[1].
- Round Trip Time [ms]: this metric indicates the current Round Trip Time (RTT) value between two hosts (for all the available interfaces). This measurement is performed using the Linux *ping* tool.
- RX/TX Traffic [b/s]: this metric indicates amount of traffic (received and transmitted) on each available network interface (e.g., WiFi, 5G, 4G/LTE).
- Power consumption [W]: this metric indicates the current power consumption of each host. This measurement is performed using the UM34C USB multimeter.

This information coming from the *network monitoring application* is useful not only to the SDN controller (to make the appropriate forwarding decisions at any given moment). It is also of special interest to other players such as GCS operators, who can observe in real-time what is happening in the network. On the other hand, this information is stored in a database for two primary purposes: *(i)* to perform an exhaustive ex-post analysis if something has not worked as expected, and *(ii)* as a communication flight recorder system (a.k.a black box) in the case of an accident or emergency.

The *network monitoring application* has been released as open-source [192]. The scripts have been developed for Ubuntu 18.04 LTS but can be easily adaptable to any operative system. The data is transmitted in JSON format to incorporate custom metrics efficiently.

8.3. Validation: Practical experiences

This section presents three Proof-of-Concept (PoC) experiments that show the potential of the CIM in multi-UAV environments. For this purpose, a testbed has been set

---

[1] It should be noted that the connectivity metric does not only refer to neighboring nodes, i.e., single-hop communication but also considers multi-hop communication.

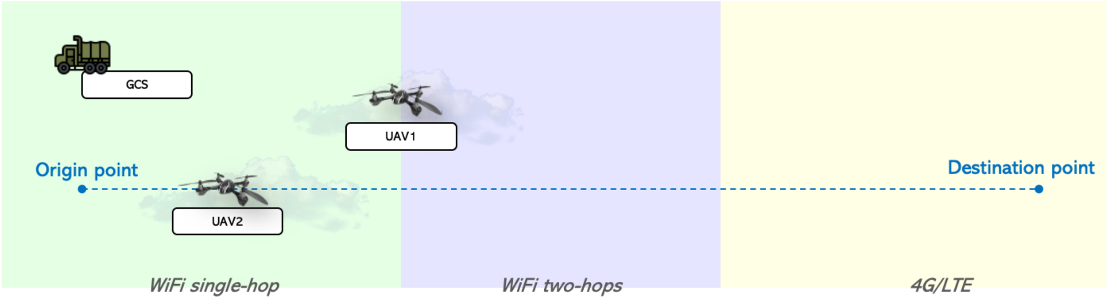


Figure 8.3: PoC 1 scenario.

up at (5TONIC). This testbed aims to be a stable version of the H2020 Labyrinth Project ecosystem located at 5TONIC premises. The main objective is to test new developments and potential improvements (e.g., CIM progress, new communication alternatives) during the project period before attempting them on prototypes on-boarded into real UAVs. For this purpose, miniITX computers (8 CPUs, 8 GB RAM, and 128 GB for storage) have been used, which provide an ideal platform for development. These configurations are relatively smoothly transferred to smaller devices that can be embedded, such as Raspberry Pi (RPi) Single Board Computer (SBC), as demonstrated in chapter 5.

This testbed (following the system explained in section 8.2) includes a GCS and two UAVs. All of them have the CIM installed and have two available interfaces. At 5TONIC laboratory, we have cellular communication (4G/LTE commercial deployment and 5G New Radio) and WiFi. Because the flight tests to be conducted in the Labyrinth project (end of 2022) will take place in locations where commercial 5G has not yet been deployed, it has been decided to use commercial 4G/LTE connectivity to give verisimilitude to the future experiments. On the other hand, the WiFi channel and the movement of the UAVs (flight trajectories) have been emulated using the VENUE (chapter 2) platform. The WiFi channel could have been configured with the built-in WiFi interface of the devices. However, we would not provide movement to the UAVs in that case. Accordingly, with the native WiFi, we would not be able to test the changes between one-hop and two-hop connectivity or those cases where WiFi connectivity is impossible.

8.3.1. Choosing the minimum RTT path

This first experiment attempts to prove the performance of the CIM following a SDN strategy to select the fastest (taking into account the RTT) communication alternative (among the available ones) in a scenario where the network topology changes frequently. The experiment includes a GCS and two UAVs (UAV1 and UAV2). The GCS and UAV1 remain static, while UAV2 follows the trajectory (dashed blue line) depicted in the Figure 8.3 from *Origin point* to *Destination point* and vice versa. Thus, at some point, *(i)* the UAV2 can directly communicate either using WiFi with the GCS (green background), or *(ii)* using WiFi via the UAV1 (blue background), or *(iii)* it has to employ the 4G/LTE (yellow background) when the WiFi is out of range. The UAV2 sends a continuous 1.2 Mb/s stream to the GCS to test the operation. Of this 1.2 Mb/s, 1 Mb/s corresponds to an HD video stream, while the remaining 200 Kb/s corresponds to telemetry information sent by the UAVs to the station. This value (200 Kb/s) has been estimated by analyzing the telemetry generated by the DJI UAVs used in the Labyrinth program.

In this experiment, one of the objectives of the CIM (it can be any other depending on the mission needs) is to manage the forwarding plane of the programmable elements (i.e., the OvS switches in the UAV and the GCS) to select the shortest path considering the RTT. For this reason, whenever there is a WiFi path available (even if it has to go through UAV1), this will be the one preferred (always faster than using the public cellular network). Thanks to the report received by the *Network Monitoring Application* installed in each UAV and GCS, this information is available at the SDN controller.

As shown in Figure 8.4, during the beginning of the experiment, the stream is received over the WiFi interface as the UAV2 and the GCS are in range. From the second 38 onwards, this information is no longer received over the WiFi interface because the UAV2 goes out of range, and the one-hop communication is no longer available. However, two-hop WiFi communication is available, although it will not be possible until the routing protocol (OLSR as explained in section 8.2.2) discovers the path. While this process is in progress, the SDN controller install a rule in UAV2 to transmit flow is over the 4G/LTE interface.

From the second 52 onwards, the protocol enables the multi-hop path, and the WiFi interface becomes available again (indicated as well by the SDN controller). From the second 65, the UAV2 is no longer in the WiFi coverage range of the UAV1, so the only possible way to communicate UAV2 and GCS is through the 4G/LTE interface.

From the 160th second, UAV2 is back in range of UAV1's WiFi coverage. However, the change does not occur until second 170, which corresponds to the moment when the OLSR protocol converges and enables the path. From the 180th second, UAV2 is back in range of the GCS and communicates using the WiFi interface with one hop. Unlike the previous case (when the UAV2 is moving away), in this case, there are no communication breaks. This behavior is because, in this case (when approaching), alternatives do not disappear, but the process is to move towards the optimal option according to the RTT. As mentioned above, the interface changes are managed by the SDN controller.

RTT measurements have been taken from the UAV2 to the GCS using the *ping* tool to verify this behavior. This value is reflected in the background of Figure 8.4. When the figure is shaded with green, the average RTT value is 10 ms. This period corresponds to one-hop WiFi connectivity. When the figure is shaded with blue, the average RTT value is 21 ms. This period corresponds to two-hop WiFi connectivity. When the figure is shaded with yellow, the average RTT value is 98 ms. This period corresponds to 4G/LTE

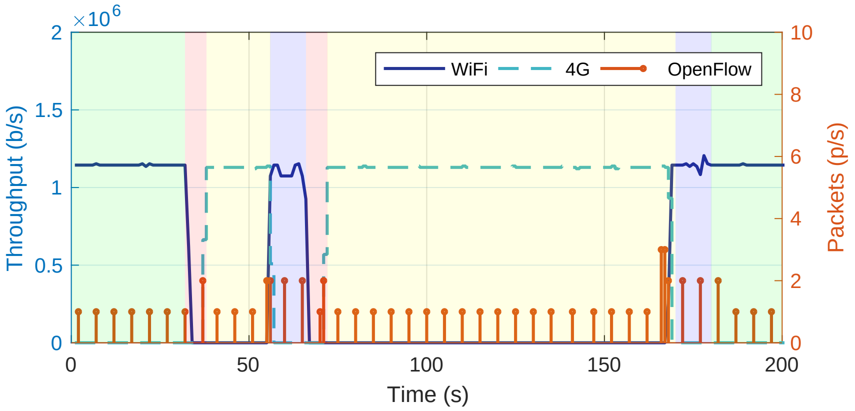


Figure 8.4: Traffic from UAV2 captured in the GCS (Red background - Losses).

connectivity. Finally, connectivity is not possible when the figure is shaded with a red background.

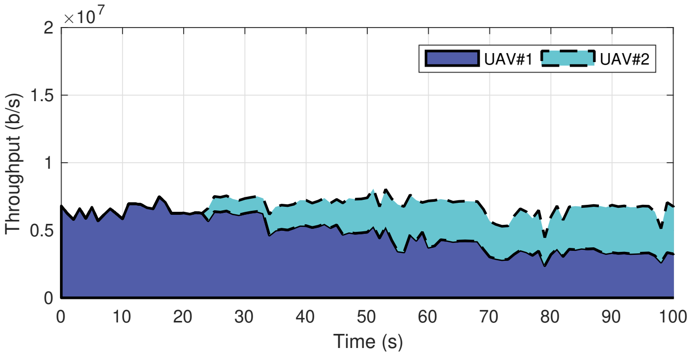
Figure 8.4 also displays the OpenFlow messages exchanged between the GCS, the UAV2, and the SDN controller (right y axis). As it can be noticed, a periodic OpenFlow message is sent every 5 seconds (2 Kb/s). In the same way, when a rule is installed in an OvS, there is an increase in OpenFlow packets.

Let us compare the performance of this system, which includes the CIM combined with the SDN approach, and uses all the available interfaces to communicate the UAV and GCS domains with a traditional system that only uses WiFi technology. We can see a significant difference because there are 12 seconds of communication losses using the CIM, compared to the 118 seconds for a traditional system where the 4G interface is not used. Since connectivity between GCS and UAV2 is always possible (using different alternatives), we obtain a performance of 94% compared to the ideal scenario using the CIM-SDN approach. When we only used the WiFi interface (traditional system), the performance decreased to 41%. On the other hand, another strategy would have always been to use the 4G/LTE interface since it is assumed that coverage is available throughout the UAV2 trajectory. However, the average RTT would have been 100 ms in this case, while using the CIM-SDN approach, the average RTT is 65 ms (excluding the losses periods).

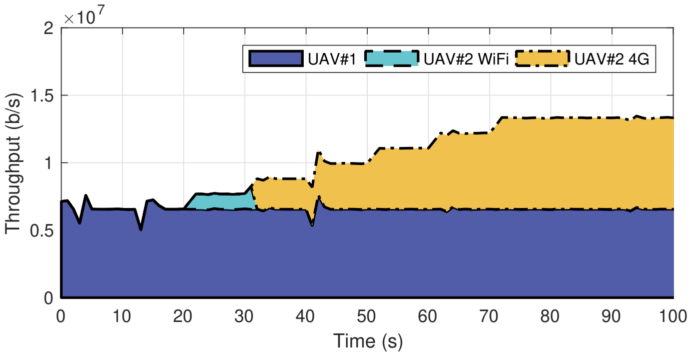
8.3.2. Load balancing

The main goal, in this second experiment, is to use the CIM to distribute the traffic coming from each of the UAVs over different interfaces to perform a load balancing strategy that maximizes the Quality of Service (QoS) and minimizes packet losses.

