Leveraging Next Generation Mobile Networks for Drone Telemetry and Payload Communication

Architectural design and implementation

Presented by:
Ngonidzashe Mombeshora

Prepared for:
Dr Jane Wyngaard and Dr Joyce Mangwama
Dept. of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in fulfilment of the academic requirements for a Master of Science degree in Electrical and Computer Engineering.

February 2023

Key words:
4G; 5G; New Radio; Drone communications

Declaration

1. I know that plagiarism is wrong. Plagiarism is using another person's work and pretending that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Ngonidzashe Mombeshora

Signature: Signed by candidate

Date:10/02/2023

Acknowledgements

I would love to extend my most sincere gratitude to the following people who have been my cornerstone and pillars of strength and support throughout the duration of this project as well as my undergraduate studies:

My family, especially my parents for their unwavering support and sacrifices made to get me to and through University.

Dr Jane Wyngaard for her support as my supervisor. For the funding of my Masters' studies, and always being there in times of need.

Dr Joyce Mangwama for her support as my supervisor, and for making provisions of all the needed equipment to see this project to completion.

To **LANDRS and the Alfred P. Sloan Foundation** for making all this endeavour possible with the research grant.

To my **friends**. You made the journey much more bearable and memorable.

Abstract

small Unmanned Aerial Systems (sUAS) have seen their adoption increasing over the past recent years. The adoption is by hobbyists for leisure or by the industry for business and commercial use and as such, use case applications may vary enormously. Such use cases include but are not limited to drone delivery, precision agriculture, search and rescue and surveillance. As the adoption continues to increase, so do the use cases and drone applications. However, drones have much more to offer, and their capabilities are not to be limited to the current possible applications. There is a plethora of drone applications that have not been made possible, mainly due to technological limitations. The main limitation to be addressed in this project pertains to communication.

Drone use cases such as 8K video streaming, Augmented Reality and Virtual Reality (AR/VR), autonomous flights, and long-range surveillance requiring Beyond Visual Line of Sight (BVLOS) command and control are yet to be realized with efficiency for commercial viability. Limitations to be addressed in terms of communication include line of sight usage, data rates and latencies. This project investigates the use of mobile/cellular networks, specifically 5G (Fifth Generation) mobile networks, as a feasible option to address these limitations.

Experiments will be done by creating a mobile network test-bed using open-source mobile network stacks such as OpenAirInterface and integrating that with current drone communication technologies such as MAVlink to realize a drone communication stack that utilizes mobile networks for communication. 4G Long Term Evolution (LTE), 5G Non-Standalone (NSA) and a 5G Standalone (SA) test-bed stack will be implemented, and flight tests will be carried out to draw out and assess the advantages and disadvantages that cellular networks bring forth. And how 5G can push forward the drone ecosystem towards more novel and unrealized use case applications. Whilst at the same time assessing the viability of these mobile network realisations in their current state and development roadmaps. It is to be noted that at the time of writing Open Source 5G testbeds are still quite early in their development phase, and hence might not perform according to the theoretical standards and expectations.

Contents

List of Figures

Listings

List of Tables

Nomenclature

3GPP	Third Generation Partnership Project
AR/VR	Augmented Reality and Virtual Reality
BVLOS	Beyond Visual Line of Sight
CDMA	Code-Division Multiple Access
CN	Core Network
CNPC	control and non-payload communication
COTS	Custom Off The Shelf
D-GPS	Differential Global Positioning System
DIY	Do It Yourself
E-UTRAN	Evolved-UMTS Terrestrial Radio Access Network
eMBB	Enhanced Mobile Broadband
EPC	Evolved Packet Core
FAA	Federal Aviation Administration
FANETs	Flying Ad-Hoc Networks
FDMA	Frequency Division Multiple Access
GPRS	General Packet Radio Service
GPS	Global Positioning System
HSDPA	High Speed Downlink Packet Access
HSPA+	High Speed Packet Access plus
HSUPA	High Speed Uplink Packet Access
IoT	Internet of Things
KPIs	Key Performance Indicators
LOS	Line Of Sight
LTE	Long Term Evolution
MANETs	Mobile Ad-Hoc Networks
MIMO	Multiple Input Multiple Output

