



**Politecnico
di Torino**

POLITECNICO DI TORINO

Master Course in Mechatronic Engineering
Academic Year 2021/2022
Graduation session December 2022

Master Thesis

Cellular Networks Support for Real-Time Control of UAVs

Supervisor

Prof. Alessandro Rizzo

Candidate

Pietro Brach del Prever

Student ID: 286120

Co-supervisor

Prof. Tommaso Melodia

Abstract

This thesis aims at studying the feasibility of using UAVs in BVLOS applications and focuses on the cellular communication support for the UAVs. The communication between GCS and UAVs is studied in two different scenarios, a real-world one and an emulated one. Showing that the two systems have similar behavior, despite presenting different performances, enables the use of hardware-in-the-loop emulators to develop new features to improve the cellular support for UAVs and comply with the 3GPP standards, and then to prototype solutions for cellular-supported UAVs in BVLOS flight.

The study has been conducted by collecting over-the-air measurements of the real system, as well as by collecting data of the emulated system within a hardware-in-the-loop emulator. For the purposes of this study, the massive channel emulator *Colosseum*, available at Northeastern facilities, was used.

The results obtained about the performances of the emulated and real systems and about the similarities of their behavior are satisfactory and match the expected outcomes. The two systems, the real-world one and the emulated one, show similar behaviors, despite having significant differences in terms of performances. This allows to consider the emulator (Colosseum) as a proper testbed for the design and development of new applications and features to improve the performance of cellular support for UAVs and thus meet the requirements of the 3GPP for enhanced LTE support for the UAS Traffic Management (UTM), while the real-world system can be used in a later phase.

Acknowledgements

I am extremely grateful to my Thesis Supervisor Professor Alessandro Rizzo of Politecnico di Torino and to my Thesis Co-supervisor Professor Tommaso Melodia of Northeastern University for allowing me to do research and write my thesis in Boston, and for the help in structuring and correcting this work: without them, this project would not have been possible. I would like to extend my sincere thanks to Research Assistant Professor Salvatore D'Oro for his kind guidance and continuous help during my months in Boston and after my return to Italy. I am also thankful to my colleagues at the WiNES Laboratory of the Institute for the Wireless Internet of Things at Northeastern University for their frequent help and useful advice. Everyone steered me in the right direction and offered help and support whenever I needed it.

In addition, I would like to thank my family and friends for supporting me during my years of university and specifically during the period of my thesis. I have always received constant support and love, even on the hardest days, from my mother Gianna, my father Massimiliano, and my sister Teresa. A special thank goes to my dear aunt Elena for her precious advice and the continuous encouragement that pushed me to always give my best.

Firenze, November 2022

Pietro Brach del Prever

Contents

1	Introduction	1
1.1	Research Contributions	2
1.2	Motivations	3
1.3	Structure of the Thesis	4
2	Related Work	6
3	Objectives	9
4	Materials and Methods	11
4.1	Hardware and Software Description	11
4.1.1	UAS	11
4.1.2	Colosseum Virtual Emulator	13
4.1.3	Colosseum Scenario	14
4.1.4	USRP Management	15
4.1.5	Traffic Analysis and Generation	16
4.1.6	Devices Synchronization via <i>Network Time Protocol</i> (NTP)	16
4.1.7	Data Analysis	17
4.2	Methods and Experimental Procedure	17
4.2.1	Traffic Analysis and Traffic Model Definition	17
4.2.2	Design of Experiments	20
4.2.3	Real-World System Setup and Experimental Procedure	20
4.2.4	Emulated System Setup and Experimental Procedure	21
5	Results	24
5.1	Data Cleaning and Preparation	24
5.2	Data Representation and Analysis Tools	25
5.2.1	Performance Indicators Covariances	26
5.2.2	CQI and MCS	30
5.2.3	Performance Indicators Trends	33
5.2.4	Round-Trip Time Plots	35

6 Discussion	39
7 Conclusion	42
7.1 Future Work	43
A Glossary and Acronyms	45

Chapter 1

Introduction

Drones are a technology that is already widely adopted in the civil and military domains, but it is gaining more and more interest both in academic and industrial communities because of the increasing number of their applications and the possibility to deploy them in autonomous missions. The commercial purposes for which drones are used include earth and atmospheric observations, civil and commercial services (e.g. delivery, vigilance, and facilities monitoring), disaster management, security, and warfare. Drones can be used alone, but also in combination with other emerging technologies and paradigms¹ [1].

Autonomous drones, termed Unmanned Aerial Vehicles (UAVs), are part of a wider system called Unmanned Aerial System (UAS), which includes one or more UAVs, a Ground Control Station (GCS) from which the UAVs are controlled and monitored, and a communication link. The legal and normative frame for these systems will be the UAS Traffic Management (UTM), which is a project still in a development phase that is supposed to regulate and support multiple Beyond Visual-Line-Of-Sight (BVLOS) drone operations in the near future, working in a separate but complementary way with FAA's Air Traffic Management (ATM) [2].

UAVs represent an attractive way of exploiting the advantages of the robotic and drone industry in combination with the continuously advancing field of AI and autonomous algorithms. Many technology domains are affected by the advances in drone technologies, and many domains also contribute to it. Wireless communication is an essential aspect of the correct functioning of UAVs since most applications and functions for which autonomous UAVs are designed rely on the availability of Ultra-Reliable and Low-Latency Communications (URLLC). Specifically, cellular communication has the potential to satisfy such requirements, and

¹For instance, drones can easily be integrated in IoT platforms, and they are used in combination with AR and VR for virtual real estate tours.

that is why cellular support for UAS has been extensively studied. In general, there are three main opportunities for UAV technologies and cellular technology integration [1]:

1. *UAV-Assisted Cellular Communication*: UAVs as flying Base Stations (BS);
2. *Cellular-Assisted UAV Communication (cellular-connected UAVs)*: UAVs as aerial UEs;
3. *UAV-UAV Communication*: UAVs directly communicating with each other for cooperation and collision avoidance.

To guarantee the safety and real-time response of the UAVs, it is of the utmost importance to define strict cellular communication requirements in terms of latency, throughput and packet loss rate, as done by the 3GPP (Third Generation Partnership Project) in [3]. Considering the applications and possibilities of integration of the UAV and cellular technologies mentioned before, it is clear that guaranteeing URLLC is essential in each scenario. The nature of the new flying platform generates new challenges whose solutions are not straightforward. The requirements expressed by the 3GPP concern the latency, data rate, and packet loss rate. Such requirements are not yet met, but a lot of researches are being conducted in this regard.

It is clear that building an URLLC is relevant for all the domains mentioned before: once the requirements are met complete autonomy in operations will be possible. For instance, a network of flying UAVs working as Base Stations could be deployed in complete autonomy in emergency scenarios where regular communication links have failed and communication must be re-established for emergency response purposes. Another example include the use of UAVs for autonomous package delivery or autonomous facility monitoring, for which the drones could fly and complete tasks without any human assistance. Furthermore, this research will enable a new development framework and pipeline for the implementation of new functionalities requiring URLLC on the cellular network: this will ease the innovation process of technologies related to cellular network support fro autonomous drones, by making it easier to create innovative solutions or improve existing ones. For instance, if the

1.1 Research Contributions

This work contributes to the field addressing the three fundamental questions that follow.

1. *How can we model the main features of the traffic of a real-world system so that the network performances can be studied indifferently on both a real-world*

system and an emulated one? A pipeline to model the traffic of a real-world system, validate such model, and reproduce it on both real and emulated systems is devised.

2. *What are the differences, in terms of network behavior and performance, between the real-world and emulated systems?* To tackle this question, the network performance parameters of interest according to the 3GPP [3] are taken into account and it is shown that, despite a difference in performance, the communication link in the two scenarios has indeed similar behavior.
3. *Is it possible to study, develop, and test complex solutions for BVLOS UAV applications on the emulated system first, and only then deploy and test them on the real system?* Given what previously answered to the other questions, a framework that includes several development platforms and steps is envisioned.

Colosseum will be used to emulate the investigated system: Colosseum is a massive wireless channel emulator and traffic generator born to support DARPA's Collaborative Spectrum Challenge, which is now owned, managed, and made available by Northeastern University [4].

1.2 Motivations

The third point listed in Section 1.1 represents an important contribution that will be further analyzed in future works and represent the real motivation for this work, which fits in the research paths for the wireless communication support for UAVs. The underlying idea is to simplify and accelerate the design process for new applications and features for the cellular network stack, while keeping the 3GPP requirements always in mind. The envisioned framework will therefore allow researchers and developers to study systems and design solutions in a quicker and more effective way. Experiments and app development will start and first be carried out in an emulated environment, which is much more flexible and easier to manage than a real world system that consists of one or more Base Stations and one or more flying UAVs, requires specific setup, and is prone to face a lot of issues not directly related to the research topic. For instance, one important limitation of experimenting with real drones is the limited flying time of small- and medium-sized UAVs, whose nominal flying time is generally around a few tens of minutes and never exceeds an hour. However, using AI and ML for research requires huge amounts of data, which takes a lot of time to be collected, and thus requires drone to fly for a long time; however drone flight time is subject to battery capacity, as well as flight conditions and flight style, and drones require often stops to recharge batteries. On the other hand, an emulated system does

not require downtime for battery recharge. Furthermore, the emulated system also avoids all the issues related to flight safety, weather conditions, correct functioning of the robotic system, and untracked or unexpected sources of disturbance and interference. Last but not least, an emulated system allows to reproduce the same conditions for the experiments over and over again, while a real-world scenario is stochastic and the same conditions are not likely to repeat in two different experiments.

The purpose is then to have a series of platforms related to different levels of virtualization and abstraction of the system under scope, and to be able to focus only on the new complexities and challenges introduced at each level. According to the envisioned development approach, the experiments would proceed according to the following steps:

1. develop the applications in a software based system, i.e. a channel emulator (such as Colosseum, which virtually requires only an access to the Internet²);
2. validate the solution on an over-the-air testbed (such as Arena[5]);
3. validate the solution on real drones in over-the-air experiments in a controlled environment (such as the anechoic chamber and the drone cage in the Burlington Campus of Northeastern University [6]);
4. validate the solution in an over-the-air long-range structured environment (as it could take place in AERPAW—Aerial Experimentation and Research Platform for Advanced Wireless³).

1.3 Structure of the Thesis

The thesis is organized in 7 chapters, including this Introduction, and one appendix.

Chapter 2 presents the academic work related to the research presented here.

Chapter 3 expresses the objectives of this research.

Chapter 4 describes the tools, methods, procedures, and setups adopted to conduct the study.

²Access must be granted through a VPN to authorized researchers by the Colosseum managing team.

³AERPAW is a ‘wireless research platform to study the convergence of 5G technology and autonomous drones’[7].

Chapter 5 presents the collected data by means of informative representations and provide descriptions of the results.

Chapter 6 offers an analysis of the results with the respect to the expected outcomes and the objectives of the study.

Chapter 7 consists of the conclusion and a section about the future work.

Appendix presents the explanation of the technical terms and of the acronyms.

Chapter 2

Related Work

3GPP has provided extended studies and specifications on cellular network (especially 4G LTE and 5G) support for UAVs connectivity in Releases 15, 16, 17, and 18 [8, 9, 10, 11]. The aim is twofold: ensure safe operations of UAVs, and ensure that other (terrestrial) users do not experience any loss of service due to the proximity of UASs [12]. There are two main kinds of data wirelessly transmitted that are relevant to UASs, according to the 3GPP Releases: command-and-control (C&C) data, and application data. 3GPP expresses a series of connectivity requirements, performance evaluation scenarios, and key identified challenges for LTE connectivity for UAS Traffic Management (UTM), summarized in Table 2.1 [3].

Items	Command & Control	Application Data
Data type	Telemetry, waypoint update for autonomous UAV operation, real time piloting, identity, flight authorization, navigation database update, etc.	Video (streaming), images, other sensors data, etc.
Latency	50 ms (one way from eNB to UAV)	same as terrestrial UE
DL/UL data rate	60-100 Kbps for DL/UL	50 Mbps for UL
Reliability	Up to 10^{-3} Packet Error Loss Rate	-

Table 2.1: 3GPP requirements for UAS connectivity [3].

Many papers have covered the issues and challenges related to the support

of cellular networks for UASs, making reference to the study on enhanced LTE support for low-altitude connected drones published in Release 15 [3], but also to the more general integration of 4G, 5G, and B5G cellular networking and UAV technologies.

[1] presents a survey on the integration of 5G and B5G (beyond 5G) cellular networking and UAV technologies, focusing on cellular connected UAVs. [13] studies the limitations of using 4G LTE for the management of small UAVs and the possible enhancements that can be enacted.

Given the likelihood of the drone to be in line-of-sight (LOS) with multiple eNodeBs (eNBs) while flying at a high altitude, one of the main problems identified is linked to the interference with the eNBs or the other User Equipments (UEs), both in uplink and downlink [14, 15].