In an earlier analysis, we have identified that the maximum bandwidth available on the WiFi ad hoc interface (emulated channel using VENUE) is roughly 7 Mb/s. Therefore, if the sum of the incoming traffic received from UAV1 and UAV2 over the GCS WiFi



(a) Traffic from UAV2 captured in the GCS without using CIM-SDN approach.



(b) Traffic from UAV2 captured in the GCS using the CIM-SDN approach.

Figure 8.5: Traffic from UAV2 captured in the GCS for the load balancing experiment

interface is higher than 7 Mb/s, it is reasonable to expect data loss.

For this second experiment, the UAVs remain static (hovering), so the topology/grid of the system remains stable (initial position in Figure 8.3). To demonstrate the potential of the CIM, we have performed the same experiment twice. Firstly, it uses the logic of a traditional system, only communicating over the WiFi, and secondly, using the CIM system and using the other available interfaces (WiFi and 4G/LTE). The experiment runs as follows: The UAV1 sends a constant stream of 6 Mb/s to the GCS; from the second 20, the UAV2 starts sending a flow of 1 Mb/s that will increase by 1Mb/s every 10 seconds, i.e., at the second 30, UAV2 will send 2 Mb/s until reaching the 6Mb/s sent by the UAV1.

As shown in Figure 8.5(a) with a traditional system (traffic captured at the GCS), losses start to appear from the second 30 onwards because the sum of incoming traffic at the GCS exceeds 7 Mb/s at its WiFi interface. Therefore, although the UAV2 periodically increases the data flow, the total received at the GCS remains stable, always close to the maximum bandwidth available.

However, as shown in Figure 8.5(b) with CIM, from second 30, the SDN controller using the information available from the *network monitoring application* installed in the GCS, discovers that the traffic received at the GCS WiFi interface is close to the maximum available. At that moment, the SDN controller installs a rule in the UAV2 OvS so that traffic with GCS destination is forwarded through the 4G/LTE interface. In this way, throughput increases, and no packet loss can be appreciated.

It should be noted that the maximum bandwidth capabilities of both the WiFi interface and the 4G/LTE interface may vary due to different environmental factors. However, although performed under controlled conditions (laboratory experience), this experiment serves as a PoC to demonstrate that some QoS (both in RTT and packet loss) can be guaranteed using the CIM and the SDN approach.

8.3.3. Real flight campaign

Finally, we scheduled an experiment with a real UAS from the Spanish National Institute of Aerospace Technology (INTA). In the first place, this test aims to validate the correct functionality of CIM and the ease of transferring the implemented software to different aerial platforms.

For this purpose, we mainly operated with the INTA UAS. The UAS includes a mobile GCS (depicted in Figure 8.6(a)). The INTA GCS communication elements are onboarded in a shelter truck that can move to different locations depending on the UAS operation.

The INTA's UAV (for the Labyrinth project) is a DJI Matrice 600 (depicted in Figure 8.6(b)). The Matrice 600 is the DJI's top-of-the-range aerial platform designed for professional aerial and industrial applications. This unit has a built-in onboard computer (as payload) called DJI MANIFOLD, a Linux-based SBC that allows the incorporation of the CIM software at the same time it introduces the capacity of developing the Labyrinth project requirements (e.g., the specific report from UAV to GCS, the specific reports from UAV to UTM).

This experiment also includes a second UAV provided by the Universidad Carlos III. This UAV is a Parrot Bebop that includes a RPi SBC as a payload. In the same way as the DJI MANIFOLD, the RPi allows to include the developments of both CIM and the Labyrinth project requirements.

In addition to the communication alternatives contemplated above, this experiment includes a proprietary RF link that communicates the INTA UAV and INTA GCS provided by Silvus Technologies. This RF link has a much greater range than WiFi (over 10 kilometers). However, the RF technology has a more significant delay and more considerable power consumption. Therefore, it makes sense to use WiFi for the UAV operations whenever it is close to the GCS.

The experiment (depicted in 8.7) will have a similar trajectory to the one performed in the first experiment (section 8.3.1). The INTA UAV will depart from a source point in the WiFi range of the INTA GCS. After a certain distance, the INTA UAV will no longer have one-hop WiFi with the GCS but will be able to communicate through the UC3M UAV as a relay. Later, it will need to activate the RF link to communicate with the

(a) INTA GCS.

(b) INTA UAV: DJI M600.

Figure 8.6: INTA UAS

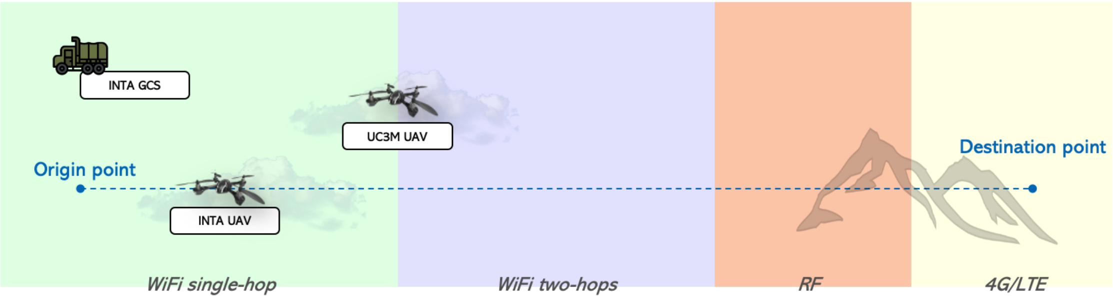


Figure 8.7: Real flight scenario.

GCS. Finally, the INTA UAV will enter an area with no line of sight where the RF link is impossible, requiring the 4G/LTE. All these steps have been detailed in Figure 8.7. The expected results are similar to those obtained in 8.3.1. However, it will be necessary to analyze in detail since field experimentation may vary compared to the laboratory results.

In addition to communicating the UAV and GCS domains, the necessary reports are performed from both the UAV and GCS to the UTM. This fact is because the flight will take place in non-segregated airspace, where it is necessary to make periodical reports so that the UTM is aware of the UAS operation continuously.

Testing will take place in September 2022 at Rozas Airfield, coinciding with the final experiments of the Labyrinth project.

8.4. Discussion

This chapter introduces the Communication Infrastructure Manager (CIM) for multi-interface UAV environments following an SDN approach. It also presents a reference scenario, where all the stages to properly achieve communication between the domains through all the available interfaces are detailed.

Moreover, once the system is introduced, two laboratory experiences have been reported to showcase the CIM prospect compared to traditional UAV systems where the communication between UAV and GCS domains only involves the radio interface. These experiments have produced a testbed available in the 5TONIC laboratory, where all the Labyrinth project advances will be tested. In the same way, this testbed will be used to work on the different research lines listed below.

In addition, a system prototype has been incorporated into a real INTA UAS. Flight tests of this prototype (coinciding with the Labyrinth project's final trials) will be carried out in September 2022.

8.5. Research impact/dissemination

The CIM solution has been included in the Labyrinth project communications prototype. Flight tests (coinciding with the Labyrinth project's final trials) will be carried out in September 2022. Moreover, the prototype will be integrated into actual UAS communication equipment from professional manufacturers (also project participants) such as INTA and Arquímea. The project also includes relevant members from both industry and academia in the consortium, such as DLR, DGT, Telefónica, Samur, Eurocontrol, DIN, Port of la Spezia, AIT, PONS, and INNCOME.

This work has been carried out in collaboration with INTA engineers, continuing the connections established during the research internship (detailed in Appendix A). As a result, an article has been submitted to the *IEEE Communication Magazine* and is currently under revision process.

PART V

CONCLUSIONS

> *"Drones overall will be more impactful than I think people recognize in positive ways to help society."*
>
> Bill Gates

9

Conclusions and future work

This chapter details the general conclusions of the thesis, which are a compendium of the individual conclusions of the previous chapters. Secondly, it enumerates different future research lines to continue the investigation begun during this Ph.D.

9.1. Conclusions

One of the main contributions of the thesis is Virtualized Environment for Multi-UAV Network Emulation (VENUE) (Chapter 2), an emulation platform for the validation of network services deployed in multi-Unmanned Aerial Vehicle (UAV) systems. VENUE operates not only with virtual entities (e.g., Virtural Machine (VM), Linux Containers (LXC) but also with real hardware that can be directly used in real infrastructure afterward. This functionality reduces prototyping and validation cycles and reduces the time-to-market or time-to-operation period. This platform is showcased in two different scenarios: *(i)* to validate Flying Adhoc Network (FANET) routing protocols and *(ii)* to evaluate the feasibility of Network Function Virtualization (NFV) deployments in multi-UAV networks with such unique features. This platform is also employed throughout the thesis contributions, highlighting that VENUE is currently the core of the virtualization platform used to validate the advancement/improvements of the communications components of the MILANO system (Chapter 3). This fact has allowed us to virtualize specific communication components and network systems and open research lines related to the virtualization of network functions in tactical UAV of this relevance. It should be taken into account that the VENUE platform is available as open-source contribution [58].

On the other hand, this thesis makes different contributions with a practical approach, aiming to solve different issues identified as some of the most significant challenges during the beginning of the Ph.D.

First, we have integrated a multi-UAV-NFV testbed with a newly designed Fifth Generation of cellular network technology (5G) access network, the outcome of the 5GRANGE project targeting a cell radius of 50 km with 100 Mb/s at the edge and closing the connectivity gap in remote and rural/underserved areas. This work has been performed in collaboration with INATEL, a Brazilian institution where rural/underserved areas of the country have infrequent access to the Internet. This work shows the whole experimentation scenario that has been built, integrating both solutions and presenting them in two different domains that can interact with each other. The work is also showing the methodology to perform this type of complex integration with different domains located in remote locations. The final integration has been done at 5G Telefonica Open Network Innovation Centre laboratory (5TONIC) premises, which holds a 5G core functionality, allowing the integration of the different network domains. This contribution is detailed in Chapter 4.

Secondly, a practical experience of cellular-assisted UAVs enabling the control of Beyond Radio Line-of-Sight (BRLoS) fleets is presented as a contribution (Chapter 5). This research demonstrates that using commercial equipment in combination with the 4G/LTE commercial cellular network, the Key Performance Indicator (KPI)s defined by the 3rd Generation Partnership Project (3GPP) for a correct BRLoS operation are met. However, some improvement points (regarding mostly to upcoming multimedia applications) are identified, which will be solved with the massive implementation of the commercial 5G network. This fact is demonstrated with a benchmark of the 5G Stand Alone (SA) network available the 5TONIC laboratory. In addition, indoor flight tests have been carried out to corroborate the proposal's validity.

Thirdly, we face the problem of providing long-duration services (in the order of hours) with small UAVs with limited autonomy (in the order of minutes) (Chapter 6). In this regard, a mathematical formulation is presented that demonstrates the complexity of the problem (NP-hard), showing at the same time that it is a challenge that needs to be solved and needs efficient solutions. In this sense, we propose the Betweenness Centrality Heuristic Algorithm (BETA), which is compared not only with the optimal solution but also with the baseline solution to provide a valid settlement to this problem.

Finally, this thesis presents the Communication Infrastructure Manager (CIM). The CIM is an entity in charge of creating the network infrastructure between the different domains (UAV, Ground Control Station (GCS), UAS Traffic Management (UTM)) that we find in an Unmanned Aircraft System (UAS). Additionally, the CIM also distributes the traffic/flows between all the different communication alternatives available. Its development is covered from its beginnings to its current state based on Software Defined Networking (SDN) technology (Chapters 7, and 8). This CIM uses as input the contributions of all the previous chapters of the thesis giving sense and shape to the previous work. In addition, it is worth mentioning that the CIM has been selected to be incorporated in two actual UAV prototypes of the Labyrinth project and will be tested in flight campaigns in the final trials of the project.