mMTC	massive Machine-Type Communication
NASA	National Aeronautics and Space Administration
NP	Non Public
NSA	Non-Standalone
OFDMA	Orthogonal Frequency Division Multiple Access
PRB	Physical Resource Block
QoS	Quality of Service
RAN	Radio Access Network
SA	Standalone
SACAA	South African Civil Aviation Authority
SC-OFDMA	Single Carrier Orthogonal Frequency Division Multiple Access
SMS	Short Message Service
sUAS	small Unmanned Aerial Systems
TDMA	Time Division Multiple Access
UAV	Unmanned Aerial Vehicles
UE	User Equipment
UMTS	Universal Mobile Terrestrial / Telecommunication Systems
URLLC	ultra-Reliable Low Latency Communication
uRLLC	ultra-Reliable Low Latency Communication
USIM	Universal Subscriber Identity Module
VANETs	Vehicular Ad-Hoc Networks

1 Introduction

This project reports on work done to implement and deploy an open-source 5G communication link for use on sUAS. This section outlines the context within which this work is done, the challenges being addressed and the key outcomes of the effort.

This is done by introducing the topic of interest in Section 1.1. Followed by outlining the gaps in current research using the information gleaned from literature on the field of research, thus stating the main purpose of this research. (Section 1.2). The scope and limitations are outlined in Section 1.4, this is followed by mentioning the contributions made by this research endeavour. Lastly, a report outline, (Section 1.6) showing the holistic structure of this report.

1.1 Background of the study

sUAS also known as drones have gained substantial recognition over the past few years. This is largely due to the numerous and evolving use cases that sUAS offer and are able to offer. Because of this interest and adoption sUAS technology has seen constant improvement and innovation at a rapid rate. sUAS have an enormous potential in enabling new applications in various areas ranging from military, security, medicine, and surveillance to traffic-monitoring applications [14]. Some more specific sUAS use cases include but in no way limited to aerial photography, express shipping and delivery, information gathering or supply of essentials for disaster management, thermal sensing for search and rescue operations, geographic mapping of inaccessible terrain, law enforcement and border control surveillance [15].

This research is motivated by drones specifically used for environmental data capture. In which these drones are equipped with suitable sensors such as carbon dioxide and nitrogen sensors to extract required data from the environment, and much of the time are flying autonomously. sUAS have been recognised as machines that bring about improved work efficiency, productivity and accuracy whilst simultaneously decreasing workload, production and labour costs [15]. For example, spraying crops on farmland may be done with a single drone, faster, with more efficiency and with minimum labour. As opposed to conventional methods such as using human labour to individually fumigate crop by crop.

The emergence of new wireless technologies such as high-bandwidth 4G and 5G networks further increases the capabilities of sUAS, especially those equipped with sensors producing large amounts of data. sUAS are now able to deliver Internet of Things (IoT) services into the airborne space. This is because of the services offered by these mobile networks such as real-time cellular-based command and control, BVLOS operation with areas serviced by the mobile network and the high bandwidth offered enables streaming and transmission of copious amounts of data.

However, despite the successful adoption of drones in the various sectors mentioned above, there are still several use cases to be realised. For example, the automated urban deliveries service which requires BVLOS. Also, real-time streaming of data such as 4K/8K, AR/VR requires high bandwidth, throughput and latencies achievable by the 5G technologies.

Several challenges need to be addressed for these new use cases to be realised and commercialised. Such challenges include security, privacy and management of aerial traffic [14]. Another pertinent problem to be solved is that the current terrestrial infrastructure being used for drone networks has many limitations. These include limited area coverage and management of the increasing densification of drones [16]. This research will focus more on the communications of sUAS. This includes researching the capabilities and limitations of the current UAV communications such as terrestrial communication, radio frequency, Wi-Fi and mobile networks i.e. 4G and the most recent 5G, which is yet to be ubiquitously deployed for commercial and personal use.