[16] further explores the limits of horizontal communication distance within which the round-trip delay and the overall packet loss probability can be ensured with a required probability, and propose an algorithm to solve the non-convex problem to optimize the altitude of the UAVs to maintain a URLLC.

[15] highlights the correlation between the number of eNBs in LOS and the number of handovers. As a consequence, the frequency of handovers is related to the number of eNBs visible. Moreover, given the downward tilting of antennas, UE at higher altitudes are worse served and experience drops in RSRP [15].

[17] aims at modelling the radio channel between UAV and cellular network, specifically considering path loss exponents and shadowing, showing that path loss exponent decreases as the UAV moves up (100 m in height corresponds to approximately tens of kilometers in horizontal range).

[15] provides an experimental study of LTE-A throughput, signal-to-interference ratio, and received power levels for drones flying at different heights, proving that the performance of the wireless communication link depends on the height of the flight. 20 Mb/s in the downlink and 40Mb/s in the uplink are achieved on average at 150 m.

[18] explores through simulations on *ns-3* the circumstances where the video-streaming capacity over the 4G LTE communication network are affected, proving that variation in the channel has a direct link with the transmitting capacity of the UAV. [19] studies the video quality and latencies for UAV teleoperation over LTE using *ns-3* simulations, reaching the conclusion that the latency of the video is higher and more sensitive to mobility than that of the control traffic, and that high mobility tends to increase latency.

There are also studies aiming at using swarms of UAVs. [20] presents a review of UAV swarm communication and control architectures. [21] presents some results of using UAVs as a LTE relay, including a placement algorithm that updates the UAV position in order to maximize the throughput is also presented.

[22] present a 4G connected fixed-wing UAV.

This work has also been possible thanks to the following experimental frameworks: [23] provides drone design specifically for wireless research purposes; [24] provides a containerized platform to manage USRPs; the massive channel emulation part happened on Colosseum [4].

Chapter 3

Objectives

The goal of this thesis is to *enable the use of hardware-in-the-loop emulators to prototype solutions for cellular-supported UAVs in BVLOS flight*, which can facilitate the prototyping and development of applications and solutions to improve the performance of cellular support for UAVs and thus meet the requirements of the 3GPP for enhanced LTE support for the UAS Traffic Management (UTM). With this aim in mind, the objective of this work is to study the cellular communication between the BS and the UAVs in two different scenarios, a real-world one and an emulated one, evaluate the performance of the system in the different scenarios in terms of latency, throughput, and packet loss rate (as expressed in Table 2.1), and compare the general behavior of the system in the two scenarios.

In practice, the objectives unfold in the following tasks.

1. *Analyze the traffic model of the real communication between GCS and UAV:* a model of traffic represents the outcome of this first step.
2. *Design a simulated traffic model, thus being able to reproduce the real traffic with a traffic generator software:* the model is reproduced by means of a traffic generator application.
3. *Validate the traffic model for the cellular communication link between BS and UAVs:* it is checked that the traffic model generated by the traffic generator has the same features and behaviors of the original traffic that has been modeled.
4. *Run the traffic generator in both the real and the emulated environments and collect data about the traffic and the network performance:* the traffic exchanged in the system in the two scenarios is recorded, and useful data are extracted from the collected packets.

5. *Compare the performance and behavior of the LTE communication in the two scenarios:* the data extracted is compared by means of informative graphs and plots to highlight differences and similarities between the two scenarios.

Chapter 4

Materials and Methods

4.1 Hardware and Software Description

This section describes what tools have been used for the experiments, how the experiments were designed and performed, and the subsequent analysis performed on the collected data.

4.1.1 UAS

The drone used as UAV is not commercially available and is the outcome of the research presented in [23]. The drone model, named *Monarch*, is a custom-made quadcopter equipped with a companion computer and a USRP, visible in Figure 4.1. A scheme indicating where each component is mounted on the drone is shown in Figure 4.2. The companion computer is an Intel NUC (model NUC7i7DN),



Figure 4.1: Drone ready for arming and takeoff on a field.

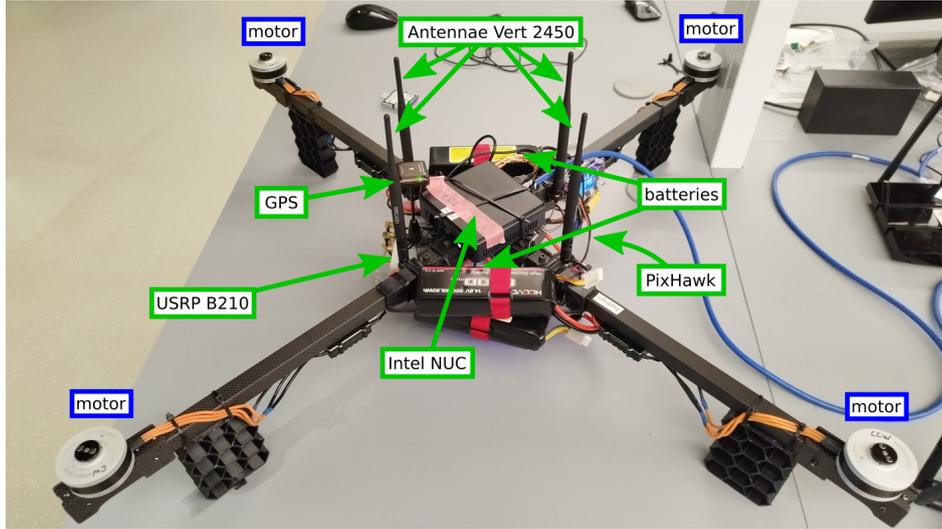


Figure 4.2: Scheme of *Monarch* drone with the position of main components.

a mini PC of 0.47 Kg featuring an Intel Core i7 CPU and a 32 GB RAM. The USRP (model B210) is a fully-programmable software radio module of 0.35 kg featuring 70 MHz - 6 GHz carrier frequency range, 56 MHz of real-time bandwidth, 2 TX and 2 RX chains with MIMO capabilities. The drone can be configured and controlled through a Ground Control Station (GCS), a computer application that allows configuring, controlling and commanding robots of different kinds, including a drone like ours. The drone can also be commanded through the RF controller, or through mission scripts written in different programming languages that use existing and available libraries to connect to the drone and control it. It is important to notice that, for the setup used, the drone has to be connected to the RF controller even if the commands are sent by the GCS or with the mission scripts: this is to ensure that, in case of an emergency, manual control can be taken and it overrides the mission uploaded to the drone. In the emulated environment in Colosseum the model of USRP used is not B210 but X310, but this does not make any difference for our purposes in the functioning or performances of the system.

The GCS, the companion computer, and the Flight Control Unit (FCU) communicate through the MAVLink protocol [25]. MAVLink is an application-layer protocol that is based on the TCP/IP protocol and, in its second version, enables reliable and low-latency communication between base station and up to 255 drones connected at the same time. MAVLink also supports the use of a companion computer thanks to the MAVLink Router software [26], to be installed on the companion computer, that forwards all the messages between the GCS and the FCU.

For the GCS, the GUI application *QGroundControl* [27] was used. The application was installed on different laptop models, all mounting recent versions of the Ubuntu operative system. The experiments and the data collections were performed with different laptops since the behavior of the whole system is not dependent on the kind of machine the GCS is mounted on.

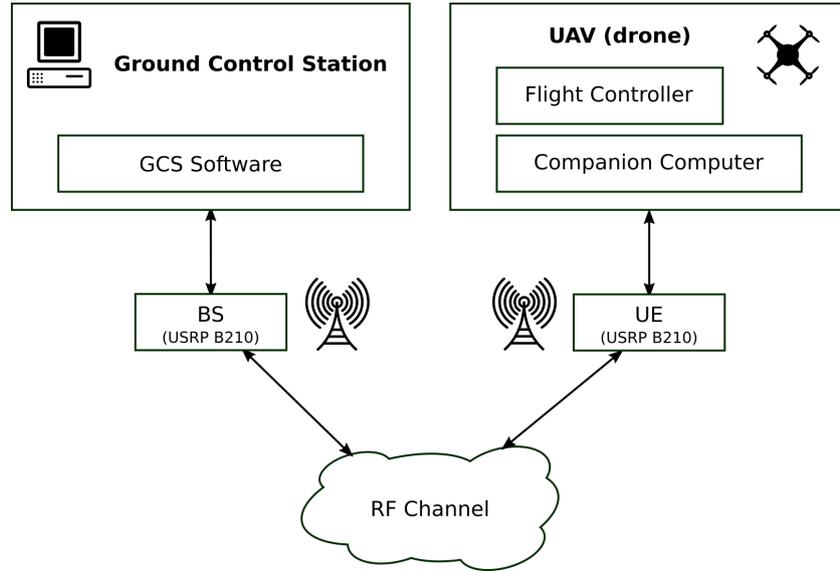


Figure 4.3: Simplified scheme of an UAS: the main components and the links between them are visible.

4.1.2 Colosseum Virtual Emulator

Colosseum consists of a Massive Channel Emulator (MCHEM) and a Traffic Generator (TGEN) [24], the latter of which is not used for the experiments described in this thesis. A picture of the Colosseum servers is shown in Figure 4.4. Figure 4.5 shows in a simplified scheme what are the main components of Colosseum, how it works, and what are the steps needed to use it. The user first reserves Colosseum resources (nodes) on a web interface accessible through a VPN, then accesses the containers and SRNs made available by the system through an `ssh` connection and performs the experiments within the chosen scenario; at the end, the collected data are downloaded.



Figure 4.4: Colosseum server in the Northeastern University facilities at the Northeastern University Burlington Campus. The various sections of the structure host the SRNs, the GPUs, and the USRPs, all of them connected through the visible wires and cables.

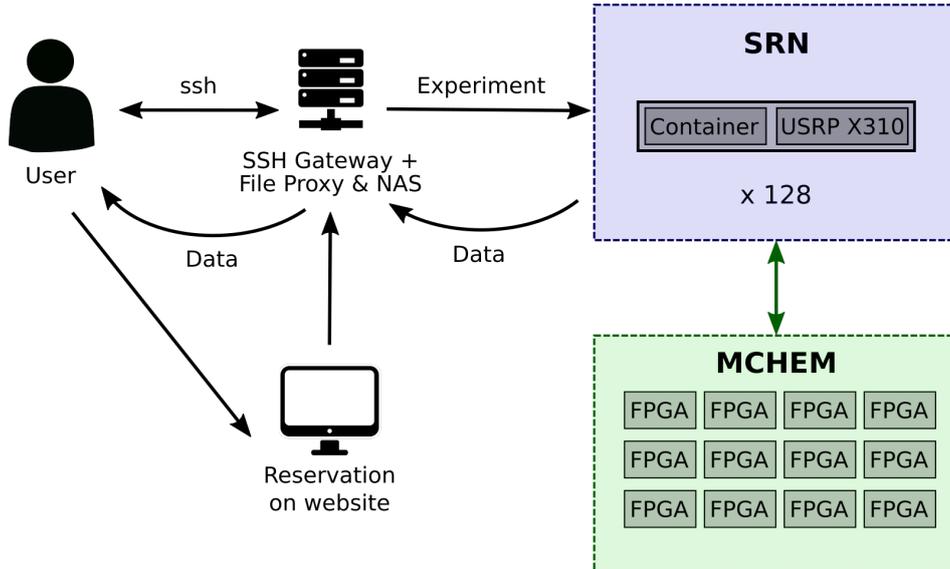
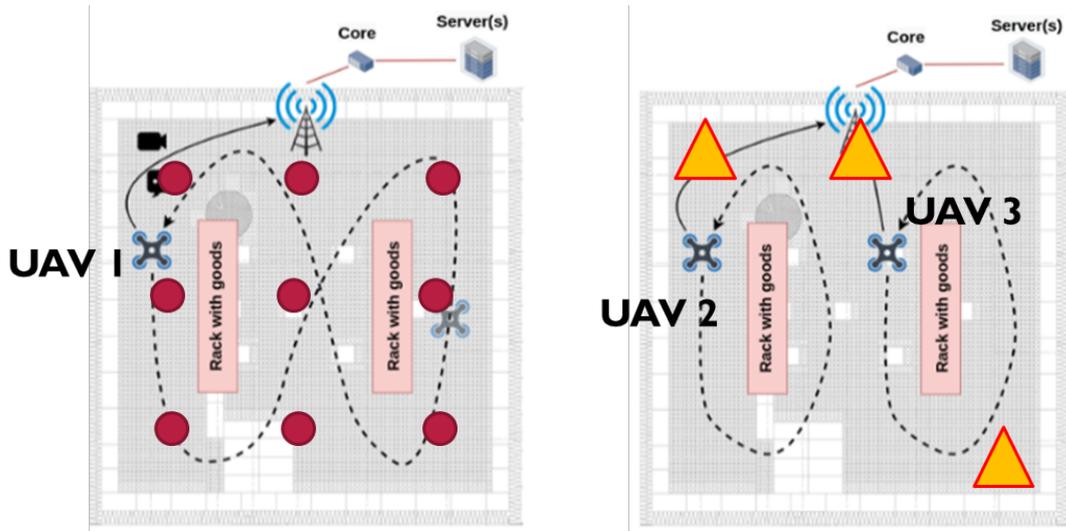


Figure 4.5: Simplified scheme of the Colosseum facility, showing the main components and the main steps to use it: once the user reserves the resources for an experiments, the initial setup of the SRNs is performed, then the experiment is run, and lastly the data collected can be downloaded. All the operations between user and the Colosseum resources happen through an `ssh` connection.