These contributions have resulted in 13 research publications with 112 citations by different research articles. Of these articles, eight have been published in Journal Citation Reports (JCR) journals; four have been published in international and Spanish conferences (including a best paper award in *CNERT IEEE INFOCOM workshop*). An additional article is now under review.

9.2. Future work

With the idea of persisting in contributing to the growth of communications in UAV networks, different future research lines have been identified during this thesis:

Lightweight virtualization: one of the main lines of research to be continued is the virtualization of network services and functions. This virtualization will be carried out with lightweight alternatives such as Kubernetes (already introduced in Chapter 5), a portable and extensible open-source platform for managing workloads and services. Kubernetes provides a container-centric management environment, making it possible to virtualize network services in low-resource environments such as payloads of onboard devices in UAVs. In this way, we obtain a flexible service, giving rise to the possibility of providing communication solutions with different functionalities depending on the requirements/needs of each user. To enable correct operation, it is not only necessary to employ Kubernetes, but it is also required to modify its native/default operation since, as usual, this solution is not designed for multi-UAV networks. Therefore, this solution will complement: *(i)* an SDN network in charge of routing the data packets and *(ii)* an overlay network manager, enabling communications between the different services in a fully customizable and tailored form.

Energy: during this Ph.D., we analyze the sources of energy consumption regarding the communications in multi-UAV environments. Moreover, we propose a solution to replace UAVs to extend the network's life beyond the battery life. However, there has not been an in-depth exploration of solutions and mechanisms that save energy consumption or efficient ways to perform transmissions (e.g., use different interfaces depending on energy consumption, radiate at more or less power). In this sense, this exploration and the proposal of new solutions is an essential assignment. In addition, the virtualization research line should be fed by this research line since it would also be interesting, from an energy consumption point of view, to deploy virtual services in those UAVs where virtual services affect the life of the network the least.

Integration: one of the most direct and most tangible future work lines is the integration of the thesis proposals with actual UAV prototypes. Throughout the Ph.D., we have tried to make the research as practical as possible and carry out the developments for different UAV models. However, it would be necessary for these testbeds and use cases to move a step further than a Proof-of-Concept (PoC) experiment. In this way, the functionality of the proposals is to be analyzed from an operational point of view by UAV operators, which may contextualize the progress assembled with the research and identify weaknesses where further work will undoubtedly be needed.

Appendices

A

INTA Internship

In order to collaborate in the framework of the European Labyrinth project and promote the already existing links between the University Carlos III of Madrid and the IMDEA Networks Institute with the research center Instituto Nacional de Técnica Aeroespacial "Esteban Terradas", as well as the acquisition and transfer of knowledge between researchers of both institutions, a 5-month doctoral internship has been carried out (July 1 through November 30, 2021).

A.1. About INTA

The National Institute for Aerospace Technology "Esteban Terradas" (commonly known as INTA) is an autonomous Spanish organization considered a public research institution, attached to the Secretary of State for Defense of the Ministry of Defense. INTA specializes in dual research and technological development in different fields such as aeronautics, space, hydrodynamics, security, and defense. Its headquarters are located in Torrejón de Ardoz, Madrid, having other Centres in Madrid, Ávila, Granada, Huelva, Lugo and the Canary Islands. Furthermore, INTA has representatives in Europe, such as Seconded National Experts in the European Maritime Safety Agency (EMSA) and the European Space Agency (ESA), and is a member of EREA, the Association of European Research Establishments in Aeronautics.

INTA's main activities and functions are *(i)* to develop aerospace technologies at the system, subsystem, and components level; *(ii)* to provide technical advisory to official organizations and private companies; *(iii)*to act as a technological center for the ministry of Defense; *(iv)* to manage heavily instrumented test ranges in El Arenosillo (Huelva) and Rozas (Lugo); *(v)* to perform functional, environmental and electromagnetic compatibility tests for qualification and certification.

Regarding the aeronautical field, since 1992, INTA has developed four complete Unmanned Aircraft System (UAS) obtaining the Airworthiness Certificate for

Experimentation:

- **SIVA**: Medium size tactical Unmanned Aerial Vehicle (UAV) system of 300 kg Maximum Take-Off Weight (MTOW), five hours endurance and 150 km range, designed as an investigation platform and as a reconnaissance and aerial surveillance system. Between 2006 and 2014, the SIVA was operated by the Spanish Army as a training system. Since 2012 the Spanish Air Forces have been using the Matacán Airbase system to train military operators. The platform has been used to develop *(i)* an IP LOS communication system, a *(ii)* a structural health monitoring system (INESASSE). Both developments were financed in a competitive call.

- **ALO**: Small size tactical UAV system of 60 kg MTOW and three hours endurance and 65 km range, designed as investigation platform. The system was used in the DEMORPAS project financed by Single European Sky ATM Research (SESAR) providing an operational experience of unmanned traffic management in non-segregated airspace.

- **DIANA**: Target UAV of 150 kg MTOW, 1-hour endurance and 65 km range used by the Brazilian Airforce for anti-aircraft artillery systems training

- **MILANO**: Medium altitude long endurance system of 1000 kg MTOW, 20 hours endurance, and able to perform beyond the line of sight operations with a Satellite. The system has also been developed with the dual aim of an investigation platform and reconnaissance and operational surveillance system.

A.2. Summary of activities

This section summarizes the activities realized in collaboration with INTA, not only during the internship but also during the Ph.D:

- Design of the IP communications architecture of the SIVA system.

- Design of the IP communications architecture of the MILANO system.

- Development of a virtualization platform to test/validate new developments related to communication components.

- TCP/IP teaching courses for INTA staff.

- Field support during UAS SIVA flight campaigns.

- Development and integration of the DJI M600 communication system with the UAS Traffic Management (UTM)(in the framework of the European project Labyrinth).

Figure A.1: MILANO UAS at Rozas airfield.

A.3. Research outcome

The research has been carried out in collaboration with the Unmanned Aerial Platforms Area team to identify and analyze new lines of research and development in TCP/IP communications systems for their implementation in the different UAS of INTA. Moreover, the conducted research was positively assessed by the INTA Subdirectorate General of Aeronautical Systems, as it aligns with INTA's strategic objectives in its Strategic Plan 2021-2025 for dual research at the service of society.

As a result of this collaborative work, we have published two research papers: *(i)* at the *IEEE Global Communications (Globecom) 2021* conference in the "*Cellular UAV and Satellite Communications workshop*" track and *(ii)* at the *IEEE International Conference on Computer Communications (INFOCOM) 2022* conference in the "*International Workshop on Computer and Networking Experimental Research using Testbeds*" track, which has been selected as best paper award. A third paper, also resulting from the internship, has been submitted to *IEEE Communication Magazine.*

B

Efficient data dissemination

During the thesis, the efficient dissemination of telemetry data originating from UAVs (related to aeronautical and application data) to the Ground Control Station (GCS) has been discussed several times. This topic is included in all the document chapters, but it is critical in Chapters 6 and 8 for the proposed solutions to operate correctly. In this regard, this appendix analyzes the potential of data-centric communications to enhance collaborative operations via efficient information sharing and build systems supporting flexible UAV mission objectives. In particular, we analyze the primary contributions to data dissemination in UAS that can be given by Data Distribution Service (DDS) open standard as a solid and industry-mature data-centric technology.

This appendix presents a practical experience to validate the feasibility and the advantages of using DDS in a multi-UAV deployment scenario. For this purpose, we use a UAS testbed built up by heterogeneous hardware equipment, including some interconnected small UAVs and actual equipment for a tactical UAS from the Spanish Ministry of Defense.

B.1. DDS practical experience

This appendix presents a practical experience on the deployment of DDS in a UAS testbed (see Figure B.1). The proposed testbed includes real network equipment from the TCP/Internet Protocol (IP) communication system of an existing tactical surveillance system called SIVA [69], an UAS from the Spanish Ministry of Defense.

The UAS testbed integrates network equipment corresponding to the SIVA UAV and the SIVA GCS (see Figure B.1). The UAV router is an embedded system following the PC/104 industry standard (Intel Core i7 2.3 GHz, 8 GB RAM, 128 GB SSD, 4GbE ports, operating with Ubuntu 16.04.3 LTS, 64 bits). This system integrates an Ethernet switch that allows the interconnection of local on-board equipment, such as a *data acquisition unit*) and a camera. It also integrates a GbE port that implements a data-link with

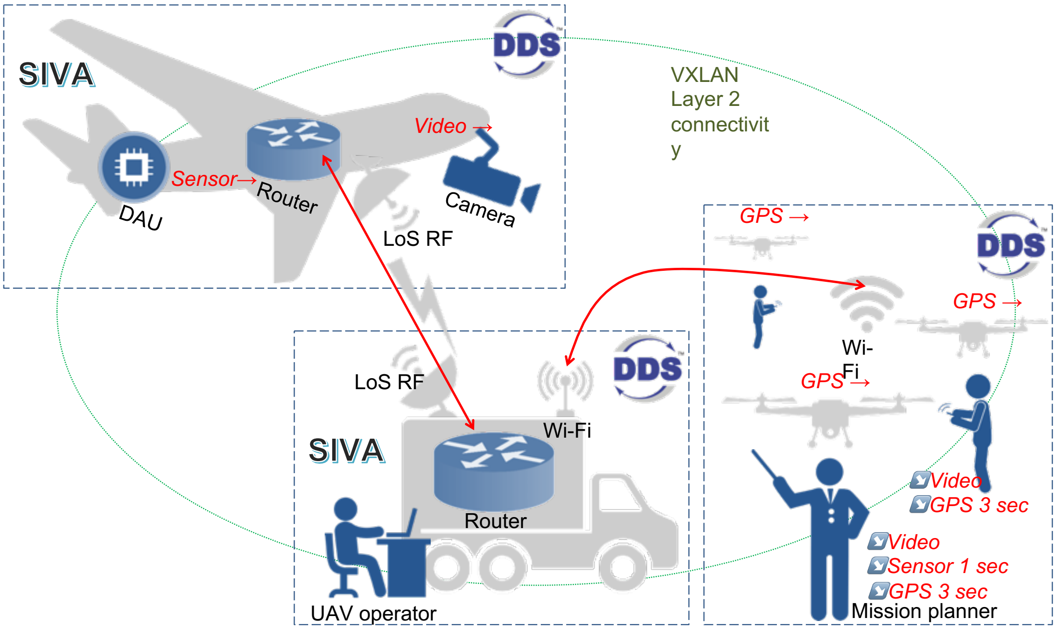


Figure B.1: Validation testbed including tactical UAV, GCS and small UAVs.

the GCS. The *data acquisition unit*, deployed as a computer in the testbed, is capable of serving different telemetry patterns towards a workstation in the GCS (emulating information captured from the different on-board sensors of the UAV, and also the video stream captured from the camera).

The GCS contains a barebone computer (Advanced Micro Devices, Santa Clara, CA, USA Athlon II X2 3 GHz processor, 8 GB memory, 500 GB hard disk) with 3 GbE ports, as the GCS router, operating under the same Linux distribution as the UAV router. The GCS router provides network connectivity to a workstation that receives and processes the telemetry information from the UAV (UAV operator in Figure B.1), and provides a Wi-Fi interface that supports the attachment of external devices (e.g., external remote video terminals, that may receive the UAV telemetry, or even a set of small-sized UAVs). The detailed description of our UAS testbed can be found in [69].