1.2 Purpose and Objectives of the study

Although the drone industry is still considered to be at the beginning in terms of maturity and application, it is reaching a bottleneck. Noting the background description above, a large contribution to this problem has to do with communication. The current and conventional communication systems in use are failing to meet the performance specifications required by these new use cases. Hence the need for alternative solutions arises.

The main purpose of this study is to deliver a proof-of-concept communication stack that will utilise Mobile Networks for sUAS telemetry and payload communication. Such a stack will provide insight into the feasibility of taking such an approach and whether or not it will be able to mitigate the problems faced by current drone users and the industry as a whole to make way for new use cases. It will provide useful insights both for researchers, Do It Yourself (DIY) users and even professional drone users, especially those looking for bespoke solutions to address non-common use cases.

Work has been broken into smaller objectives to be met so as to realise this purpose. Hence based on the background above and the main purpose of the study, this research endeavour is going to:

1. Provide a research-based review on the sUAS industry. This review will cover the usage of drones and their requirements. Discuss the current state of the drone industry with a specific emphasis on communication. This review will identify the current gaps and problems being faced and provide a possible solution to tackle these problems.

2. Design and implement a drone communication stack that utilises an open-source mobile network stack.

3. Test the architectural stack from the sub-system up to the system as a whole.

4. Carry out flight tests, using the proposed and implemented communication stack. Assessing its performance.
5. Compare the implemented solution to currently existing solutions in the commercial industry.
6. Make recommendations for further investigations
7. Provide full system documentation.

1.3 Research questions

With these objectives, some of the research questions to be answered are as follows;

- Whether the mobile network stacks to be implemented a viable communication option for not only hobbyists but also for commercial adoption.
- Does their performance fairs against other communication technologies that are available?
- What use cases can be currently supported by implemented open source mobile network stack?
- and whether or not this mobile network stack route that has been chosen can help realise new drone use cases

1.4 Scope and Limitations

This research will be limited to sUAS/drones only as defined in Section 2 and will not be concerned with other forms of drones. As well as utilising Non Public (NP) mobile networks, limited only to open source mobile networks and all equipment used and designed should be open-source and available to the general public. This allows research to be easily carried on from where this project concludes.

1.5 Research Contribution

At the time of writing, much research has been done in this area, but current results still leave a lot to be desired. The primary reason is that not much experimentation has been conducted, especially flight tests and the utilisation of open-source mobile networks for this application. [17] implemented a 4G LTE software stack for drone flights, but this is only capable of using commercial mobile networks.

This work implements a communication stack that can be used with any open source NP mobile network. The outcomes of this work have been presented to the Linux Foundation at the Embedded Linux Conference and can be found at [18]. Work regarding open-source mobile networks has also been presented to the 18th Wireless On-demand Network systems and Services Conference (WONS 2023) and a paper is to be published, currently in press [8].

1.6 Report outline

This section outlines the document's structure following the Introduction and provides a brief summary of each section's respective content.

Section 2, presents a narrative of both academic and consumer literature specific to the topic at hand. This section will explore the drone industry with respect to use case applications and then explore mobile networks. In doing so, addressing the gaps in research and literature and creating solution suggestions.

Section 3 will provide details on implementation. This section will provide details on all the equipment required, both software and hardware to see the project from start to finish. It also provides a detailed guide on how to implement the designed software stack and how the software stack itself came to be.

After the Implementation comes Testing (**Section 4**). This section lists all the tests to be carried out to assess performance. This includes indoor and outdoor flight tests.

Section 5 provides the results obtained from the various experiments carried out and discusses the key takeaways from the results.

Section 6 provides a conclusion to the project, outlining the key findings, as well as accomplishments and limitations.

This is followed up by Recommendations, **Section 6.1**, where suggestions for future work and further research are presented.

2 Literature Review

This section will start by providing insight into the past and current sUAS industry and research industry. As the topic at hand is more focused on the communication aspect of drones i.e. command, control and data transfer, this section will dive more deeply into this area, focusing on how the currently possible drone use cases have reached their plateau, the major reason for this being the fact that sUAS communication technology is becoming the bottleneck as the industry attempts to achieve a myriad of unrealised sUAS use cases. This section will cover the latest industrial/commercial implementations of sUAS communication. The latest research that has been done to realise and address the current communication problems being faced and advance the industry further. This section further provides the literature and technical insights to justify the use of Mobile Networks in the sUAS industry. And why leveraging Mobile Networks could possibly address the issues and limitations currently being faced in the sUAS industry with a specific focus on the advent of Fifth Generation (5G) mobile networks.