4.1.3 Colosseum Scenario

Colosseum servers allow the user to reserve a certain number of SRNs, choose one of the predefined scenarios of Colosseum reproducing a real world environment,

and then connect each of the reserved SRNs to a specific node of the scenario and start the experiments and the data collections. For the purposes of this study, only the scenario 12356 was used: it is a reproduction of the indoor environment of the anechoic chamber available at the Northeastern University Burlington Campus. Some schemes representing the scenario are shown in Figure 4.6. As the schemes show, the whole scenario includes 9 base stations (represented by the red large dots, numbered from 1 to 9, being 1 the one in the top left-hand corner, 2 the one in the top center position, and so on), 3 UAVs, and 3 additional UEs (represented by the yellow triangles). The 3 UAVs are the only SRNs that move during the simulation.



(a) The positions of the nine BS in the scenario, corresponding to the red large dots, are shown. Only one or multiple among them can be used during the experiments. The path of UAV 1 is also shown, flying along an 8-shaped trajectory inside the chamber. In the chamber, there are also two racks present, that determine, together with the walls, the reflecting surfaces for the waves and thus cause interference.

(b) The trajectories of UAVS 2 and 3 are shown here, each one flying in a cyclic trajectory around each one of the racks. The positions of three additional UEs is also shown with the yellow triangles: these represent three additional cellular users that are not autonomous UAVs that can use the spectrum and the bandwidth at the same time as the UAVs.

Figure 4.6: Schemes for the Colosseum scenario 12356.

4.1.4 USRP Management

As for the USRP, an LXD container with SCOPE [24] was used to manage the radio. SCOPE allows quick and easy configuration of the USRPs and setup of the network components: the EPC, the ENB, and the UEs.

4.1.5 Traffic Analysis and Generation

The traffic exchanged over the network between the GCS and UAV has been collected and analyzed with *tshark* and *Wireshark*, tools freely available on Linux. Both the softwares can record the packages exchanged over a certain network interface, optionally filter them according to many different parameters related to the different layers of the stack, and save the captures to *.pcapng* files. In the final phase of data analysis, some python libraries such as *pyshark* and *seaborn* have been used to analyze the traffic and represent the results.

In a first phase, the real traffic exchanged between GCS and UAV was studied and analyzed. In order to make the conditions equal for all the scenarios, both real and emulated, the traffic exchanged between GCS and UAV was captured with the help of *tshark*, the tool mentioned before, and recorded and saved into file. The available files containing information about the traffic to be reproduced can then be manipulated in order to be read by a traffic generator that can also work in the emulated environment inside Colosseum. The *DITG* software was then used [28] as traffic generator to reproduce the traffic. In our case, *DITG* reproduces a series of packets of determined size and inter-departure time, according to the values read from two *.txt* files extracted from the *tshark* captures.

4.1.6 Devices Synchronization via *Network Time Protocol* (NTP)

While performing the experiments, it is essential that the devices exchanging packets, on which *tshark* or *Wireshark* are running and record packets with their respective times, are synchronized. Colosseum servers keep all their nodes synchronized by default. However, synchronization is needed for the devices used in the real-world scenario. In this case, a local network of NTP servers and clients was created to avoid drift and offset between the devices. *NTP* stands for Network Time Protocol, which is a protocol used to synchronize clock over networks, including the Internet [29]. The devices are connected to each other via Ethernet to a local network established by a router. Among all devices, one is set to be the server, while all the others are set to be clients. The clients continuously interrogate the server for the time, compute the drift and the offset, and constantly try to reduce them. The server itself, if a connection to the Internet is available, interrogates a higher-level server and synchronizes itself to it.

To verify and evaluate the drift and offset among the computers connected to the local network both NTP and a python implementation of IEEE 1558 Precision Time Protocol (PTP) [30] were used. NTP or similar protocols are used on every computer world-wide to synchronize local time, achieving good results in terms of offset: NTP achieves offset of few milliseconds in a local network. PTP, on the

other hand, is only used in local networks when offsets of the order of microseconds or less are needed to be tracked (such as in the world of finance and high frequency trading). There exists two implementation for it: one is exclusively software-based, while the other one is hardware-based and interrogates compatible hardware devices to be even more precise. For the purposes of this thesis, the software-only based implementation of the NTP protocol were enough in terms of precision.

4.1.7 Data Analysis

After the experiments, data were handled and analyzed by means of several softwares, that include MATLAB, Wireshark, and python (and its libraries pyshark, matplotlib, and seaborn).

4.2 Methods and Experimental Procedure

This section describes the methods adopted for this research. The experimental procedure adopted proceeded according to the following steps:

1. *analysis of the traffic* exchanged between the UAV and the BS;
2. *definition of the traffic model and design of the simulated traffic* with the traffic generator;
3. *validation of the traffic model* by comparing it with the real traffic recorded in the first step;
4. *design and performing of experiments*;
5. *analysis of the results*¹.

4.2.1 Traffic Analysis and Traffic Model Definition

The first step for analyzing the traffic consisted of defining the traffic model for the communication exchanged between the GCS and the UAV. In order to study and design a model of the traffic between the GCS and the UAV, the whole real world system was prepared and configured. The hardware and software components were installed on the respective devices according to Table 4.1. The setup is also shown in Figure 4.7. With this setup, it was possible to record and save the traffic between the GCS and the UAV object of the subsequent analysis.

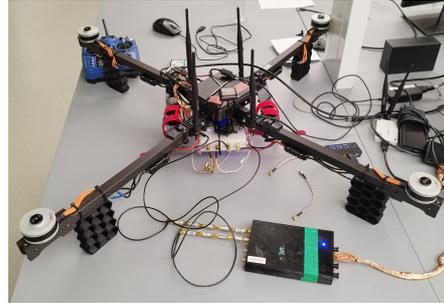
¹The analysis of the results is carried out in Chapter 5.

Device	Hardware	Software
GCS	Laptop	QGroundControl
	USRP B210	LXD + SCOPE container tshark/Wireshark
		DITG
UAV	Intel NUC	mavlink-router
	USRP B210	LXD + SCOPE container
	FCU PX4	tshark/Wireshark DITG

Table 4.1: Software configuration for the experiments in the real world environment.



(a) Setup of laptop connected to the USRP, and to a router via Ethernet cable.



(b) Setup of drone and USRP with a wired connection to the USRP working as BS.

Figure 4.7: Pictures of drones, URSPs, laptops, and setups for the first phase of the work.

Thanks to the functionalities of Wireshark and tshark, it was possible to filter the desired packets and conversations and extract specific information from them. For instance, Table 4.2a shows the size of some specific packages of the MAVLink protocol², while Table 4.2b shows the characteristics in terms of average size and average frequency of packets sent overall and in each of the two directions. It is evident from Table 4.2b that most of the traffic goes from UAV to GCS, in terms of number of packets per second and of bandwidth.

The traffic has been differentiated into two main flows according to the different

²This information was not readily available from the MAVLink library documentation.

Message type	Packet Length [Bytes]	Command Name
Arm/Disarm Command	110	COMMAND_LONG: MAV_CMD_COMPONENT_ARM_DISARM
Arm/Disarm ACK	88	COMMAND_ACK: MAV_CMD_COMPONENT_ARM_DISARM
Takeoff Command	110	COMMAND_LONG: MAV_CMD_NAV_TAKEOFF
Takeoff ACK	88	COMMAND_ACK: MAV_CMD_NAV_TAKEOFF
Set FLight Mode	84	SET_MODE
Set Flight Mode ACK	88	COMMAND_ACK: MAV_CMD_DO_SET_MODE
	82	MISSION_COUNT
	115	MISSION_ITEM_INT
Mission Upload	[...]	[...]
	115	MISSION_ITEM_INT
	109	MISSION_COUNT PING
	82	MISSION_COUNT

(a) Packet size and description for some MAVLink messages. Note: the Mission Upload consists of a series of messages that include also a number of *MISSION_ITEM_INT* equal to the number of the waypoints, including takeoff and landing points.

Stream direction	Packets per second [pps]	Packet Size [Bytes]
Both directions	448.1	115.7
From GCS	91.2	66.8
From UAV	355.9	127.8

(b) These characteristics are provided by Wireshark.

Table 4.2: Real traffic analysis: some message examples, and traffic flows characteristics.

characteristics of the packets sent: on one hand, the two devices continuously exchange small-size high-frequency packets bringing information, such as telemetry and heartbeat, that are essential for the correct functioning and flight of the drone; on the other hand, they also exchange command and mission packets of various sizes, but only when a mission is uploaded or a command is sent to the drone. Considering this main difference between the two kinds of packets, the traffic model includes two separate ‘conversations’:

1. **Telemetry:** this traffic include all the small-size, high-frequency messages that do not directly determine any action of the UAV;
2. **Commands & Missions:** this traffic is generated by the user or by a mission script and makes the UAV execute specific actions.

The traffic model consists then of two streams of packets generated at the same time. The first flow simply consists of the replay of the ‘Telemetry’ traffic from the

size and inter-departure times of a 10-minute test session and saved in the files. The second flow consists of a series of packets of fixed size and sent at a constant time interval. For the experiments, it was decided to send a 44 Bytes package every 200 ms. During the experiments, DITG was launched using the command `./ITGSend bs`, where `bs` is the name of a file that contains all the required information for the generation of traffic for the case of a single UE/UAV attached to the Base Station. Listing 4.1 shows the content of the file `bs`.

```
1 -a 172.16.0.8 -Fs background_traffic_size_a -Ft
   background_traffic_time_a -T TCP -t 2700000
2 -a 172.16.0.8 -C 5 -c 44 -T TCP -t 2700000
```

Listing 4.1: Line 1 shows the traffic flow reproduced from file: `-Fs` indicates the file with the packets sizes, `-Ft` indicates the file with the packet inter-departure times. Line 2 shows the traffic flow artificially built: `-C` indicates the constant number of packets per second, `-c` indicates the packet size in Bytes. In both lines, `-a` indicates the IPv4 address of the recipient, `-T` indicates the TCP protocol, `-t` indicates the duration in milliseconds.

4.2.2 Design of Experiments

Once the traffic model of the communication between the GCS and the UAV has been defined and it is possible to reproduce it from the saved files, the experiments can be designed. Given the objectives expressed in Chapter 3, the outline of the experiments to be performed is straightforward. Simulated traffic is generated in the two different scenarios in different conditions, the traffic is recorded and analyzed with the tools already used (`tshark`), the data are cleaned, and finally the desired information is extracted from them. The analysis is focused on the performance indicators stated in the 3GPP Technical Report [3] and listed in Table 2.1, that are Latency, Data Rate, and Packet Loss Rate.

4.2.3 Real-World System Setup and Experimental Procedure

Performing the experiments in the real-world environment requires a hardware setup similar to, but not the same of, the one used for the traffic analysis. The new setup for the UAV and the GCS is detailed in Table 4.3. In this case, the traffic passing through the LTE connection is just a series of random binary information and does not convey any MAVLink message in any sense. Another communication channel between the two devices was guaranteed by a router, to which both devices were attached through an Ethernet cable.

The real world experiments performed consisted in recording with `tshark` the simulated traffic exchanged through the LTE channel between the BS and the

(Virtual) Component	Side	Hardware	Software
GCS		Laptop	LXD + SCOPE container
		USRP B210	tshark/Wireshark
			DITG
UAV		Intel NUC	LXD + SCOPE container
		USRP B210	tshark/Wireshark
			DITG

Table 4.3: Software configuration for the experiments in the real-world scenario. The GCS and the UAV are ‘virtual’ in this case, meaning that the drone flight and the functioning of the whole system does not depend on the simulated communication, through which just random binary information passes.

UE. A brief description of the experiments performed follows. The experiments in the real-world scenario were performed in two separate sessions, on July 20, 2022 and on July 28, 2022. During the first session, data were collected with the drone laying still on the ground with the propellers not rotating and disarmed in an urban environment³. In this condition, data were collected with an increasing distance between the BS and the UE: first 5.0 m, then 7.5 m, and lastly 10.0 m. During the second session, data were collected with the drone laying still on the ground, but armed and with active rotating parts, in the same urban environment of the previous session. In this condition, data were collected with an increasing distance between the BS and the UE: first 5.0 m, then 7.5 m, and lastly 10.0 m. During the second session, an additional experiment was performed, with a physical person bringing the disarmed drone floating around the BS at a distance ranging from 1 m to 7.5 m.