To support experimentation activities with DDS and networks of UAVs, we extended this testbed with a number of hardware and software components. First, we incorporated three small-sized UAV (Parrot AR.Drone 2.0), interconnected through a Wi-Fi ad hoc network. Each of them transports a single board computer as payload, in particular a Raspberry Pi (RPi) 3 Model B and, in a real deployment, would be capable of flying and landing on a desired area to provide a regular Wi-Fi access point to mobile users over the operation field (or perform a stationary flight with the same purposes). The Wi-Fi access points and the Wi-Fi ad hoc network are supported through wireless interfaces available at the RPis. Secondly, we installed RTI ConnextR DDS Professional software in the different components of the UAS, so that the tactical UAV can share its telemetry information (sensor data and video) using DDS with specific Quality of Service (QoS) constraints. Small-sized UAVs can also publish data to be accessible over the deployment area, such as the Global Positioning System (GPS) coordinates corresponding to their current location. All the information made available through DDS will promptly be accessible to the users of the infrastructure, following the dynamic discovery mechanisms of DDS, and will be shared according to matching QoS requirements expressed by publishers and subscribers. In our experiments, we consider three users: *(i)* a UAV operator, who works at the GCS and is in charge of monitoring and managing the operation of the tactical UAV; *(ii)* a user coordinating the mission that requires the UAS deployment (shown as the mission planner in Figure B.1), who gets network access connectivity using a Wi-Fi access point provided by a small-sized UAV and *(iii)* a user that is carrying out some field operation at a distant area, and gets network access connectivity through a second small-sized UAV. The communications among small-sized UAVs, as well as between each of these UAVs and the GCS, is provided via the Wi-Fi ad hoc network (in particular, one of the small-sized UAVs acts as a network relay between the GCS and the other two small-sized UAVs).

To support the utilization of network-level multicast in our testbed, and this way take advantage of this technology to support the fast delivery of DDS messages to multiple receivers, we configured all the client devices of our infrastructure to be in the same Local Area Network (LAN) domain, using Virtual Extensible Local Area Network (VXLAN) technologies [68] and Linux bridges. The configuration, which is illustrated in Figure B.2, guarantees that: *(i)* every end-user terminal connected through a Wi-Fi access point (provided by a small-sized UAV) gets attached to the same LAN segment, and therefore belongs to the same broadcast domain; *(ii)* the UAV router, the GCS router and the RPis themselves have a virtual interface to the LAN, hence they can also behave as DDS publishers and subscribers over the broadcast domain (in particular, each aerial vehicle includes a Data Acquisition Unit (DAU) that publishes data using the DDS middleware); and *(iii)* multicast/broadcast data originated by the tactical UAV is sent towards the GCS, where it is replicated by a VXLAN interface towards the small-sized UAVs that provide a Wi-Fi access point (i.e., multiple copies of the same multicast/broadcast data are not transmitted over the UAV-GCS communication channel).

With the aforementioned configuration, we used the DAU at the tactical UAV to publish a continuous data stream, emulating the delivery of real-time video from the UAV camera at a constant bit rate of 5 Mb/s (this was supported with the utilization of the RTI Perftest tool[1]). This DAU also offered sensor information from the UAV (e.g., representing temperature, humidity, air pressure, etc.) to interested subscribers, producing telemetry values every 4 seconds. This feature was provided with the implementation of a specific-purpose DDS application, producing periodic raw sensor values with a configurable

---

[1] RTI Perftest , accessed Tuesday 15th November, 2022. Available: https://github.com/rticommunity/rtiperftest

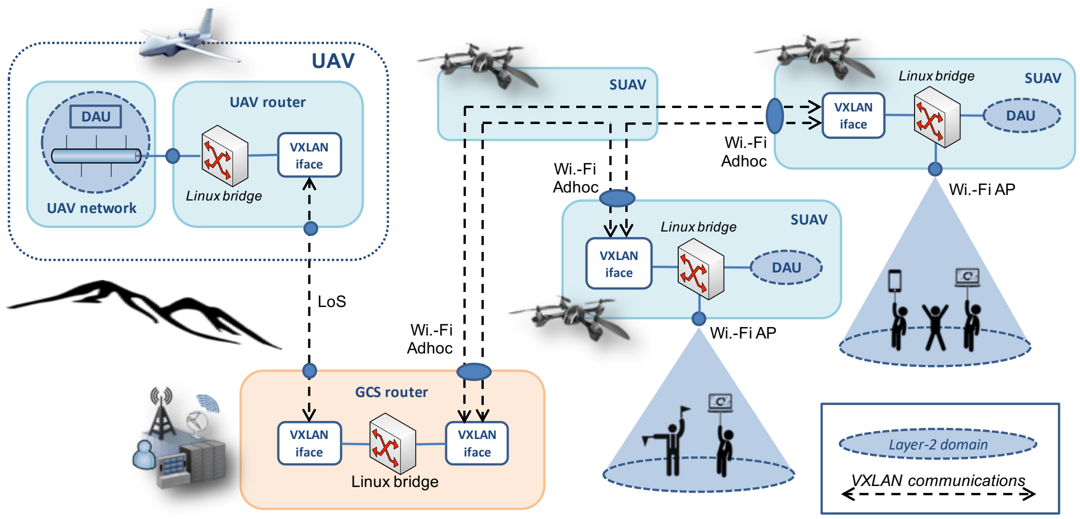


Figure B.2: VXLAN and Linux bridge configuration.

time interval. The application was also used to serve telemetry information from the DAUs of small-sized UAVs, emulating the publication of the updated GPS coordinates corresponding to each of these UAVs every 8 seconds.

In our experiments, the small-sized UAVs were positioned on the ground. A mobile ground station (the mission planner in Figure B.1) subscribes to the sensor information of the tactical UAV, as well as to the GPS positions of all the small-sized UAVs, specifying QoS requirements that match the publisher offers. The user working at a distant area subscribes to the video stream of the tactical UAV using matching QoS requirements. Figure B.3 reflects the telemetry traffic that appears in the network and the DDS signaling traffic coming from the publishers and the subscribers (this traffic has been captured at the equipment of the mission planner). DDS discovery and subscription messages are sent using specific multicast IP addresses. Hence they are distributed over the broadcast domain, and can be observed at the mission planner equipment.

Figure B.3a shows the different DDS messages sent by the publishers. An endpoint discovery phase takes place between Second 0 and Second 60, with different discovery messages being sent related first to the sensors, then to the GPS data, and then to the video stream. For the sake of clarity, Figure B.3a only shows the discovery messages corresponding to the GPS position of one of the RPis. Figure B.3b shows the related messages sent by the subscribers, i.e., the mission planner and the user working at a distant area. In addition, we can see in both figures how the different publishers and subscribers enter a steady state when the subscriptions match the discovery messages. This is observed after approximately Second 20, in the case of the sensor information; after Second 40 for GPS data; and after Second 50 for the video stream (to better observe this effect in our experimental setup, subscriptions have been configured to take place at

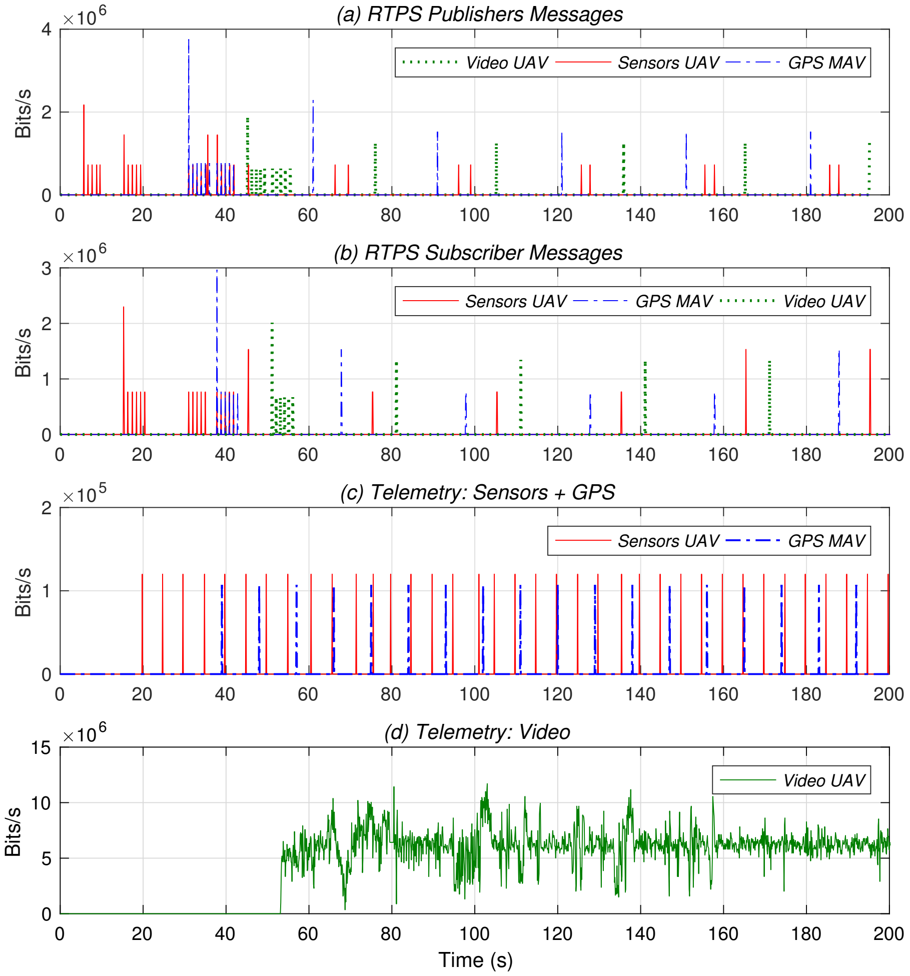


Figure B.3: DDS signaling (a,b); and data exchange (c,d).

those instants).

Once the corresponding subscriptions have been carried out, the delivery of data starts from publishers to subscribers. This is shown in Figure B.3c, which represents the data traffic corresponding to the sensor information and the GPS data received by the mission planner. Analogously, Figure B.3d represents the data traffic corresponding to the video camera of the tactical UAV. For simplicity, and similar to the previous case, the figure only shows the data received from one of the RPis. Our practical results confirm that DDS can effectively support the flexible deployment of applications and services, and eases the sharing of information and situational awareness in UAS deployments, through the utilization of application-defined topics and the automated discovery of the published information, which can be requested with specific QoS requirements.

As we have commented, a straightforward implication of our virtual layer-2 network configuration is that all the communicating endpoints will be sharing the same diffusion domain, so the advantages provided by multicast delivery can easily be acquired without the need of further multicast specific infrastructure (e.g., the execution of a multicast routing protocol at the RPis). As a disadvantage of our VXLAN-based configuration, we can mention that all multicast/broadcast control and data traffic is in fact received (and eventually discarded) by all the terminals in the broadcast domain. Although this issue could be addressed with the integration of well-known techniques to learn on active multicast subscriptions on layer-2 network segments [193], it has however been considered to present an assumable cost in this type of critical scenarios, where such redundancy may be beneficial for instance in terms of better handover performance (i.e., when a mobile station changes its Wi-Fi access point).