Before drones were commercialised, they were mostly used by the military for various purposes including Training, Airstrikes, Bomb detection and Target practice [19]. Their commercial and non-military use only began in 2006, the same year that the United States Federal Aviation Administration (FAA) issued its first commercial drone permit. The FAA also clearly defined the drone/(sUAS) in [20] as a model aircraft if:

1. the aircraft is flown strictly for hobby or recreational use;
2. the aircraft is operated in accordance with a community-based set of safety guidelines and within the programming of a nationwide community-based organization;
3. the aircraft is limited to not more than 55 pounds (25 kg) unless otherwise certified through a design, construction, inspection, flight test, and operational safety program administered by a community-based organization;
4. the aircraft is no less than 0.55lbs (250g) in mass;
5. the aircraft is operated in a manner that does not interfere with and gives way to any manned aircraft;
6. when flown within 5 miles of an airport, the operator of the aircraft provides the airport operator and the airport air traffic control tower (when an air traffic facility is located at the airport) with prior notice of the operation (model aircraft operators flying from a permanent location within 5 miles of an airport should establish a mutually-agreed upon operating procedure with the airport operator and the airport air traffic control tower (when an air traffic facility is located at the airport)); and
7. flown within the visual line of sight of the person operating the aircraft;

On top of these specifications, the are some local rules (in South Africa) to be followed as well, these have been outlined in [21] by the South African Civil Aviation Authority (SACAA). Amongst these are when and where a drone license is required.

This research is only concerned with sUAS (referred to as drone in the rest of this report) that meet the above specifications. Ground or water-based drones will not be the main focus of this paper, although this research is certainly transferable to those areas.

2.1 Drone use cases and requirements

Drones can have many purposes from military to non-military commercial and hobbyist use cases. This sub-section delves into the different drone use cases and their respective requirements (in terms of the technology both hardware and software) needed for such cases to be achieved. This sub-section will discuss both the realised and unrealised, currently theoretical use cases of drones.

Some of the drone use cases within the military include bomb detection, surveillance and airstrikes as well as target practice. Some commercial use cases began to be realised, these include drone delivery, drone filming and imaging, precision agriculture, search and rescue, and environmental data capture. Table I and Table II summarise the most common drone use cases and the needed requirements to realize these use cases. These requirements include data rates (Uplink and Downlink), GPS accuracy, end-to-end latency as well network latency. Some of the use cases mentioned are yet to be realised such as 8K video streaming and AR/VR this is due to communication bottlenecks and is further discussed in Section 2.2.

2.1.1 Data rates/throughput requirements

Data rate requirements can vary greatly depending on the use case and the drone application. For example, applications that require simple command and control such as precision agriculture or a drone fleet show need only 200-300Kbps both in the uplink and downlink. Whereas applications such as drone filming (4K and 8K) video require data rates of 60Mbps and 1Gbps respectively in the uplink direction. As shown in Table I. It is to be noted that, unlike most communication systems, drone data rate requirements are more dominant in the uplink direction. (i.e. The drone is sending information to some destination, or to the main computing node) [10]

2.1.2 Latency requirements

From Tables I and II it can be seen that much like the data rates requirements, latency requirements also vary depending on the use case application. In general, the more real-time the use case at hand, the more stringent the latency constraints will be. Real-time remote control requires an end-to-end latency of $\leq 100ms$ and a network latency of $\leq 20ms$. As compared to the precision agriculture use case where an end-to-end latency of 3000ms would do just fine.

sUAS Application	**Traffic Latency (ms)**	**Data rates (UL/DL)**
Drone delivery	500 ms	200 Kbps/300 Kbps
Precision agriculture		200 Kbps/300 Kbps
Search and rescure		6 Mbps/300 Kbps
Drone filming (4K)		30 Mbps/300 Kbps
Surveilance	3000 ms	10 Mbps/300 Kbps
Drone fleet show	100 ms	200 Kbps/200 Kbps

Table I: Typical drone use cases and their communication requirements. [9], [10]

2.1.3 Other requirements

Other than latency and data rates, Table II shows two more important connectivity requirements, namely Coverage and Positioning. Coverage in terms of area and coverage in terms of height. Applications such as upper air inspection or logistics and transportation (transporting medicine in and around a small town) require a communication channel that is capable of high coverage, (300-3000m), whereas spraying of agricultural chemicals in a small field would not require much coverage.