4.2.4 Emulated System Setup and Experimental Procedure

The experiments performed on Colosseum are similar to the ones performed in the real-world scenario in terms of software setup; however, the hardware is completely remote-located in the Northeastern facilities. The setup for the UAV and the GCS is described in Table 4.4. As in the case of the real-world scenario, the traffic

³The experiments were performed in the Carter Playground of the Northeastern University Boston Campus



(a)



(b)

Figure 4.8: View from distance of the setup for the data collection in the real-world scenario in the Carter Playground of the Northeastern University Boston Campus. The Ethernet cables and the router used to synchronize the clocks of the laptop and the companion computer is visible laying on the ground in white color; the antennas of the two USRPs used are visible close to the laptop and on the drone; the USRP attached to the BS is visible in a white case, while the USRP of the UAV is attached underneath the drone and is not clearly visible.

passing through the LTE connection is just a series of random binary information and does not convey any MAVLink message. In this case, differently from the real-world scenario, there is no need to create a parallel communication channel between the devices, since Colosseum allows the *ssh* login into each node and synchronizes their clocks by default.

Similar to what was done for the real-world scenario, data were collected in

(Virtual) Component Side	Hardware	Software
GCS	Laptop	LXD + SCOPE container
	USRP B210	tshark/Wireshark
		DITG
UAV	node of Colosseum	LXD + SCOPE container
	USRP B210	tshark/Wireshark
		DITG

Table 4.4: Software configuration for the experiments in the real-world scenario. The GCS and the UAV are ‘virtual’ in this case, meaning that the drone flight and the functioning of the whole system does not depend on the simulated communication, through which just random binary information passes.

different conditions. In this case, the data collection conditions must comply with the Colosseum scenario used, which was the scenario 12356 in our case, whose scheme is explained in 4.1.3.

The experiments performed in the emulated environment consisted in recording with tshark the simulated traffic exchanged through the LTE channel between the BS and the UE. A brief description of the experiments performed follows. During the several sessions of experiments, data were collected in different conditions by activating and using only some of the nodes available. The different conditions experimented are listed below, according to the nodes used:

- 1 BS and UAV 1;
- 1 BS and UAV 1, 2, and 3;
- 1 BS and UAV 1, and UE 1, 2, and 3.

Only one BS was used for all the experiments, since changing the position of the BS inside the virtual chamber does not affect the results of our experiments, as long as only one and only BS is used for all of them.

Chapter 5

Results

5.1 Data Cleaning and Preparation

Before the actual analysis, data have been prepared and cleaned. Collected data is in the form of `.pcapng` and `.csv` files: the first file format is the output of `tshark` and `Wireshark`, while the second file format is the output of `SCOPE`. Data have been cleaned of all those packets that were not part of the conversations under scope between the BS and the UAV (such as those packets sent to establish the network and sent for the attachment of the UE), and of those packets that showed inconsistent values (such as 0 round-trip time).

The collected data was then grouped in 4 subsets to help presenting the information. The four groups are described as follows:

1. emulated scenario¹ with 1 UAV (mobile UE) attached to the BS;
2. emulated scenario¹ with 3 UAVs (mobile UEs) attached to the BS;
3. emulated scenario¹ with 1 UAV (mobile UE) and three fixed UEs attached to the BS;
4. real-world scenario with 1 UAV attached to the BS.

Table 5.1 shows more information on the four subsets of data. As explained before, the traffic is simulated in all cases, as to make all the conditions equal except for the ones being investigated.

¹Colosseum scenario 12356, as described in Section 4.1.3.

Scenario	Group Number	Group Description	Number of packets	Collection time
Emulated	1	1 UAV	9065	0 h 37 min 54 s
Emulated	2	3 UAVs	34,781	2 h 52 min 57 s
Emulated	3	1 UAV + 3 UEs	11,395	1 h 3 min 19 s
Total			55,241	4 h 34 min 11 s
Real World	4	1 UAV	24,113	1 h 27 min 25 s
Total	-	-	79,354	6 h 1 min 36 s

Table 5.1: Subsets of data used for the portrayal of the results.

5.2 Data Representation and Analysis Tools

Different kinds of graphs are used to represent the results of the analysis of the collected data.

The first kind of graph is the color matrix created with the function `imagesc`, which shows how some variables of the packets of the traffic recorded are related to each other in terms of covariance, with the values represented through a colormap. This graph allows to see the correlation between each pair of variables for the dataset considered. An example of this graph is Figure 5.3a.

The second kind of graph used to represent and understand the data is the scatter plot with marginal histograms created with the function `scatterhist`: an example of such a graph is shown in Figure 5.4. Similarly to the scatter plots of the group scatter plot matrix, this kind of graph helps compare two scenarios and highlights the differences and similarities in the trends of *CQI* and *MCS*.

Another kind of graph used is the group scatter plot matrix created with the function `gplotmatrix`: an example is shown in Figure 5.9. This graph helps compare the scatter plots of two different scenarios (among the ones listed in Table 5.1) and highlight if the variables present the same trends in the different scenarios.

The last kind of graph used to represent the data is the round-trip time plot, created with an internal tool of Wireshark. Relative time in seconds with reference to the start of the experiment is on the x axis, while the round-trip time in milliseconds is on the y axis. An example of such graph is shown in Figure 5.13. This kind of graph is useful for our purposes because it shows what is the round-trip time for each packet sent, i.e. how much time elapses between the moment a TCP packet is sent and the moment the ACK TCP response is received by the same computer. This means that the time value shown in the graph does not correspond to the 'one-way latency' mentioned in Table 2.1 and taken from [3]: the round-trip time includes the amount of time required for the message to reach

the destination device (as the one-way latency), but also the time for the ACK response to reach the original sender of the message (excluded from the one-way latency computation). However, it is assumed for the moment that the round-trip time is approximately double the time of one-way latency: therefore, if the threshold for the one-way latency is of 50 ms, then the threshold for the round-trip time should be of 100 ms.

The variables taken into account for the analysis of the traffic for the first three kinds of graph (Sections 5.2.1, 5.2.2, and 5.2.3) are the following:

- MCS in downlink,
- CQI,
- transmitted packet in downlink,
- MCS in uplink,
- received packet in uplink,
- errors,
- SINR in uplink,
- sum of requested PRBs.

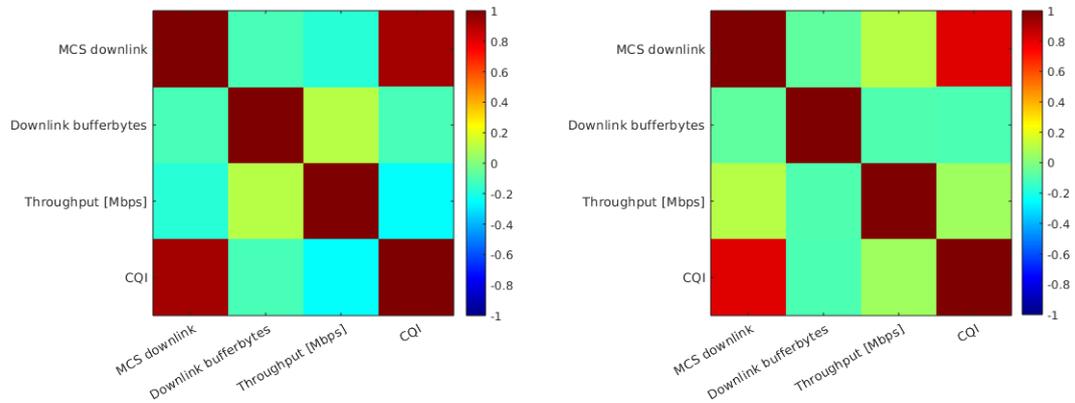
Other variables were considered at first, but it appeared that their behavior was similar to the behavior of the ones listed and therefore does not provide further information. For instance:

- the number of samples in downlink is similar to the number of packets in transmitted in downlink;
- the number of samples in uplink is similar to the number of packets received in uplink;
- the sum of requested PRBs has similar behavior to the sum of granted PRBs;
- the brate in Uplink for rx has similar behavior to the number of packets received in uplink.

5.2.1 Performance Indicators Covariances

Figure 5.1 shows a comparison of the color matrices of the real-world dataset and the Colosseum datasets considered together. the number of variables taken into account for this dataset is smaller: only *MCS*, *CQI*, *Throughput* and *Downlink buffer bytes* are considered. The correlation of the covariances of *MCS* and *CQI*

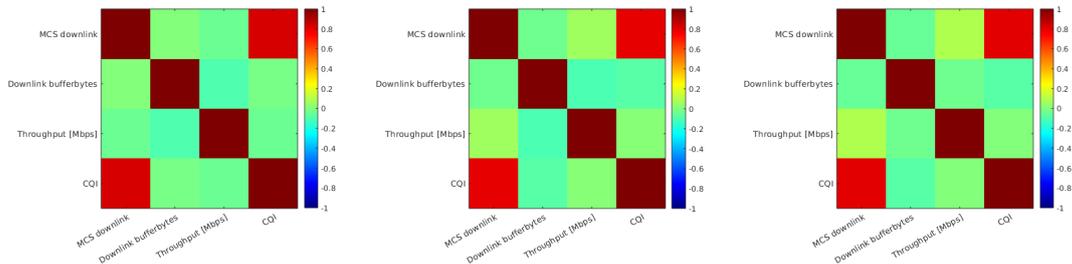
is high and close to 1 in both scenarios. The correlation between the covariances of *Downlink buffer bytes* and *Throughput* is slightly positive in both scenarios, while the correlation between the *MCS Downlink* and the *Downlink buffer bytes* is around 0 in the real-world scenario, but it is positive in the emulated scenario. It is counterintuitive that there is a negative correlation between *Throughput* and *CQI* in the real-world scenario. This could be explained considering that the *CQI* is very low and therefore the *Throughput* cannot have a high value.



(a) Color matrix for the real-world scenario (group number 4). (b) Color matrix for the Colosseum scenarios (groups number 1, 2, 3).

Figure 5.1: Comparison of the color matrices of the real-world scenario and the Colosseum scenarios, taking into account a small number of variables, i.e. MCS, Throughput, CQI, and downlink buffer bytes.

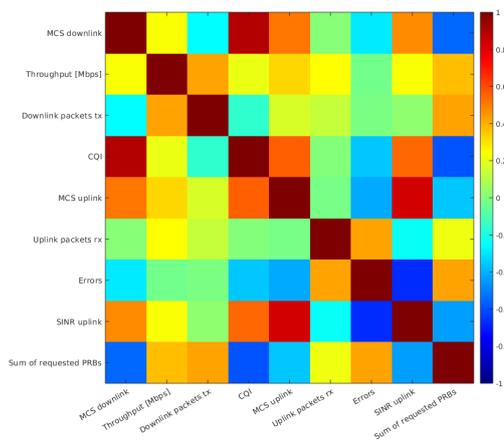
Figure 5.2 shows a comparison of the color matrices in the Colosseum scenario for the three conditions of the data collection considered separately. Similarly to what said for Figure 5.1, it is evident a strong correlation (~ 0.8) between the covariances of *MCS Downlink* and *CQI*, while the values of correlations for the other combinations of variables is around 0 or slightly positive.



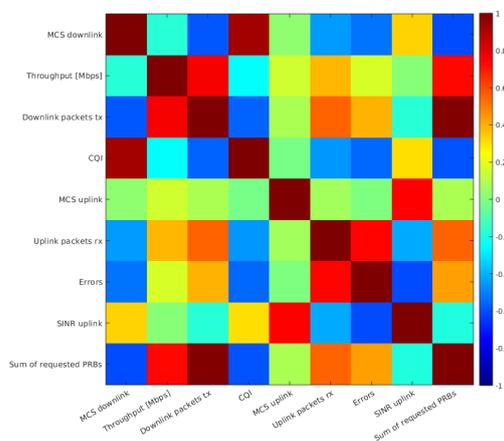
(a) Color matrix for the Colosseum scenario (group number 1). (b) Color matrix for the Colosseum scenario (group number 2). (c) Color matrix for the Colosseum scenario (group number 3).

Figure 5.2: Comparison of the color matrices of the three different Colosseum datasets, taking into account a small number of variables, i.e. MCS, Throughput, CQI, and downlink buffer bytes.