To verify this, we carried out a second experiment, where the user working at the distant area moves and changes its Wi-Fi access point to the network, obtaining network access connectivity via the same small-size UAV that serves the mission planner. Figure B.4 represents the throughput of data traffic perceived by the user in this experiment, where we can see an interruption on the reception of the real-time video stream. This interruption corresponds to the time required to execute the attachment procedures to the new wireless access point. Once these procedures are completed, the user continues receiving the video content, as this content is also being served through the new access point.

Finally, we used the deployed infrastructure to perform an experiment aiming at evaluating the effective throughput that can be obtained using DDS. For this purpose, we used the RTI Perftest tool to estimate the throughput that can be provided by the DDS middleware to applications, considering application-level data units (that is, the number of data that are published at a time by a DDS application) of two different sizes: 1000 bytes and 64 bytes. For each of these sizes, we used the RTI Perftest tool at the tactical UAV to publish data at different rates (1 Mb/s, 2 Mb/s, 4 Mb/s, 8 Mb/s and

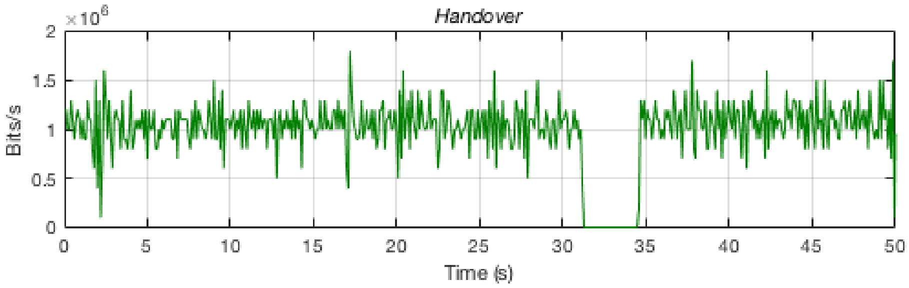


Figure B.4: Interruption of the reception of real-time video due to handover.

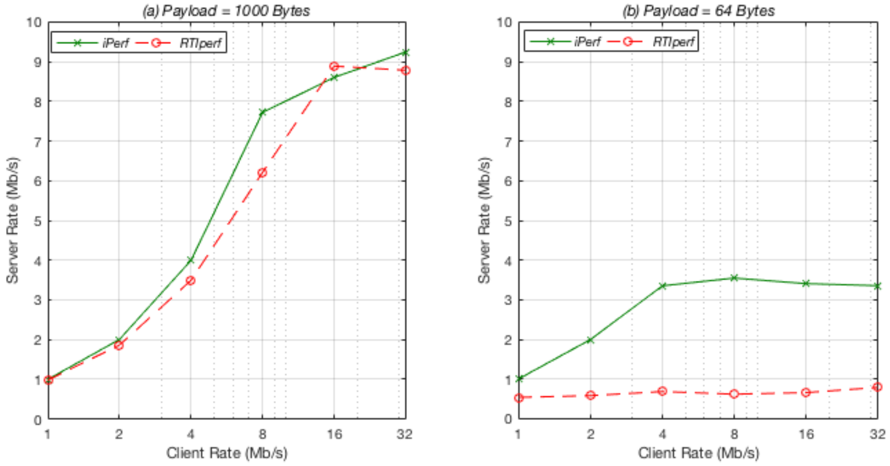


Figure B.5: Measurements of the achieved throughput with: large PDU (left); and small PDU (right).

32 Mb/s), with another instance of the tool subscribing to that information from one of the client devices. For each rate and size of the application-level data unit, we measured the throughput received at the subscriber. To better frame the efficiency achieved by DDS, we repeated the same experiment using the *Iperf* tool, which can be used to send data traffic over User Datagram Protocol (UDP) at the aforementioned rates, using the same size of the application-level data unit. The results of these experiments are shown in Figure B.5. In this figure, we observe that, for a large size of the application-level data unit, the performance of DDS closely follows that of UDP; on the contrary, with a low size of the application-level data units, the performance of DDS decreases, due to the data overhead introduced by the DDS middleware. In any case, we want to highlight that this expected decrease of performance does not prevent the utilization of DDS as the experiment represents an extreme situation (an application writing a continuous stream of reduced-size data units), which can be easily avoided in practice with a convenient design of the DDS application (for instance, using topics that enable the publication of larger or aggregated data units).

B.2. Main findings

DDS has the potential to enable interoperable UAS deployments. This fact is supported by the dynamic discovery of information publishers and subscribers and the negotiation of a flexible set of QoS policies, which need to be satisfied for successful publisher-to-subscriber communication. In addition, DDS eases the integration of new software and services with the utilization of a standardized middleware platform, enhancing the adaptability of the UAS to different mission purposes.

However, there are also some limitations regarding the utilization of network-level multicast (an enabling technology in DDS). To address the identified limitations in our motivating scenarios, we have to use VXLANs, which, although they work correctly in the reference scenario, create an additional layer of complexity at the configuration level that has to be taken into account.

References

[1] V. Sanchez-Aguero, B. Nogales, F. Valera, and I. Vidal, “Flexible services deployment using small unmanned aerial vehicles for emergency situations,” *XXXIII Simposium Nacional de la Union Cientifica Internacional de Radio (URSI’18)*, 2018.

[2] I. Vidal, P. Bellavista, V. Sanchez-Aguero, J. Garcia-Reinoso, F. Valera, B. Nogales, and A. Azcorra, “Enabling multi-mission interoperable uas using data-centric communications,” *Sensors*, vol. 18, no. 10, p. 3421, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/10/3421

[3] I. Vidal Fernández, V. Sánchez Agüero, F. Valera Pintor, B. Nogales Dorado, J. Cabezas, C. Vidal, A. Lopez, D. Gonzalez, J. Diez, L. Berrazueta *et al.*, “Milano: una visión futura para un uas táctico,” 2018. [Online]. Available: https://e-archivo.uc3m.es/handle/10016/28959

[4] B. Nogales, V. Sanchez-Aguero, I. Vidal, and F. Valera, “Adaptable and automated small uav deployments via virtualization,” *Sensors*, vol. 18, no. 12, p. 4116, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/12/4116

[5] C. Tipantuña, X. Hesselbach, V. Sánchez-Aguero, F. Valera, I. Vidal, and B. Nogales, “An nfv-based energy scheduling algorithm for a 5g enabled fleet of programmable unmanned aerial vehicles,” *Wireless Communications and Mobile Computing*, vol. 2019, 2019. [Online]. Available: https://www.hindawi.com/journals/wcmc/2019/4734821/

[6] V. Sanchez-Aguero, F. Valera, B. Nogales, L. F. Gonzalez, and I. Vidal, “Venue: Virtualized environment for multi-uav network emulation,” *IEEE Access*, vol. 7, pp. 154 659–154 671, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8880644

[7] B. Nogales, I. Vidal, V. Sanchez-Aguero, F. Valera, L. F. Gonzalez, and A. Azcorra, “Automated deployment of an internet protocol telephony service on unmanned aerial vehicles using network functions virtualization,” *JoVE (Journal of Visualized*

Experiments), no. 153, p. e60425, 2019. [Online]. Available: https://www.jove.com/es/t/60425/automated-deployment-an-internet-protocol-telephony-service-on

[8] V. Sanchez-Aguero, F. Valera, I. Vidal, C. Tipantuña, and X. Hesselbach, "Energy-aware management in multi-uav deployments: Modelling and strategies," *Sensors*, vol. 20, no. 10, p. 2791, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/10/2791/htm

[9] V. Sanchez-Aguero, I. Vidal, F. Valera, B. Nogales, L. L. Mendes, W. Damascena Dias, and A. Carvalho Ferreira, "Deploying an nfv-based experimentation scenario for 5g solutions in underserved areas," *Sensors*, vol. 21, no. 5, p. 1897, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/5/1897

[10] V. Sanchez-Aguero, L. F. Gonzalez, F. Valera, I. Vidal, and R. A. López da Silva, "Cellular and virtualization technologies for uavs: An experimental perspective," *Sensors*, vol. 21, no. 9, p. 3093, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/9/3093

[11] V. Sanchez-Aguero, M.-A. Fas-Millan, F. Valera, I. Vidal, A. Paniagua-Tineo, R. A. Lopez da Silva, and J. M. Manjon, "Multi-interface network framework for uav management and data communication," *IEEE Globecom Workshops (GC Wkshps): Workshop on Cellular UAV and Satellite Communications*, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9682019

[12] V. Sanchez-Aguero, F. Valera, I. Vidal, B. Nogales, J. Cabezas, and C. Vidal, "A virtualization approach to validate services and subsystems of a male uav," *IEEE INFOCOM Workshops: The 9th International Workshop on Computer and Networking Experimental Research using Testbeds.*, 2022.

[13] V. Sanchez-Aguero, F. Valera, I. Vidal, B. Nogales, and G. del Rio Daniel, "Software defined rpa networks," *Submited to IEEE Communication Magazine.*, 2022.

[14] V. Sanchez-Aguero, B. Nogales, F. Valera, and I. Vidal, "Investigating the deployability of voip services over wireless interconnected micro aerial vehicles," *Internet Technology Letters*, vol. 1, no. 5, p. e40, 2018.

[15] B. Nogales, V. Sanchez-Aguero, I. Vidal, F. Valera, and J. Garcia-Reinoso, "A nfv system to support configurable and automated multi-uav service deployments," in *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, 2018, pp. 39–44.

[16] R. Fanou, V. Sanchez-Agüero, F. Valera, M. Mwangi, and J. Coffin, "A system for profiling the ixps in a region and monitoring their growth: Spotlight at the internet

frontier," *International Journal of Network Management*, vol. 29, no. 2, p. e2056, 2019.

[17] L. F. Gonzalez, I. Vidal, F. Valera, V. Sanchez-Aguero, B. Nogales, and D. R. Lopez, "Nfv orchestration on intermittently available suav platforms: challenges and hurdles," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 301–306.

[18] L. F. Gonzalez, I. Vidal, F. Valera, B. Nogales, V. Sanchez-Aguero, and D. R. Lopez, "Transport-layer limitations for nfv orchestration in resource-constrained aerial networks," *Sensors*, vol. 19, no. 23, p. 5220, 2019.

[19] I. Vidal, B. Nogales, F. Valera, L. F. González, V. Sanchez-Aguero, E. Jacob, and C. Cervelló-Pastor, "A multi-site nfv testbed for experimentation with suav-based 5g vertical services," *IEEE Access*, vol. 8, pp. 111 522–111 535, 2020.

[20] L. F. Gonzalez, I. Vidal, F. Valera, and V. Sanchez-Aguero, "A comparative study of virtual infrastructure management solutions for uav networks," in *Proceedings of the 7th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, 2021, pp. 13–18.

[21] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis, "A review on uav-based applications for precision agriculture," *Information*, vol. 10, no. 11, p. 349, 2019.

[22] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous uav surveillance in complex urban environments," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 2. IEEE, 2009, pp. 82–85.

[23] C. C. Haddal and J. Gertler, "Homeland security: Unmanned aerial vehicles and border surveillance." Library of Congress Washington DC Congressional Research Service, 2010.

[24] S. Alizadehsalehi, I. Yitmen, T. Celik, and D. Arditi, "The effectiveness of an integrated bim/uav model in managing safety on construction sites," *International journal of occupational safety and ergonomics*, vol. 26, no. 4, pp. 829–844, 2020.

[25] Y. Ham, K. K. Han, J. J. Lin, and M. Golparvar-Fard, "Visual monitoring of civil infrastructure systems via camera-equipped unmanned aerial vehicles (uavs): a review of related works," *Visualization in Engineering*, vol. 4, no. 1, pp. 1–8, 2016.