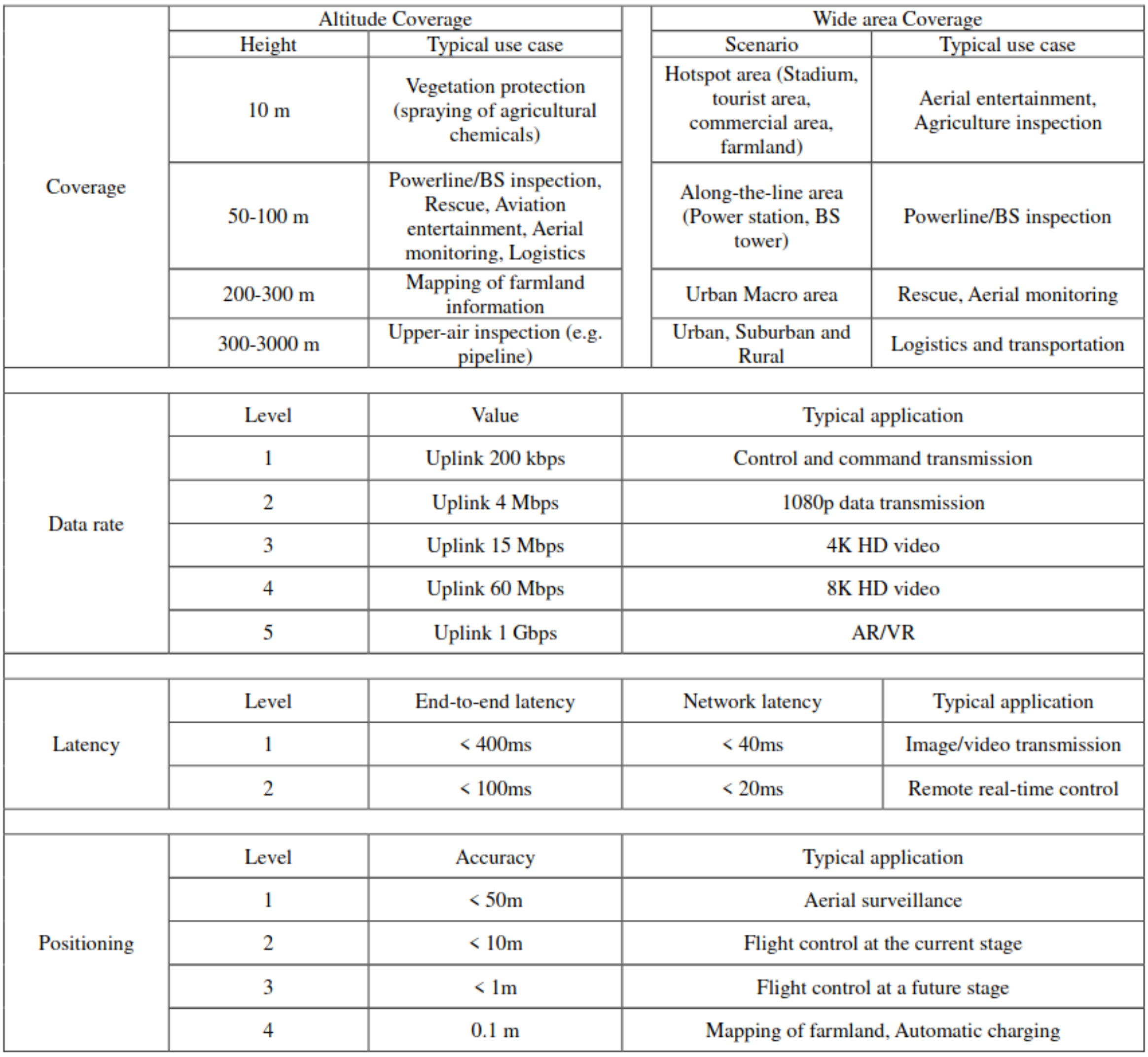
Positioning is also an important requirement, mostly dependent on the GPS location information. The requirements can be as tight as 0.1m position accuracy (e.g for automatic charging) or a more lenient 10m - 50m for applications such as aerial surveillance or flight control.