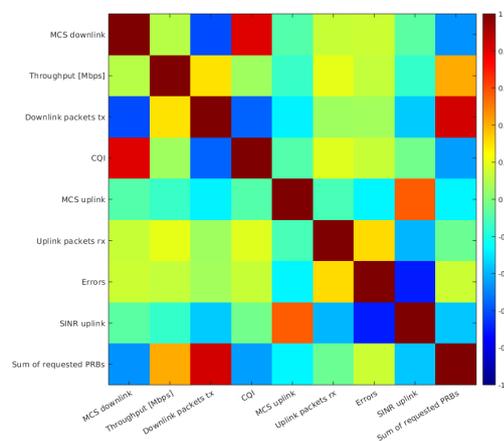
Figures 5.3a, 5.3b, 5.3c show the same color matrices for three dataset: 5.3a considers all the data available (both real world and Colosseum), 5.3b only considers the real-world data, 5.3c considers only the three Colosseum datasets. In all the three cases, more variables than in Figures 5.1 and 5.2 are taken into account. Looking at the color patterns of the three matrices, two main aspects come to the eye. First, from Figures 5.3b and 5.3c it emerges that the squares of the matrix referred to the Colosseum scenario have a color tone generally closer to the green, i.e. to the 0 (0 meaning no correlation), meaning that most variables are more loosely correlated in the Colosseum scenario; on the other hand, a stronger correlation between the covariances of some of the considered variables is present in the real-world scenario. Secondly, both in the real-world and the Colosseum scenarios, the *MCS in uplink* is correlated strongly only to the *SINR in uplink*.



(a) Color matrix for all the scenarios (group number 1, 2, 3, 4).



(b) Color matrix for the real-world scenario (group number 4).



(c) Color matrix for the Colosseum scenarios (groups number 1, 2, 3).

Figure 5.3: Comparison of color matrices for all data, real-world data, and Colosseum data.

5.2.2 CQI and MCS

The scatter plots with marginal histograms provide information about the differences and similarities of the different scenarios and conditions in terms of MCS and CQI . Figure 5.4 shows a comparison between the real-world scenario and the Colosseum scenario taken together. There is quite a big difference in terms of packets analyzed (24,113 for the real-world scenario versus 55,241 for the Colosseum ones), but the number of packets is high enough in both cases to be able to make statistical inferences. The quality of the channel, given by the CQI , is definitely higher in the Colosseum scenario. As a consequence, the MCS is also higher in the Colosseum scenario. Such correlation is expected and is in agreement with the graphs shown in Figures 5.1a, 5.2, and 5.3. The new information provided by this graph consists in the higher values of MCS and of CQI in the Colosseum scenario with respect to the real-world scenario. From the marginal histograms, it also emerges clearly that the real-world scenario data are much more scattered than the ones of Colosseum scenario. and that the Colosseum data generally show higher value of both CQI and MCS with respect to the real-world data.

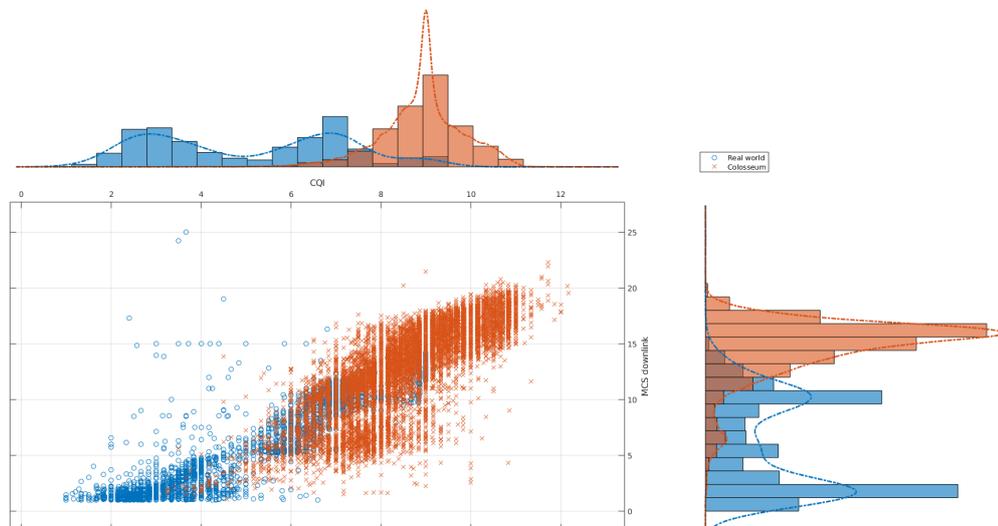


Figure 5.4: Scatter plot with marginal histograms with the comparison between the real-world scenario (groups number 4) and the Colosseum scenarios considered together (groups number 1, 2, 3).

Figure 5.5, on the other hand, show the same data, but this time the packets collected in the Colosseum scenarios are divided according to the subset of data to which they belong in order to highlight the differences between the three groups

1, 2, 3. It is possible to qualitatively order the three different environmental conditions of the Colosseum scenario and the real-world scenario according to the peak of the value distribution of CQI : there are, in descending order, the colosseum scenario with 1 UAV (group 1) and the Colosseum scenario with 1 UAV and 3 fixed UEs (group 3), then the Colosseum scenario with 3 UAVs (group 2), and lastly the real-world scenario with 1 UAV (group 4).



Figure 5.5: Scatter plot with marginal histograms with the comparison among the real-world scenario and the Colosseum scenarios considered individually (groups number 1, 2, 3, 4).

Another useful grouping of the same data in the same kind of scatter plots with marginal histograms is provided in Figures 5.6, 5.7, and 5.8. These three graphs confirm what stated above, that the Colosseum scenarios show overall better performances with respect to the ral-world scenario.

¹With reference to Table 5.1.

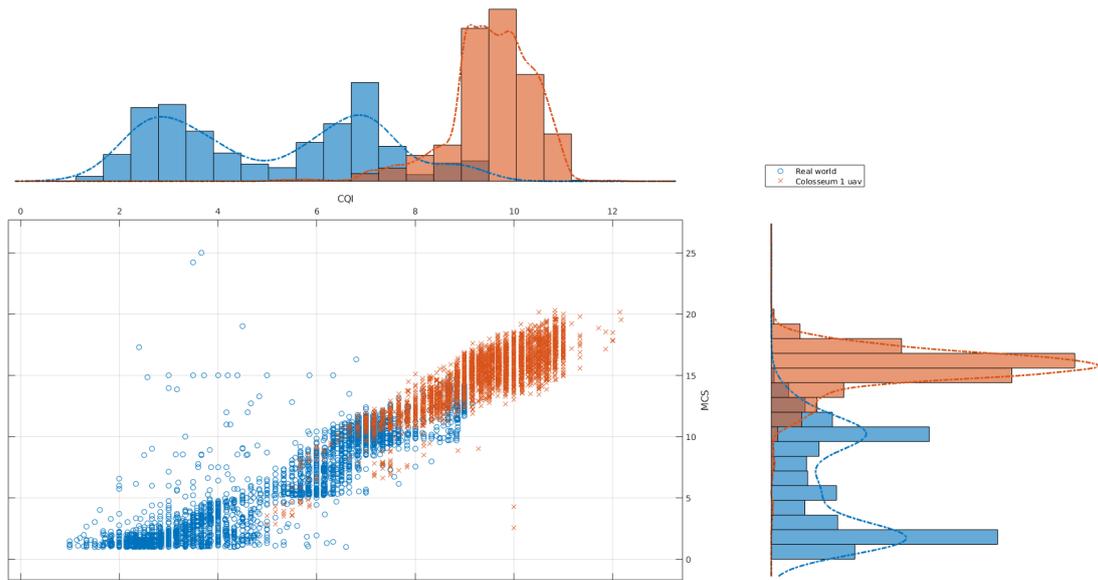


Figure 5.6: Scatter plot with marginal histograms with the comparison between the real-world scenario (groups number 4) and the Colosseum scenario (group number 1).

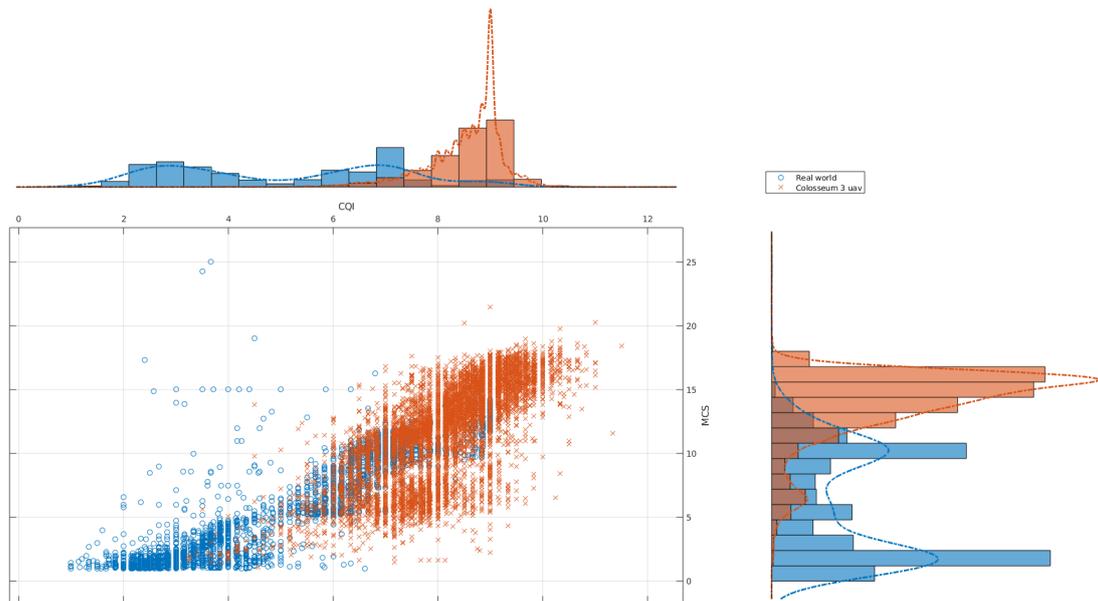


Figure 5.7: Scatter plot with marginal histograms with the comparison between the real-world scenario (groups number 4) and the Colosseum scenario (group number 2).

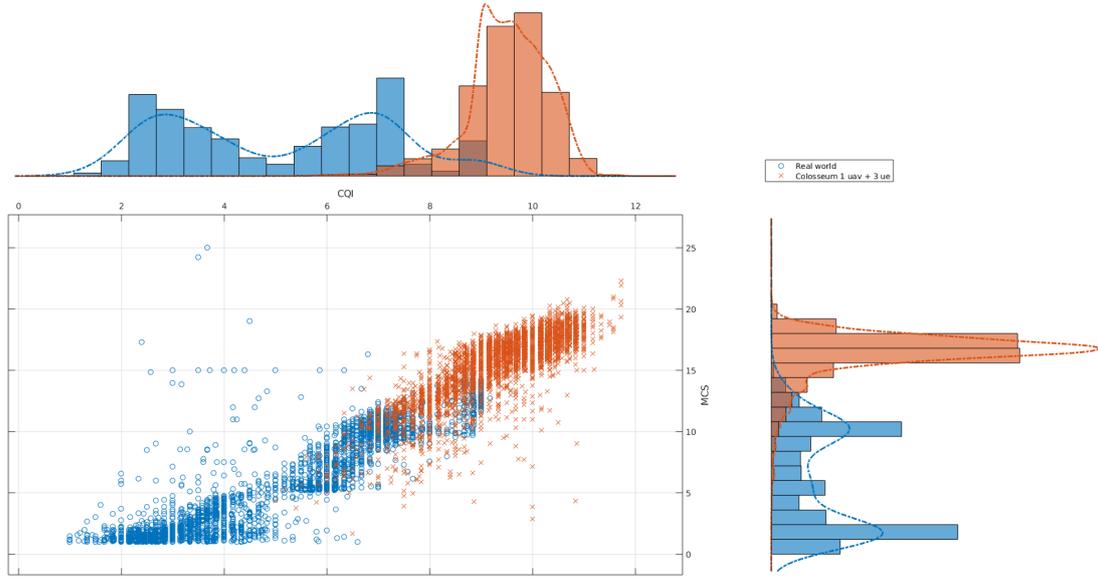


Figure 5.8: Scatter plot with marginal histograms with the comparison between the real-world scenario (groups number 4) and the Colosseum scenario (group number 3).

5.2.3 Performance Indicators Trends

The group scatter plot matrix diagrams are shown in Figures 5.9, 5.10, 5.11, and 5.12. The diagonal of the matrix show the distribution corresponding to the variable on that row (or column, as they correspond for the diagonal), while the scatter plots in the non-diagonal cells are scatter plot of the same kind of the ones shown in Figures 5.5, 5.6, 5.7, and 5.8.

Figure 5.9 shows a comparison of the real-world scenario and the three Colosseum scenarios aggregated: it clearly emerges that Colosseum scenarios have better performances, since the peaks of the histograms on the diagonal of the matrix show peaks for the Colosseum data at higher values: that is the case for all the variables except *Downlink bufferbytes* and *Sum of requested PRBs*. The fact that figures for Colosseum scenario show better performance is expected and is in agreement with the graphs and plots shown in the previous Sections 5.2.1 and 5.2.2. This feature is also common to the other three plots shown in Figures 5.10, 5.11, and 5.12.