[26] M. Eichleay, E. Evens, K. Stankevitz, and C. Parker, "Using the unmanned aerial vehicle delivery decision tool to consider transporting medical supplies via drone," *Global Health: Science and Practice*, vol. 7, no. 4, pp. 500–506, 2019.

[27] J. Grzybowski, K. Latos, and R. Czyba, “Low-cost autonomous uav-based solutions to package delivery logistics,” in *Advanced, Contemporary Control.* Springer, 2020, pp. 500–507.

[28] G. Skorobogatov, C. Barrado, and E. Salamí, “Multiple uav systems: A survey,” *Unmanned Systems*, vol. 8, no. 02, pp. 149–169, 2020.

[29] B. Yang, T. L. Hawthorne, M. Hessing-Lewis, E. J. Duffy, L. Y. Reshitnyk, M. Feinman, and H. Searson, “Developing an introductory uav/drone mapping training program for seagrass monitoring and research,” *Drones*, vol. 4, no. 4, p. 70, 2020.

[30] A. Srivastava and J. Prakash, “Future fanet with application and enabling techniques: Anatomization and sustainability issues,” *Computer Science Review*, vol. 39, p. 100359, 2021.

[31] M. F. Khan, K.-L. A. Yau, R. M. Noor, and M. A. Imran, “Routing schemes in fanets: A survey,” *Sensors*, vol. 20, no. 1, p. 38, 2019.

[32] A. Bujari, C. E. Palazzi, and D. Ronzani, “Fanet application scenarios and mobility models,” in *Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, 2017, pp. 43–46.

[33] H. Hildmann and E. Kovacs, “Using unmanned aerial vehicles (uavs) as mobile sensing platforms (msps) for disaster response, civil security and public safety,” *Drones*, vol. 3, no. 3, p. 59, 2019.

[34] P. Fanta-Jende, D. Steininger, F. Bruckmüller, and C. Sulzbachner, “A versatile uav near real-time mapping solution for disaster response–concept, ideas and implementation,” *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 429–435, 2020.

[35] M. Banić, A. Miltenović, M. Pavlović, and I. Ćirić, “Intelligent machine vision based railway infrastructure inspection and monitoring using uav,” *Facta Universitatis, Series: Mechanical Engineering*, vol. 17, no. 3, pp. 357–364, 2019.

[36] 5GPPP, *KPIs*, 2020 (accessed February 5, 2021), https://5g-ppp.eu/kpis/.

[37] B. Li, Z. Fei, and Y. Zhang, “Uav communications for 5g and beyond: Recent advances and future trends,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, 2018.

[38] L. Tomaszewski, R. Kołakowski, and P. Korzec, “On 5g support of cross-border uav operations,” in *2020 IEEE International Conference on Communications Workshops (ICC Workshops).* IEEE, 2020, pp. 1–6.

[39] M. Marchese, A. Moheddine, and F. Patrone, “Iot and uav integration in 5g hybrid terrestrial-satellite networks,” *Sensors*, vol. 19, no. 17, p. 3704, 2019.

[40] L. Gupta, R. Jain, and G. Vaszkun, “Survey of important issues in uav communication networks,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2015.

[41] C. T. Cicek, H. Gultekin, B. Tavli, and H. Yanikomeroglu, “Uav base station location optimization for next generation wireless networks: Overview and future research directions,” in *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS).* IEEE, 2019, pp. 1–6.

[42] W. H. Maes and K. Steppe, “Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture,” *Trends in plant science*, vol. 24, no. 2, pp. 152–164, 2019.

[43] Z. Zhen, D. Xing, and C. Gao, “Cooperative search-attack mission planning for multi-uav based on intelligent self-organized algorithm,” *Aerospace Science and Technology*, vol. 76, pp. 402–411, 2018.

[44] A. Mukherjee, V. Keshary, K. Pandya, N. Dey, and S. C. Satapathy, “Flying ad hoc networks: A comprehensive survey,” in *Information and Decision Sciences.* Springer, 2018, pp. 569–580.

[45] *HOWTO Use Linux Containers to set up virtual networks*, accessed September 19, 2019. [Online]. Available: https://www.nsnam.org/wiki/HOWTO_Use_Linux_Containers_to_set_up_virtual_networks

[46] L. Gupta, R. Jain, and G. Vaszkun, “Survey of important issues in uav communication networks,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2015.

[47] M. Pužar and T. Plagemann, “Neman: A network emulator for mobile ad-hoc networks,” *Research report http://urn. nb. no/URN: NBN: no-35645*, 2005.

[48] M. A. To, M. Cano, and P. Biba, “Dockemu–a network emulation tool,” in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops.* IEEE, 2015, pp. 593–598.

[49] J. Zhang and Z. Qin, “Taprouter: an emulating framework to run real applications on simulated mobile ad hoc network,” in *Proceedings of the 44th Annual Simulation Symposium.* Society for Computer Simulation International, 2011, pp. 39–46.

[50] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, “Rotors—a modular gazebo mav simulator framework,” in *Robot Operating System (ROS)*. Springer, 2016, pp. 595–625.

[51] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[52] M. Schmittle, A. Lukina, L. Vacek, J. Das, C. P. Buskirk, S. Rees, J. Sztipanovits, R. Grosu, and V. Kumar, “Openuav: A uav testbed for the cps and robotics community,” in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018, pp. 130–139.

[53] M. Mueller, N. Smith, and B. Ghanem, “A benchmark and simulator for uav tracking,” in *European conference on computer vision*. Springer, 2016, pp. 445–461.

[54] K. Singh and A. K. Verma, “Experimental analysis of aodv, dsdv and olsr routing protocol for flying adhoc networks (fanets),” in *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. IEEE, 2015, pp. 1–4.

[55] ——, “Applying olsr routing in fanets,” in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*. IEEE, 2014, pp. 1212–1215.

[56] A. Nayyar, “Flying adhoc network (fanets): Simulation based performance comparison of routing protocols: Aodv, dsdv, dsr, olsr, aomdv and hwmp,” in *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*. IEEE, 2018, pp. 1–9.

[57] A. Nawaz H., “A study of mobility models for uav communication networks,” in *3C Tecnología. Glosas de innovación aplicadas a la pyme. Edición Especial, Mayo 2019, pp. 276–297*, 2019.

[58] V. Sanchez-Aguero, *VENUE*, accessed September 19, 2019. [Online]. Available: https://github.com/vsaguero/VENUE

[59] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, “Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives,” *IEEE Access*, vol. 7, pp. 81 057–81 105, 2019.

[60] M. O. Kalinin, V. Krundyshev, and P. Semianov, “Architectures for building secure vehicular networks based on sdn technology,” *Automatic Control and Computer Sciences*, vol. 51, no. 8, pp. 907–914, 2017.

[61] R. Kumar, M. A. Sayeed, V. Sharma, and I. You, "An sdn-based secure mobility model for uav-ground communications," in *International Symposium on Mobile Internet Security.* Springer, 2017, pp. 169–179.

[62] Z. Kaleem and M. H. Rehmani, "Amateur drone monitoring: State-of-the-art architectures, key enabling technologies, and future research directions," *IEEE Wireless Communications*, vol. 25, no. 2, pp. 150–159, 2018.

[63] T. Clausen and P. Jacquet, "Rfc3626: Optimized link state routing protocol (olsr)," *Experimental, http://www. ietf. org/rfc/rfc3626. txt*, vol. 51, 2003.

[64] *Ryu SDN Framework*, accessed September 19, 2019. [Online]. Available: https://osrg.github.io/ryu/

[65] G. Litvinov, A. Leonov, and D. Korneev, "Applying static mobility model in relaying network organization in mini-uavs based fanet," in *2018Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO).* IEEE, 2018, pp. 1–7.

[66] *OpenStack: Open Source Cloud Computing Infrastructure*, 2021 (accessed May 31, 2021), https://www.openstack.org/.

[67] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler *et al.*, "Sip: session initiation protocol," 2002.

[68] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks," RFC 7348, Aug. 2014. [Online]. Available: https://rfc-editor.org/rfc/rfc7348.txt

[69] I. Vidal, F. Valera, M. A. Díaz, and M. Bagnulo, "Design and practical deployment of a network-centric remotely piloted aircraft system," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 22–29, 2014.

[70] C. Barrado, M. Boyero, L. Brucculeri, G. Ferrara, A. Hately, P. Hullah, D. Martin-Marrero, E. Pastor, A. P. Rushton, and A. Volkert, "U-space concept of operations: A key enabler for opening airspace to emerging low-altitude operations," *Aerospace*, vol. 7, no. 3, p. 24, 2020.

[71] *Queens- OpenStack Releases*, 2021 (accessed May 31, 2021), https://releases.openstack.org/queens/.

[72] 3rd Generation Partnership Project, "System architecture for the 5g system; stage 2," *Technical Specification Group Services and System Aspects, 3GPP Technical Specification 23.501, version 16.2.0*, 2019.

[73] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, “5g evolution: A view on 5g cellular technology beyond 3gpp release 15,” *IEEE Access*, vol. 7, pp. 127 639–127 651, 2019.

[74] D. Soldani, Y. J. Guo, B. Barani, P. Mogensen, I. Chih-Lin, and S. K. Das, “5g for ultra-reliable low-latency communications,” *Ieee Network*, vol. 32, no. 2, pp. 6–7, 2018.

[75] Q. Pan, X. Wen, Z. Lu, W. Jing, and L. Li, “Cluster-based group paging for massive machine type communications under 5g networks,” *IEEE Access*, vol. 6, pp. 64 891–64 904, 2018.

[76] A. Hoglund, D. P. Van, T. Tirronen, O. Liberg, Y. Sui, and E. A. Yavuz, “3gpp release 15 early data transmission,” *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 90–96, 2018.

[77] T. Inoue, “5g nr release 16 and millimeter wave integrated access and backhaul,” in *2020 IEEE Radio and Wireless Symposium (RWS)*. IEEE, 2020, pp. 56–59.

[78] K. Flynn, *A Global Partnership, Release 17*, 2020 (accessed February 5, 2021), https://www.3gpp.org/release-17.

[79] I. T. Union, *UN Broadband Commission sets global broadband targets to bring online the world’s 3.8 billion not connected to the Internet*, 2020 (accessed February 5, 2021), https://www.itu.int/en/mediacentre/Pages/2018-PR01.aspx.

[80] C. Kalogiros, G. Zois, G. Darzanos, H. K. Hallingby, H. Lønsethagen, M. B. Weiss, and A. Gavras, “The potential of 5g experimentation-as-a-service paradigm for operators and vertical industries: the case of 5g-vinni facility,” in *2019 IEEE 2nd 5G World Forum (5GWF)*. IEEE, 2019, pp. 347–352.

[81] A. P. Silva, C. Tranoris, S. Denazis, S. Sargento, J. Pereira, M. Luís, R. Moreira, F. Silva, I. Vidal, B. Nogales *et al.*, “5ginfire: An end-to-end open5g vertical network function ecosystem,” *Ad Hoc Networks*, vol. 93, p. 101895, 2019.

[82] X. Li, A. Garcia-Saavedra, X. C. Perez, C. J. Bernardos, C. Guimaraes, K. Antevski, J. Mangues, J. Baranda, E. Zeydan, D. Corujo, P. Iovanna, G. Landi, J. Alonso, P. Paixao, H. Martins, M. Lorenzo, J. Ordoñez-Lucena, and D. López, “5Growth: An End-to-End Service Platform for Automated Deployment and Management of Vertical Services over 5G Networks,” *IEEE Communications Magazine*, vol. 1, no. 1, pp. 1–2, May 2021.