From the plot matrices in Figures 5.9, 5.10, 5.11, and 5.12 it also appears evident that the non-diagonal plots of the matrix generally show similar patterns and trends for both the real-world scenario and the aggregated data of the Colosseum one. For instance, Figure 5.9 (as well as Figures 5.10, 5.11, and 5.12) shows similar distribution of data in the *CQI-MCS Downlink*, *Downlink packets tx-MCS Downlink*, *Throughput [Mbps]-Sum of requested PRBs*, and *SINR uplink-Uplink packets rx* plots.

Another interesting feature that emerges from the plots is the fact that the data of the Colosseum scenario with only 1 uav only (corresponding to group 1) shown in Figure 5.10 is less scattered than those of the two other Colosseum scenarios with 3 UAVs (corresponding to group 2) and with 1 UAV and 3 UEs (corresponding to group 3) shown in Figures 5.11 and 5.12 respectively. This is also expected, since the number of devices connected to the same BS increases the amount of traffic over the network and all the performance indicators are likely to present worse figures: as a consequence, the data points shown in the Figures 5.11 and 5.12 (corresponding to groups 2 and 3 respectively) are more scattered, indicating that the values of the performance indicators are also more scattered and can have lower values. Nonetheless, in agreement with what stated before, the network performance of the real-world scenario is always worse than that of the Colosseum scenario.

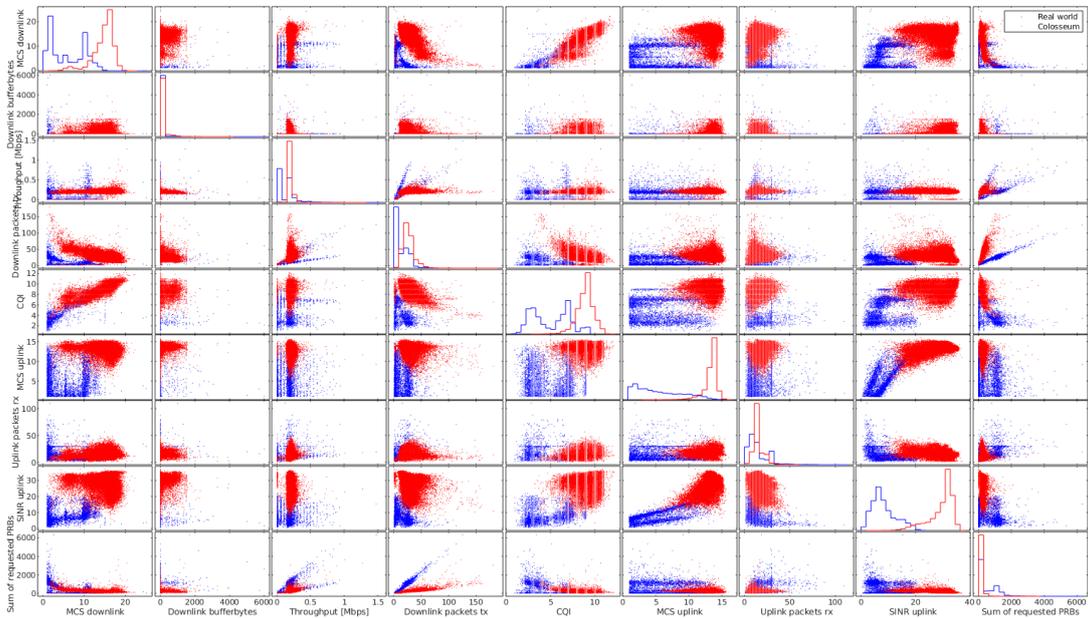


Figure 5.9: Group scatter plot matrix with the comparison between real-world scenario (group number 4) and Colosseum scenarios aggregated (group number 1, 2, 3).

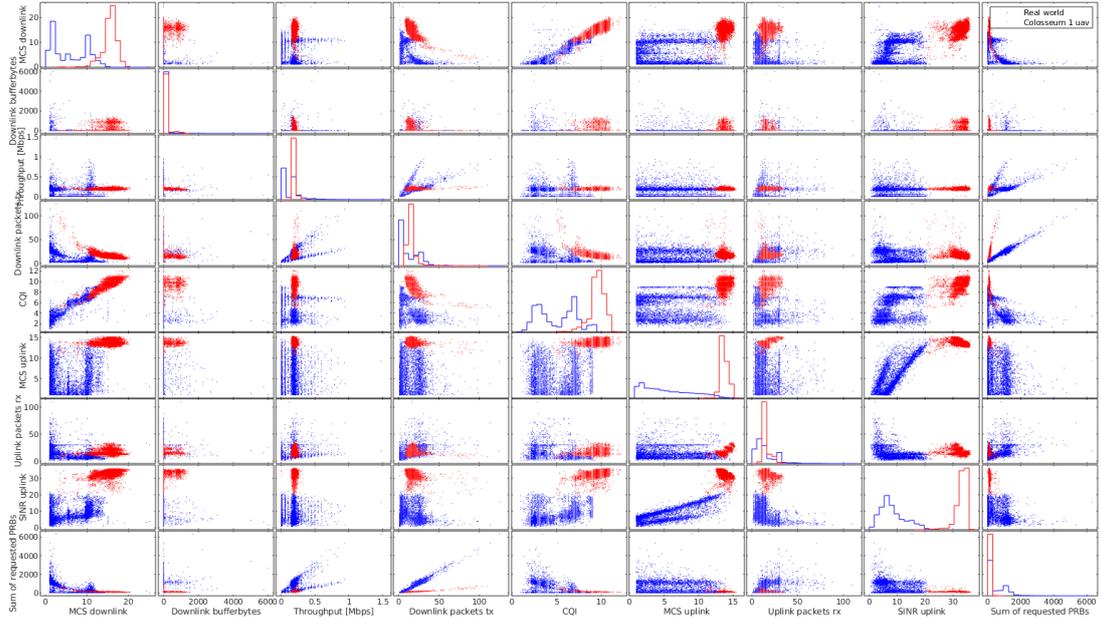


Figure 5.10: Group scatter plot matrix with the comparison between real-world scenario (group number 4) and Colosseum scenario with 1 UAV (group number 1).

5.2.4 Round-Trip Time Plots

Lastly, the round-trip time plots are shown in Figures 5.9, 5.10, 5.11, and 5.12. Figure 5.13 shows the round-trip times of packets sent from the BS to the UAV and *vice versa* in the Colosseum scenario with 1 UAV. Figure 5.14 shows the round-trip times of packets sent from the BS to the UAV and *vice versa* in the real-world scenario with a single UAV at a distance of approximately 5 m from the BS: in this second case Figures 5.14a and 5.14b highlight the position of the 100 ms threshold with a red line.

By looking at all these Figures, it is evident that, despite a large share of packets has a round-trip time of less than 100 ms, still many packet exceed the threshold and there are even spikes of more than 1 s in both scenarios. Anyway, the plots confirm once again that the Colosseum scenario presents less packets with a RTT greater than 100 ms, thus indicating that it generally has better performances in comparison to the real-world scenario.

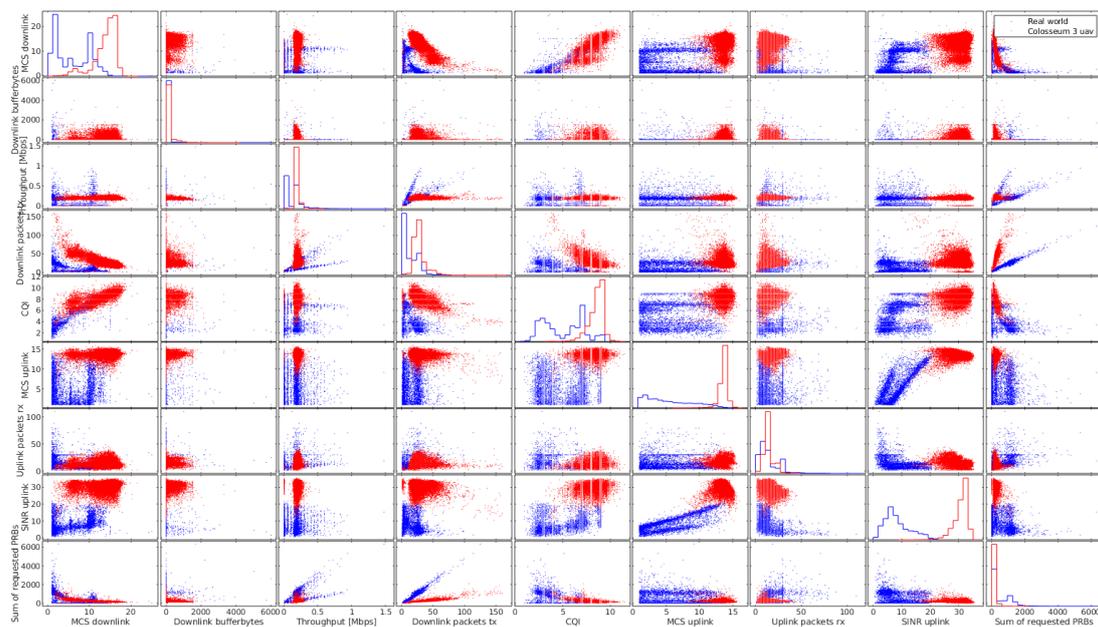


Figure 5.11: Group scatter plot matrix with the comparison between real-world scenario (group number 4) and Colosseum scenario with 3 UAVs (group number 2).

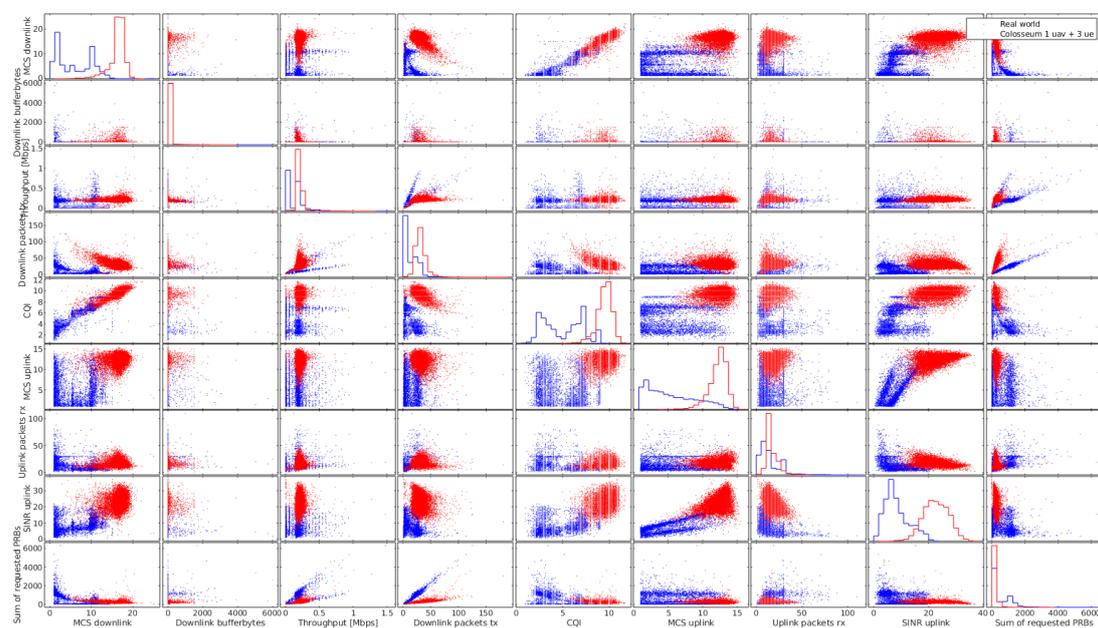
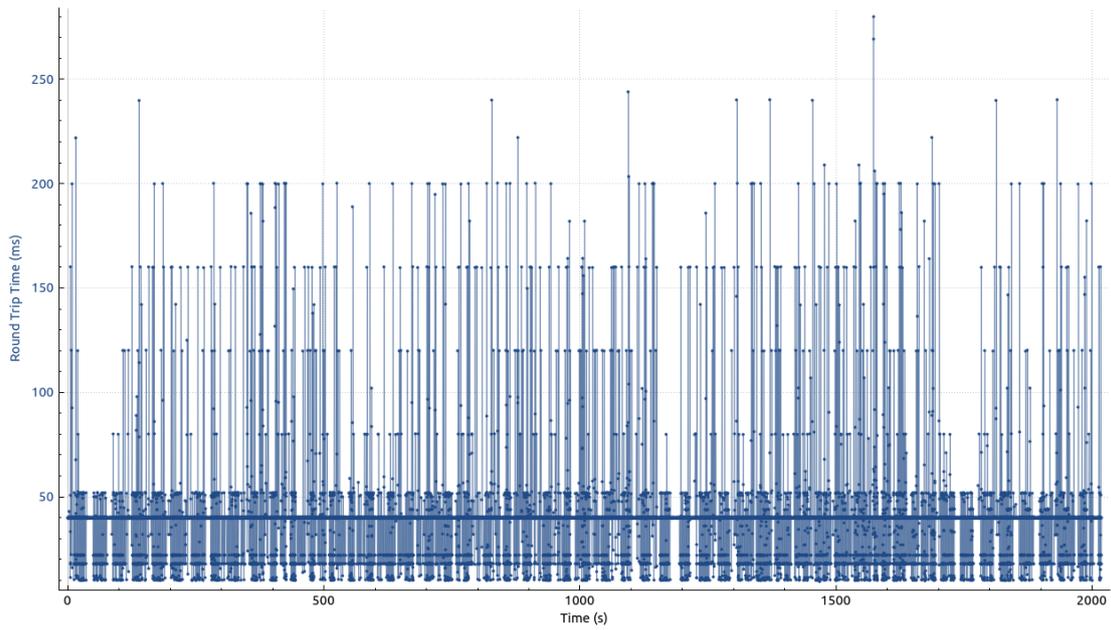
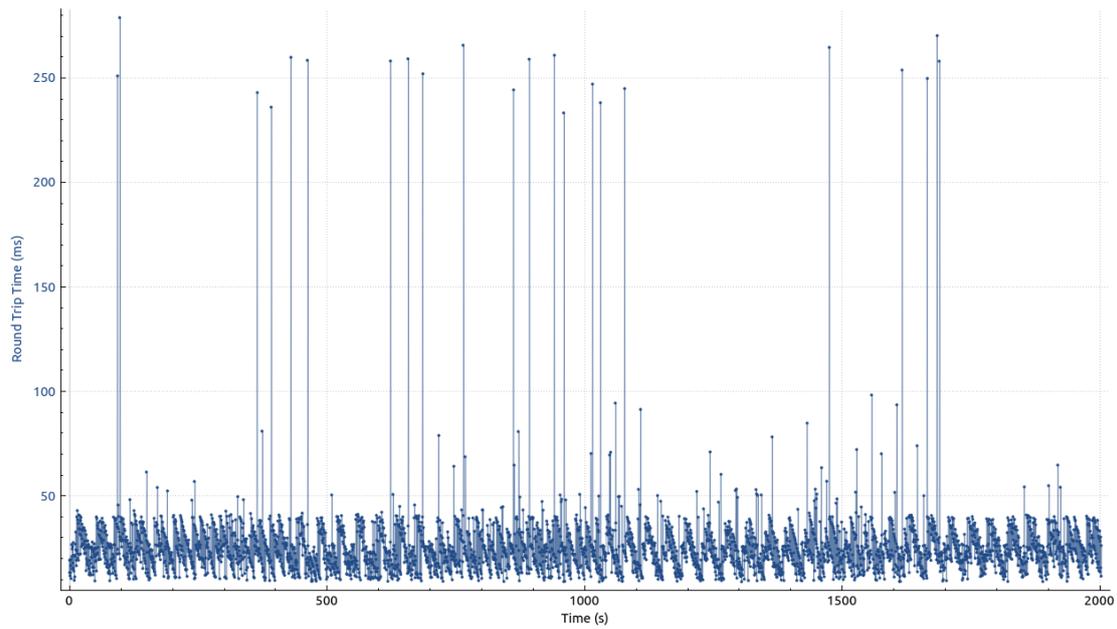


Figure 5.12: Group scatter plot matrix with the comparison between real-world scenario (group number 4) and Colosseum scenario with 1 UAV and 3 UEs (group number 3).

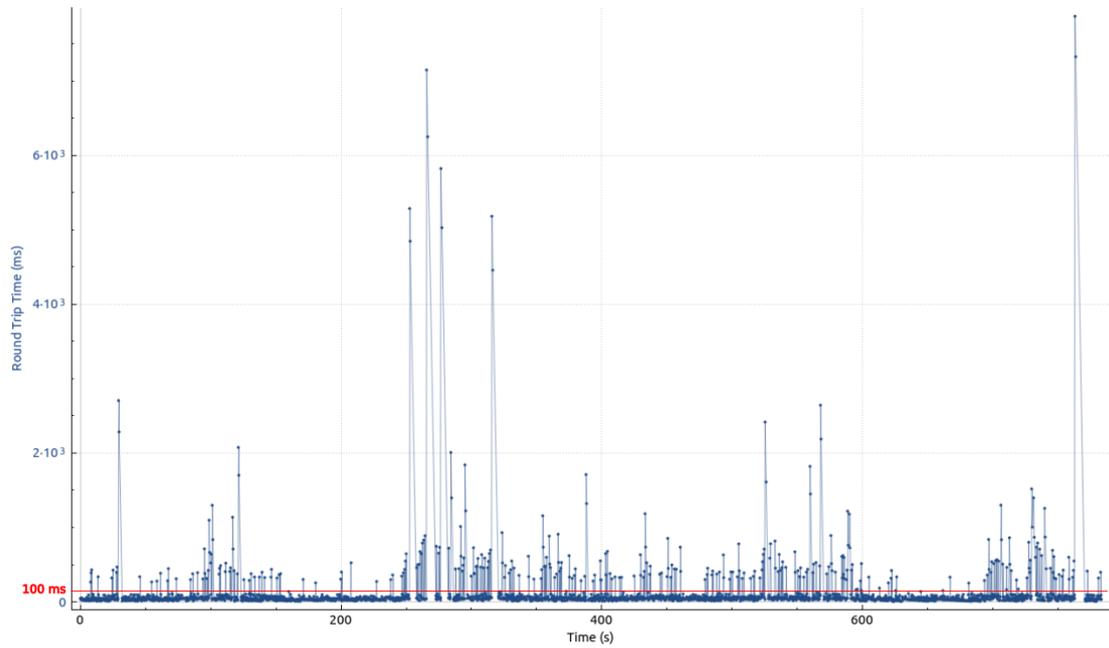


(a) Round trip plot for the colosseum scenario with 1 UAV. Packets sent from BS to UAV.

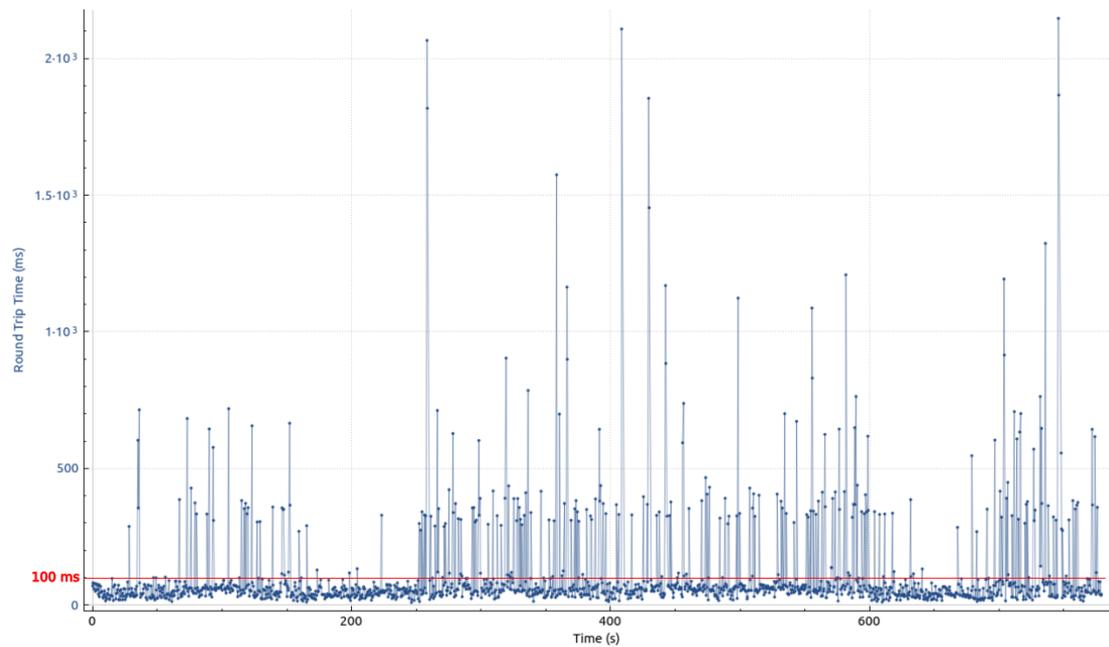


(b) Round trip plot for the colosseum scenario with 1 UAV. Packets sent from UAV to BS.

Figure 5.13: Round-trip time plots for the Colosseum scenario with 1 UAV.



(a) Packets sent from BS to UAV: zoom on lower part of graph: zoom on the 100 ms threshold.



(b) Packets sent from UAV to BS: zoom on the 100 ms threshold.

Figure 5.14: Round-trip time plots for the real-world scenario with the UAV distant approximately 5 m from the BS. The threshold of 100 ms is highlighted with a red line.

Chapter 6

Discussion

Considering the performance indicators covariances color plots, the four subset of data show similar behavior. However, the subsets of data related to the emulated scenario show values closer to either 0, 1, or -1 , meaning that the correlation between the covariances of the variables taken into considerations is either present or absent. Conversely, the data collected in the real-world scenario show different tones of colors, meaning that the correlation between the covariances of the different variables also has intermediate values (between 0 and 1 or between 0 and -1).

Considering the scatter plots with marginal histograms, it is possible to infer that the channel quality is definitely better in the emulated scenario rather than the real-world scenario. Furthermore, even if the channel quality in the emulated scenario is good, it is still possible to differentiate between the three environmental conditions (corresponding to the three subsets of data group 1, group 2, and group 3): figures of the groups with 1 UAV only (groups 1 and 3) show the best channel quality (highest value of CQI), while the group with 3 UAVs has slightly worse figures for such parameter.

Considering the scatter plot group scatter plot matrices, a few facts can be deducted. First, the performance of the emulated scenario is overall better than that of the real-world scenario, in line with what already deducted before from the the scatter plots with marginal histograms. The second fact that can be observed, which is still in line with what stated above about the scatter plots with marginal histograms, is that the emulated scenario with conditions corresponding to group 1 are the best and least scattered data among all, followed by the two other subsets of data of the emulated scenario, and then by the real-world data. However, the most interesting fact here is the similarity in the data scattering of the non-diagonal plots between the data of emulated scenario and those of the real-world scenario.

Considering the round-trip plots, it is possible to see that a great majority of the packets collected have a RTT of less than 100 ms, but there are still many

packets that do not comply with that threshold (and thus the cellular link does not comply with the 3GPP standard). These plots confirm once again the fact that the emulated scenario has better performances with respect to the real-world scenario, also in terms of latency.

To summarize what observed from the analysis of data collected, the following can be stated.

1. The number of packets used for the analysis after the preparation and cleaning of data is high, and is reckoned to be enough for the purposes of this thesis. However, there is an evident gap in the amount of packets collected in the real-world scenario (24113) and the emulated scenario (55241), with the ratio between the two being lower than 0.5, confirming the belief that collecting huge amount of data is easier in an emulated scenario rather than in the real world.
2. Figures for the emulated scenario show evident correlation or non-correlation of the covariance between variables with values close to 1, 0, or -1 , while figures for the real-world scenario do not show such clear correlation or non-correlation for all pairs of variables considered. This fact could be related to the presence of disturbance and interference that characterizes the real-world scenario and is almost absent from the emulated scenario¹.
3. In general, figures for the emulated scenario show better performance with respect to the real-world scenario, and data points of the emulated scenario in the scatter plots are less scattered than the data points of the real-world scenario. The poor performance of the real-world system in comparison to the emulated system is due to the interferences and disturbances present in the real world, as well as to the use of SNRs instead of single-purpose cellular radios for the UE and the BS. The use of SNRs for the wireless communication network has probably also influenced the maximum range of communication between the two devices: the quality of the wireless channel saw a steep decrease between 10 m and 15 m, with the impossibility of establishing a communication at a range longer than 15 m.
4. Despite the evident differences in performance and in data scatter between the two scenarios, the general behavior of the system in the two scenarios is similar and show corresponding trends. This allows to consider the emulator Colosseum as a proper testbed for the design and development of new applications and features to improve the performance of cellular support for UAVs

¹The word ‘almost’ is justified by the presence of some racks as furniture of the modelled chamber of the scenario used (number 12356).

and thus meet the requirements of the 3GPP for enhanced LTE support for the UAS Traffic Management (UTM).