[83] C. Guimarães, A. d. l. Oliva Delgado, and A. Azcorra Saloña, “5g-dive: edge intelligence for vertical experimentation,” 2019.

[84] F. Moggio, M. Boldi, S. Canale, V. Suraci, C. Casetti, G. Bernini, G. Landi, and P. Giaccone, “5g eve a european platform for 5g application deployment,” *Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization*, 2020.

[85] A. Ferreira, L. Mendes, W. Dias, T. Marins, D. Gaspar, A. Matos, C. Silva, and B. Sokal, “5G-RANGE Project Field Trial,” in *2019 European Conference on Networks and Communications (EuCNC)*, 2019, pp. 490–494.

[86] W. Dias, A. Ferreira, R. Kagami, J. S. Ferreira, D. Silva, and L. Mendes, “5g-range: A transceiver for remote areas based on software-defined radio,” in *2020 European Conference on Networks and Communications (EuCNC)*. IEEE, 2020, pp. 100–104.

[87] X. Yan, S. Qijun, Z. Hongshun, and S. Lulu, “Dynamic spectrum allocation based on cognitive radio,” in *2009 5th Asia-Pacific Conference on Environmental Electromagnetics*. IEEE, 2009, pp. 254–257.

[88] G. Deepak, K. Navaie, and Q. Ni, “Radio resource allocation in collaborative cognitive radio networks based on primary sensing profile,” *IEEE Access*, vol. 6, pp. 50 344–50 357, 2018.

[89] F. Akhoundi, S. Sharifi-Malvajerdi, O. Poursaeed, and J. A. Salehi, “Analytical studies of fragmented-spectrum multi-level ofdm-cdma technique in cognitive radio networks,” in *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2016, pp. 1–6.

[90] M. Hu, J. Li, and Y. Lv, “A comparative study of polar code decoding algorithms,” in *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE, 2017, pp. 1221–1225.

[91] H.-H. Tu, C.-W. Lee, and I.-W. Lai, “Low-complexity maximum likelihood (ml) decoder for space-time block coded spatial permutation modulation (stbc-spm),” in *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. IEEE, 2019, pp. 1–2.

[92] A. A. Kalachikov and N. S. Shelkunov, “Performance evaluation of the detection algorithms for mimo spatial multiplexing based on analytical wireless mimo channel models,” in *2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*. IEEE, 2018, pp. 180–183.

[93] J. Vartiainen, H. Karvonen, M. Matinmikko-Blue, L. Mendes, H. Saarnisaari, and A. Matos, “Energy detection based spectrum sensing for rural area networks,” *EAI Endorsed Transactions on Wireless Spectrum*, vol. 4, no. 13, 2020.

[94] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, “Rfc2784: Generic routing encapsulation (gre),” 2000.

[95] P. Hoffman, “Cryptographic suites for ipsec,” *RFC 4308 (Proposed Standard)*, 2005.

[96] C. Wu and E. Vyncke, *Resolve IPv4 Fragmentation, MTU, MSS, and PMTUD Issues with GRE and IPsec*, 2019 (accessed July 7, 2020), https://www.cisco.com/c/en/us/support/docs/ip/generic-routing-encapsulation-gre/25885-pmtud-ipfrag.html.

[97] X. Xiao, W. Wang, T. Chen, Y. Cao, T. Jiang, and Q. Zhang, “Sensor-augmented neural adaptive bitrate video streaming on uavs,” *IEEE Transactions on Multimedia*, vol. 22, no. 6, pp. 1567–1576, 2020.

[98] “Unmanned aircraft system traffic management (utm),” Jun 2020. [Online]. Available: https://www.faa.gov/uas/research_development/traffic_management/

[99] L. Gipson, “Uas traffic management (utm) project,” Aug 2018. [Online]. Available: https://www.nasa.gov/aeroresearch/programs/aosp/utm

[100] M. Huttunen, “The u-space concept,” *Reprinted from Air & Space Law*, vol. 44, no. 1, pp. 69–89, 2019.

[101] V. Lappas, G. Zoumponos, V. Kostopoulos, H. Shin, A. Tsourdos, M. Tantarini, D. Shmoko, J. Munoz, N. Amoratis, A. Maragkakis *et al.*, “Eurodrone, a european utm testbed for u-space,” in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1766–1774.

[102] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, “A tutorial on uavs for wireless networks: Applications, challenges, and open problems,” *IEEE communications surveys & tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[103] D. Mishra and E. Natalizio, “A survey on cellular-connected uavs: Design challenges, enabling 5g/b5g innovations, and experimental advancements,” *Computer Networks*, vol. 182, p. 107451, 2020.

[104] D. Fuller, “At&t detail network testing of drones in football stadiums,” Sep 2016. [Online]. Available: https://www.androidheadlines.com/2016/09/att-detail-network-testing-of-drones-in-football-stadiums.html

[105] I. Bor-Yaliniz and H. Yanikomeroglu, “The new frontier in ran heterogeneity: Multi-tier drone-cells,” *IEEE Communications Magazine*, vol. 54, no. 11, pp. 48–55, 2016.

[106] B. Nogales, M. Silva, I. Vidal, M. Luís, F. Valera, S. Sargento, and A. Azcorra, “Using aerial and vehicular nfv infrastructures to agilely

create vertical services," *Sensors*, vol. 21, no. 4, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/4/1342

[107] J. Lyu, Y. Zeng, and R. Zhang, "Uav-aided offloading for cellular hotspot," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3988–4001, 2018.

[108] G. Baldini, S. Karanasios, D. Allen, and F. Vergari, "Survey of wireless communication technologies for public safety," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 619–641, 2013.

[109] A. Merwaday and I. Guvenc, "Uav assisted heterogeneous networks for public safety communications," in *2015 IEEE wireless communications and networking conference workshops (WCNCW)*. IEEE, 2015, pp. 329–334.

[110] A. Albanese, V. Sciancalepore, and X. Costa-Pérez, "Sardo: An automated search-and-rescue drone-based solution for victims localization," *arXiv preprint arXiv:2003.05819*, 2020.

[111] S. D. Muruganathan, X. Lin, H.-L. Maattanen, J. Sedin, Z. Zou, W. A. Hapsari, and S. Yasukawa, "An overview of 3gpp release-15 study on enhanced lte support for connected drones," *arXiv preprint arXiv:1805.00826*, 2018.

[112] H. Ullah, N. G. Nair, A. Moore, C. Nugent, P. Muschamp, and M. Cuevas, "5g communication: an overview of vehicle-to-everything, drones, and healthcare use-cases," *IEEE Access*, vol. 7, pp. 37 251–37 268, 2019.

[113] M. Mozaffari, A. T. Z. Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5g with uavs: Foundations of a 3d wireless cellular network," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 357–372, 2018.

[114] I. Kovacs, R. Amorim, H. C. Nguyen, J. Wigard, and P. Mogensen, "Interference analysis for uav connectivity over lte using aerial radio measurements," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2017, pp. 1–6.

[115] A. Mukherjee, V. Keshary, K. Pandya, N. Dey, and S. C. Satapathy, "Flying ad hoc networks: A comprehensive survey," *Information and decision sciences*, pp. 569–580, 2018.

[116] A. V. Leonov, "Application of bee colony algorithm for fanet routing," in *2016 17th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)*. IEEE, 2016, pp. 124–132.

[117] A. Bujari, C. E. Palazzi, and D. Ronzani, "A comparison of stateless position-based packet routing algorithms for fanets," *IEEE Transactions on Mobile Computing*, vol. 17, no. 11, pp. 2468–2482, 2018.

[118] M. Quaritsch, E. Stojanovski, C. Bettstetter, G. Friedrich, H. Hellwagner, B. Rinner, M. Hofbaur, and M. Shah, "Collaborative microdrones: applications and research challenges," in *Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems*, 2008, pp. 1–7.

[119] K. R. L. J. C. Branco, J. M. Pelizzoni, L. O. Neris, O. Trindade, F. S. Osório, and D. F. Wolf, "Tiriba - a new approach of uav based on model driven development and multiprocessors," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.

[120] C. Rametta and G. Schembra, "Designing a softwarized network deployed on a fleet of drones for rural zone monitoring," *Future Internet*, vol. 9, no. 1, 2017. [Online]. Available: https://www.mdpi.com/1999-5903/9/1/8

[121] M. Sara, I. Jawhar, and M. Nader, "A softwarization architecture for uavs and wsns as part of the cloud environment," in *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*, 2016, pp. 13–18.

[122] K. Lu, J. Xie, Y. Wan, and S. Fu, "Toward uav-based airborne computing," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 172–179, 2019.

[123] T. Kämäräinen, Y. Shan, M. Siekkinen, and A. Ylä-Jääski, "Virtual machines vs. containers in cloud gaming systems," in *2015 International Workshop on Network and Systems Support for Games (NetGames)*, 2015, pp. 1–6.

[124] R. Dua, A. R. Raja, and D. Kakadia, "Virtualization vs containerization to support paas," in *2014 IEEE International Conference on Cloud Engineering*, 2014, pp. 610–614.

[125] B. Burns, J. Beda, and K. Hightower, *Kubernetes.* Dpunkt, 2018.

[126] C. Hong and D. Shi, "A control system architecture with cloud platform for multi-uav surveillance," in *2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2018, pp. 1095–1097.

[127] D. Behnke, M. Müller, P. B. Bök, S. Schneider, M. Peuster, H. Karl, A. Rocha, M. Mesquita, and J. Bonnet, "Nfv-driven intrusion detection for smart manufacturing," in *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2019, pp. 1–6.

[128] A. Atutxa, J. Astorga, M. Huarte, E. Jacob, and J. Unzilla, "Enhancing rescue operations with virtualized mobile multimedia services in scarce resource devices," *IEEE Access*, vol. 8, pp. 216 029–216 042, 2020.

[129] Y. Hao, "A scalable cloud for internet of things in smart cities," *Journal of Computers*, vol. 26, no. 3, pp. 1–13, 2015.

[130] G. Pardo-Castellote, "Omg data-distribution service: Architectural overview," in *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.* IEEE, 2003, pp. 200–206.

[131] J. P. Ribeiro, H. Fontes, M. Lopes, H. Silva, R. Campos, J. M. Almeida, and E. Silva, "Uav cooperative perception based on dds communications network," in *OCEANS 2017-Anchorage.* IEEE, 2017, pp. 1–8.

[132] H.-S. Park, J.-H. Jang, J.-D. Kim, H.-I. Jung, and S.-H. Lee, "Bandwidth-aware dds communication in wlan," in *2012 International Conference on Systems and Informatics (ICSAI2012).* IEEE, 2012, pp. 1542–1546.

[133] G. Rigazzi, J.-P. Kainulainen, C. Turyagyenda, A. Mourad, and J. Ahn, "An edge and fog computing platform for effective deployment of 360 video applications," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW).* IEEE, 2019, pp. 1–6.

[134] D. Maltese and J. Farjon, "Detect and avoid function for uavs: presentation of an eo/ir sensor solution," in *Infrared Technology and Applications XLVI*, vol. 11407. International Society for Optics and Photonics, 2020, p. 1140713.

[135] P. Krishnan and K. Manimala, "Implementation of optimized dynamic trajectory modification algorithm to avoid obstacles for secure navigation of uav," *Applied Soft Computing*, vol. 90, p. 106168, 2020.