Chapter 7

Conclusion

This thesis aims at studying the feasibility of using UAVs in BVLOS applications and focuses on the cellular communication support for the UAVs. The communication between GCS and UAVs is studied in two different scenarios, a real-world one and an emulated one. Showing that the two systems have similar behavior, despite presenting different performances, enables the use of hardware-in-the-loop emulators to develop new features to improve the cellular support for UAVs and comply with the 3GPP standards [3], and then to prototype solutions for cellular-supported UAVs in BVLOS flight.

The study has been conducted by collecting over-the-air measurements of the real system, as well as by collecting data of the emulated system within a hardware-in-the-loop emulator. To collect over-the-air measurements the UAV *Monarch*, a drone suitable for autonomous applications, was used, while the massive channel emulator *Colosseum*, available at Northeastern facilities, was used for the hardware-in-the-loop experiments. For both the UAV and the channel emulator, previous software developed internally at the Institute for the Wireless Internet of Things was used to manage some of the resources: this include the SCOPE software and container, used to manage the USRPs in the real world (model B210) and on the hardware-in-the-loop emulator (model X310). The traffic recorded and analyzed has been generated by a traffic generator that reproduced a model of traffic designed on the real traffic exchanged between a GCS and a UAV.

The two different scenarios used to test the system allow to distinguish and understand the different environmental conditions imposed on the UAVs. From the statistical analysis of the collected data, conclusions are drawn about the performance of the system in the two scenarios, and about the quality of the traffic model. The traffic model is thoroughly analyzed, and the traffic is reproduced in a simple and effective manner, that is by recording real traffic and sending simulated packets with the same size and frequency of the real traffic. The collected data has been divided into three datasets to highlight differences. The kinds of graphs

and plots have been carefully chosen to best present and highlight the intended inferences.

The results obtained about the performances of the emulated and real systems and about the similarities of their behavior are satisfactory and match the expected outcomes. The two systems, the real-world one and the emulated one, show similar behaviors, despite having significant differences in terms of performances. This allows to consider the emulator (Colosseum) as a proper testbed for the design and development of new applications and features to improve the performance of cellular support for UAVs and thus meet the requirements of the 3GPP for enhanced LTE support for the UAS Traffic Management (UTM). Once the development reaches a good phase, the new features can be tested on the real-world UAVs. This allows to save a lot of time, since the management of the resources and the data collection is easier and faster on Colosseum, while the real-world scenario increases the level of complexity of the environment and adds stochasticity to the experiments. The real-world system still represents a good testbed to test new features once they are validated in the emulator. However, the quality of the channel should be improved, in order to allow proper testing on the flying UAVs.

As for the study about the latency, a short note should be added. In this thesis, it is assumed that the round-trip time for a message is approximately double the one-way latency. Based on this assumption, the results obtained for the round-trip times of the messages show that most of the packets get to destination within the expected time, but a certain number of them take longer, thus not respecting the requirements imposed by 3GPP. Moreover, the analysis of this parameter confirms that there is a huge difference between the real-world scenario and the emulated one in terms of wireless network performance. However, the underlying assumption of the one-way latency being half of the RTT should be confirmed for the situations in which this parameters plays a role in the tests.

7.1 Future Work

The continuation of this work is already outlined. First, the exact same environments of the emulated scenarios are going to be created in the real world: the emulated scenarios currently available are based on the anechoic chamber (another facility available at Northeastern University), and that is where the UAVs will be flying in the future, ready for the tests after the development phase on the emulator.

At a more practical level, it is also possible to identify some issues faced during the experiments. As mentioned in Chapter 6, the radio communication range between the UAV and the BS in the real world is very short and forced to collect data

at a maximum distance of a bit more than 10 meters: this is due to the characteristics of the radio model, and to the interferences in the real world coming from other sources. The use of a commercial 4G-LTE/5G radio for the drone is believed to improve the signal quality and the overall wireless network performances, and to increase the communication range.

As for the simulated traffic, a further study on the traffic generator used (DITG), would benefit this research. Despite the many features of DITG, it results to be limiting in some sense, since some functions seem to be malfunctioning (or the way they work is obscure): one example is the minimum size of TCP packets sent, which is 24 B. A deeper analysis on its functions and features is definitely required, and it could also lead to the decision of coding of a new, simpler custom traffic generator that satisfies the peculiar needs determined by this work.

The variables taken into consideration so far in this work are sufficient to infer the behavior of the two systems and their differences: therefore, it is not believed that a further analysis in this sense is required. As for the study of the latency, two things are to be mentioned. First, the fact that not all packets satisfy the 3GPP requirements paves the way for the development of intelligent solutions for the improvement of the network performance and thus ensure the support of the cellular network for UAVs. This will be object of further investigations in the future. Secondly, the assumption about the one-way latency being approximately half of the round-trip time seems reasonable, but further explorations are required to support such assumption when this parameter will be used to test and validate new features implemented to improve the performance of the cellular network for the UAV support. The need to measure the one-way latency precisely is the reason why the NTP network was setup (see Section 4.1.6): the synchronization of the devices is then ready for further and more adequate studies.

Appendix A

Glossary and Acronyms

3GPP	3rd Generation Partnership Project: union of seven telecommunications organizations that produces reports and specifications for cellular communications technologies.
4G	Fourth generation.
5G	Fifth generation.
AR	Augmented Reality.
AL	Autonomy Level.
ATM	FAA's Aircraft Traffic Management system.
B5G	Beyond fifth generation.
BLOS	Beyond Line-Of-Sight: it refers to the state or scenario of a remotely controlled UAV that can not receive RF signals for commands and controls and must be autonomous or receive commands through other means (e.g. cellular network).
BVLOS	Beyond Visual Line-Of-Sight: similar to BLOS and often used interchangeably, it refers to the a scenario case where the remote UAV operator can not see the drone.
C&C	Command and Control.
Colosseum	Massive channel emulator and traffic generator born to support DARPA's Collaborative Spectrum Challenge and now owned, managed, and made available by Northeastern University.

CQI	Channel Quality Indicator.
DITG	Distributed Internal Traffic Generator [28].
ENB	Evolved Node B.
EPC	Evolved Packet Network.
FCU	Flight Control Unit: controller that manages the basic tasks and components of the drone, such as reading the different sensors and fusing the information, and controlling the motors.
GCS	Ground Control Station: control center that provides the interface for the control of UAVs.
GUI	Graphical User Interface.
LOS	Line of Sight.
LTE	Long Term Evolution.
LXC	Linux container runtime.
LXD	Enhanced Linux container runtime and virtual machine manager, that allows control of virtual machines over the network, too.
Mavlink	Application Layer protocol used in the drone industry.
MCS	Modulation and Coding Scheme.
NG	Next generation.
ns-3	'ns-3 is a discrete-event network simulator for Internet systems' [31].
RF Controller	Controller through which the drone can be controlled and commanded manually.
RSRP	Reference Signal Received Power.
SCOPE	'Development environment for softwarized and virtualized NextG cellular networks' [24].
SDR	Software-Defined Radio: radio system in which components that are traditionally analog are instead implemented through software.

SINR	Signal-to-Noise Ratio.
SNR	same as SINR.
SRN	Standard Radio Node, name of each node on Colosseum.
tshark	Network traffic dumper and analyzer.
UAS	Unmanned Aerial System: indicates the combination of a UAV, a GCS, and the communication link between the two.
UAV	Unmanned Aerial Vehicle: it indicates a drone that is remotely controlled.
UE	User Equipment, technical term used to indicate the mobile cellular user.
USRP	Universal Software Radio Peripheral: family of SDRs by National Instruments commonly used for research purposes.
URLLC	Ultra-Reliable Low-Latency Communications.
UTM	UAS Traffic Management.
VM	Virtual Machine.
VR	Virtual Reality.
Wireshark	Network protocol analyzer provided with a GUI.

Bibliography

- [1] Debashisha Mishra and Enrico Natalizio. «A survey on cellular-connected UAVs: Design challenges, enabling 5G/B5G innovations, and experimental advancements». In: *Computer Networks* 182 (2020), p. 107451.
- [2] *Unmanned Aircraft System Traffic Management (UTM)*. URL: https://www.faa.gov/uas/research_development/traffic_management (visited on 09/15/2022).
- [3] 3GPP. *3GPP TR 36.777 V15.0.0 (2017-12) - Study on Enhanced LTE Support for Aerial Vehicles (Release 15)*. 2017. URL: https://www.3gpp.org/ftp/Specs/archive/36_series/36.777/36777-f00.zip (visited on 08/11/2022).
- [4] Leonardo Bonati et al. «Colosseum: Large-scale wireless experimentation through hardware-in-the-loop network emulation». In: *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2021, pp. 105–113.
- [5] Lorenzo Bertizzolo et al. «Arena: A 64-antenna SDR-based ceiling grid testing platform for sub-6 GHz 5G-and-Beyond radio spectrum research». In: *Computer Networks* 181 (2020), p. 107436.
- [6] Northeastern University. *Anechoic Chamber and Drone Cage*. URL: <https://facilities.northeastern.edu/anechoic-chamber/> (visited on 11/19/2022).
- [7] AERPAW. *AERPAW*. URL: <https://aerpaw.org/> (visited on 09/27/2022).
- [8] *Release 15*. LTC3600. 3GPP. 2019. URL: [3gpp.org](https://www.3gpp.org).
- [9] *Release 16*. LTC3600. 3GPP. 2019. URL: [3gpp.org](https://www.3gpp.org).
- [10] *Release 17*. LTC3600. 3GPP. 2019. URL: [3gpp.org](https://www.3gpp.org).
- [11] *Release 18*. LTC3600. 3GPP. 2019. URL: [3gpp.org](https://www.3gpp.org).
- [12] 3GPP. *UAS - UAV*. 2019. URL: <https://www.3gpp.org/uas-uav> (visited on 08/11/2022).

- [13] William D Ivancic et al. «Flying drones beyond visual line of sight using 4g LTE: Issues and concerns». In: *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE. 2019, pp. 1–13.
- [14] Siva D Muruganathan et al. «An overview of 3GPP release-15 study on enhanced LTE support for connected drones». In: *IEEE Communications Standards Magazine* 5.4 (2021), pp. 140–146.
- [15] Samira Hayat et al. «An experimental evaluation of LTE-A throughput for drones». In: *Proceedings of the 5th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*. 2019, pp. 3–8.
- [16] Changyang She et al. «Ultra-reliable and low-latency communications in unmanned aerial vehicle communication systems». In: *IEEE Transactions on communications* 67.5 (2019), pp. 3768–3781.
- [17] Rafhael Amorim et al. «Radio channel modeling for UAV communication over cellular networks». In: *IEEE Wireless Communications Letters* 6.4 (2017), pp. 514–517.
- [18] Muhammad Naveed et al. «Evaluation of video streaming capacity of UAVs with respect to channel variation in 4G-LTE Surveillance Architecture». In: *2019 8th International Conference on Information and Communication Technologies (ICICT)*. IEEE. 2019, pp. 149–154.
- [19] Antonia Stornig et al. «Video Quality and Latency for UAV Teleoperation over LTE: A Study with ns3». In: *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE. 2021, pp. 1–7.
- [20] Mitch Champion, Prakash Ranganathan, and Saleh Faruque. «UAV swarm communication and control architectures: a review». In: *Journal of Unmanned Vehicle Systems* 7.2 (2018), pp. 93–106.
- [21] Rajeev Gangula et al. «Flying rebots: First results on an autonomous UAV-based LTE relay using open ainterface». In: *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE. 2018, pp. 1–5.
- [22] Peter J Burke. «A safe, open source, 4G connected self-flying plane with 1 hour flight time and all up weight (AUW) < 300 g: towards a new class of internet enabled UAVs». In: *IEEE Access* 7 (2019), pp. 67833–67855.
- [23] John Buczek et al. «What is A Wireless UAV? a design blueprint for 6G flying wireless nodes». In: *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization*. 2022, pp. 24–30.

- [24] Leonardo Bonati et al. «SCOPE: An open and softwarized prototyping platform for NextG systems». In: *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. 2021, pp. 415–426.
- [25] *MAVLink Developer Guide*. URL: <https://mavlink.io/en/> (visited on 08/31/2022).
- [26] *mavlink-router*. URL: <https://github.com/mavlink-router/mavlink-router> (visited on 08/31/2022).
- [27] *QGroundControl*. URL: <http://qgroundcontrol.com/> (visited on 09/06/2022).
- [28] Alessio Botta and Alberto Dainotti and Antonio Pescapè. «A tool for the generation of realistic network workload for emerging networking scenarios». In: *Computer Networks* 56.15 (2012), pp. 3531–3547.
- [29] *Network Time Protocol*. URL: <http://www.ntp.org/> (visited on 09/11/2022).
- [30] *IEEE1588-PTP: A Precision Time Protocol Implementation*. URL: <https://github.com/bestvibes/IEEE1588-PTP> (visited on 09/11/2022).
- [31] *ns-3*. URL: <https://www.nsnam.org/> (visited on 10/07/2022).