[136] C. Allignol, N. Barnier, N. Durand, and E. Blond, "Detect & avoid, uav integration in the lower airspace traffic," in *7th International Conference on Research in Air Transportation (ICRAT), Philadelphia*, 2016.

[137] S. Atoev, K.-R. Kwon, S.-H. Lee, and K.-S. Moon, "Data analysis of the mavlink communication protocol," in *2017 International Conference on Information Science and Communications Technologies (ICISCT).* IEEE, 2017, pp. 1–3.

[138] A. Koubâa *et al.*, *Robot Operating System (ROS).* Springer, 2017, vol. 1.

[139] G. T. . V0.3.1, "Technical specification group radio access network: Study on enhanced lte support for aerial vehicles (release 15)," 2018.

[140] 5TONIC, *The 5TONIC Laboratory*, 2020 (accessed February 5, 2021), https://www.5tonic.org.

[141] E. Dahlman, G. Mildh, S. Parkvall, J. Peisa, J. Sachs, and Y. Selén, “5g radio access,” *Ericsson review*, vol. 6, no. 1, 2014.

[142] X. Hu, B. Pang, F. Dai, and K. H. Low, “Risk assessment model for uav cost-effective path planning in urban environments,” *IEEE Access*, vol. 8, pp. 150 162–150 173, 2020.

[143] A. Geramifard, J. Redding, N. Roy, and J. P. How, “Uav cooperative control with stochastic risk models,” in *Proceedings of the 2011 American Control Conference.* IEEE, 2011, pp. 3393–3398.

[144] “Multiple bitrate streaming with ibm watson media.” [Online]. Available: https://support.video.ibm.com/hc/en-us/articles/207852127-Multiple-bitrate-streaming-with-IBM-Watson-Media

[145] S. Hayat, E. Yanmaz, and R. Muzaffar, “Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.

[146] C. Grasso and G. Schembra, “A fleet of mec uavs to extend a 5g network slice for video monitoring with low-latency constraints,” *Journal of Sensor and Actuator Networks*, vol. 8, no. 1, p. 3, 2019.

[147] M. Moradi, K. Sundaresan, E. Chai, S. Rangarajan, and Z. M. Mao, “Skycore: Moving core to the edge for untethered and reliable uav-based lte networks,” in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 35–49.

[148] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[149] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, “Flying ad-hoc networks (fanets): A survey,” *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013.

[150] J. Sánchez-García, J. García-Campos, M. Arzamendia, D. G. Reina, S. Toral, and D. Gregor, “A survey on unmanned aerial and aquatic vehicle multi-hop networks: Wireless communications, evaluation tools and applications,” *Computer Communications*, vol. 119, pp. 43–65, 2018.

[151] M. A. Khan, I. M. Qureshi, and F. Khanzada, “A hybrid communication scheme for efficient and low-cost deployment of future flying ad-hoc network (fanet),” *Drones*, vol. 3, no. 1, p. 16, 2019.

[152] M. Erdelj, O. Saif, E. Natalizio, and I. Fantoni, “Uavs that fly forever: Uninterrupted structural inspection through automatic uav replacement,” *Ad Hoc Networks*, vol. 94, p. 101612, 2019.

[153] O. Burdakov, J. Kvarnström, and P. Doherty, “Optimal scheduling for replacing perimeter guarding unmanned aerial vehicles,” *Annals of Operations Research*, vol. 249, no. 1-2, pp. 163–174, 2017.

[154] M. L. Fravolini, A. Ficola, G. Campa, M. R. Napolitano, and B. Seanor, “Modeling and control issues for autonomous aerial refueling for uavs using a probe–drogue refueling system,” *Aerospace science and technology*, vol. 8, no. 7, pp. 611–618, 2004.

[155] K. Fujii, K. Higuchi, and J. Rekimoto, “Endless flyer: a continuous flying drone with automatic battery replacement,” in *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing.* IEEE, 2013, pp. 216–223.

[156] N. K. Ure, G. Chowdhary, T. Toksoz, J. P. How, M. A. Vavrina, and J. Vian, “An automated battery management system to enable persistent missions with multiple aerial vehicles,” *IEEE/ASME transactions on mechatronics*, vol. 20, no. 1, pp. 275–286, 2014.

[157] K. A. Suzuki, P. Kemper Filho, and J. R. Morrison, “Automatic battery replacement system for uavs: Analysis and design,” *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 563–586, 2012.

[158] M. Lu, M. Bagheri, A. P. James, and T. Phung, “Wireless charging techniques for uavs: A review, reconceptualization, and extension,” *IEEE Access*, vol. 6, pp. 29 865–29 884, 2018.

[159] T. Chang and H. Yu, “Improving electric powered uavs’ endurance by incorporating battery dumping concept,” *Procedia Engineering*, vol. 99, pp. 168–179, 2015.

[160] T. J. Nugent Jr and J. T. Kare, “Laser power beaming for defense and security applications,” in *Unmanned systems technology XIII*, vol. 8045. International Society for Optics and Photonics, 2011, p. 804514.

[161] K. J. Duncan, “Laser based power transmission: Component selection and laser hazard analysis,” in *2016 IEEE PELS Workshop on Emerging Technologies: Wireless Power Transfer (WoW).* IEEE, 2016, pp. 100–103.

[162] B. Galkin, J. Kibilda, and L. A. DaSilva, “Uavs as mobile infrastructure: Addressing battery lifetime,” *IEEE Communications Magazine*, vol. 57, no. 6, pp. 132–137, 2019.

[163] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Wireless communication using unmanned aerial vehicles (uavs): Optimal transport theory for hover time optimization,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8052–8066, 2017.

[164] V. Sharma, K. Srinivasan, H.-C. Chao, K.-L. Hua, and W.-H. Cheng, “Intelligent deployment of uavs in 5g heterogeneous communication environment for improved coverage,” *Journal of Network and Computer Applications*, vol. 85, pp. 94–105, 2017.

[165] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, “Placement optimization of uav-mounted mobile base stations,” *IEEE Communications Letters*, vol. 21, no. 3, pp. 604–607, 2016.

[166] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, “3-d placement of an unmanned aerial vehicle base station (uav-bs) for energy-efficient maximal coverage,” *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 434–437, 2017.

[167] L. Ruan, J. Wang, J. Chen, Y. Xu, Y. Yang, H. Jiang, Y. Zhang, and Y. Xu, “Energy-efficient multi-uav coverage deployment in uav networks: A game-theoretic framework,” *China Communications*, vol. 15, no. 10, pp. 194–209, 2018.

[168] D. Reina, H. Tawfik, and S. Toral, “Multi-subpopulation evolutionary algorithms for coverage deployment of uav-networks,” *Ad Hoc Networks*, vol. 68, pp. 16–32, 2018.

[169] D. Reina, T. Camp, A. Munjal, and S. Toral, “Evolutionary deployment and local search-based movements of 0th responders in disaster scenarios,” *Future Generation Computer Systems*, vol. 88, pp. 61–78, 2018.

[170] F. Morbidi, R. Cano, and D. Lara, “Minimum-energy path generation for a quadrotor uav,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1492–1498.

[171] S. Martello and D. Vigo, “Exact solution of the two-dimensional finite bin packing problem,” *Management science*, vol. 44, no. 3, pp. 388–399, 1998.

[172] Y. Zeng and R. Zhang, “Energy-efficient uav communication with trajectory optimization,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

[173] M. R. Garey and D. S. Johnson, “A guide to the theory of np-completeness,” *Computers and Intractability*, pp. 37–79, 1990.

[174] V. Sanchez-Aguero, C. Tipantuña, and F. Valera, *Energy-aware management strategies in Multi-UAV fleets repository*, 2020 (accessed March 25, 2020), https://github.com/vsaguero/energyUAV.

[175] V. Alarcón, M. García, F. Alarcón, A. Viguria, Á. Martínez, D. Janisch, J. J. Acevedo, I. Maza, and A. Ollero, "Procedures for the integration of drones into the airspace based on u-space services," *Aerospace*, vol. 7, no. 9, p. 128, 2020.

[176] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. Robinson, "Unmanned aircraft system traffic management (utm) concept of operations," in *16th AIAA Aviation Technology, Integration, and Operations Conference.* AIAA, 2016, pp. 1–16.

[177] T. Bolić and P. Ravenhill, "Sesar: The past, present, and future of european air traffic management research," *Engineering*, vol. 7, no. 4, pp. 448–451, 2021.

[178] B. López, J. Muñoz, F. Quevedo, C. A. Monje, S. Garrido, and L. E. Moreno, "Path planning and collision risk management strategy for multi-uav systems in 3d environments," *Sensors*, vol. 21, no. 13, p. 4414, 2021.

[179] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology," RFC 7426, Jan. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7426.txt

[180] R. D. Johnson, "Unmanned aircraft system traffic management (utm) project," in *Airspace Operations and Safety Program (AOSP) Research and Development Partnership Workshop*, no. ARC-E-DAA-TN55386, 2018.

[181] H. Baek and J. Lim, "Design of future uav-relay tactical data link for reliable uav control and situational awareness," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 144–150, 2018.

[182] Y.-T. Kuo, H.-Y. Wei *et al.*, "Reliable two-hop device-to-device communications for uavs," in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS).* IEEE, 2019, pp. 1–5.

[183] Z. Yuan, J. Jin, L. Sun, K.-W. Chin, and G.-M. Muntean, "Ultra-reliable iot communications with uavs: A swarm use case," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 90–96, 2018.

[184] H. C. de Resende, J. F. N. Pinheiro, P. Reiter, C. B. Both, and J. M. Marquez-Barja, "4g/5g performance of a multi-rat uav for medical parcel delivery," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC).* IEEE, 2022, pp. 268–271.

[185] O. S. Oubbati, M. Atiquzzaman, T. A. Ahanger, and A. Ibrahim, “Softwarization of uav networks: A survey of applications and future trends,” *IEEE Access*, vol. 8, pp. 98 073–98 125, 2020.

[186] J. McCoy and D. B. Rawat, “Software-defined networking for unmanned aerial vehicular networking and security: A survey,” *Electronics*, vol. 8, no. 12, p. 1468, 2019.

[187] F. Xiong, A. Li, H. Wang, and L. Tang, “An sdn-mqtt based communication system for battlefield uav swarms,” *IEEE Communications Magazine*, vol. 57, no. 8, pp. 41–47, 2019.

[188] X. Zhang, H. Wang, and H. Zhao, “An sdn framework for uav backbone network towards knowledge centric networking,” in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS).* IEEE, 2018, pp. 456–461.

[189] S. ur Rahman, G.-H. Kim, Y.-Z. Cho, and A. Khan, “Deployment of an sdn-based uav network: Controller placement and tradeoff between control overhead and delay,” in *2017 International Conference on Information and Communication Technology Convergence (ICTC).* IEEE, 2017, pp. 1290–1292.

[190] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, “Optimized link state routing protocol (olsr),” 2003.

[191] C. Y. Hans, G. Quer, and R. R. Rao, “Wireless sdn mobile ad hoc network: From theory to practice,” in *2017 IEEE International Conference on Communications (ICC).* IEEE, 2017, pp. 1–7.

[192] “Labyrinth,” 2022. [Online]. Available: https://github.com/Networks-it-uc3m/JoVe_labyrinth

[193] F. Solensky, “Considerations for internet group management protocol (igmp) and multicast listener discovery (mld) snooping switches,” *RFC 4541*, 2006.