Guang Yang et al. and Xingqin Lin et al. in [22] and [11] respectively pointed out the current and future requirements of drones within the IoT space. And Lagkas et al. [14] and Mario et al. [16] also pointed out some requirements for UAVs in their research. These are summarised below.

- Remote and real-time control requiring low latency communication with robust navigation. Traditional Unmanned Aerial Vehicles (UAV) navigation uses the Global Positioning System (GPS), which is notoriously vulnerable to satellite signals, bad weather conditions or tall buildings and forestation. One solution to this is Differential Global Positioning System (D-GPS), described later in Section 2.5.
- Seamless coverage including BVLOS and HD image/video transmission and Global Positioning System (GPS) without range limitations. Use cases such as live video streaming from a drone to a distant audience across the globe require seamless coverage and the drone's access to a network at all times, also air traffic controllers would need

to know flight and location details of the drone at all times.

- Drone identification and regulation. With the increasing number of drones in the air space, monitoring and management become critical. With appropriate regulations and legislation in place, authorized parties, such as air traffic controllers need the ability to identify rogue drones and take over control, to avoid any foreseen safety threats.
- Interference management. The airspace is becoming increasingly populated, IoT devices are also increasing, and so is interference in the airspace. All this interference must be handled appropriately for safety and security reasons.



Coverage	Altitude Coverage		Wide area Coverage	
	Height	Typical use case	Scenario	Typical use case
	10 m	Vegetation protection (spraying of agricultural chemicals)	Hotspot area (Stadium, tourist area, commercial area, farmland)	Aerial entertainment, Agriculture inspection
	50-100 m	Powerline/BS inspection, Rescue, Aviation entertainment, Aerial monitoring, Logistics	Along-the-line area (Power station, BS tower)	Powerline/BS inspection
	200-300 m	Mapping of farmland information	Urban Macro area	Rescue, Aerial monitoring
	300-3000 m	Upper-air inspection (e.g. pipeline)	Urban, Suburban and Rural	Logistics and transportation
Data rate	Level	Value	Typical application	
	1	Uplink 200 kbps	Control and command transmission	
	2	Uplink 4 Mbps	1080p data transmission	
	3	Uplink 15 Mbps	4K HD video	
	4	Uplink 60 Mbps	8K HD video	
	5	Uplink 1 Gbps	AR/VR	
Latency	Level	End-to-end latency	Network latency	Typical application
	1	< 400ms	< 40ms	Image/video transmission
	2	< 100ms	< 20ms	Remote real-time control
Positioning	Level	Accuracy	Typical application	
	1	< 50m	Aerial surveillance	
	2	< 10m	Flight control at the current stage	
	3	< 1m	Flight control at a future stage	
	4	0.1 m	Mapping of farmland, Automatic charging	

Table II: Table showing connectivity requirements for different use cases [11]

Table I and II above depict drone applications that have been realised to date and, the latest drone technologies can perform most of these applications. However, drones are capable of a lot more. Many drone applications are yet to be realised or fully realised, some more novel than others. These include using drones for

- emergency medical supplies both in remote areas and building-dense areas such as cities and towns. [23],
- providing opportunistic networks and internet access in remote areas or dead zones [24],
- environmental data capture and agriculture [25], this can be within the atmosphere or even underwater. [26]
- transporting people and objects, this could take 5 to 15 years to mature. and fully realised [27]
- long-range surveillance requiring beyond visual line of site command and control and autonomous capabilities.
- AI based solutions in relation to UAVs including predicting the demands and adjust cell designs to meet the users requirements. [28], [29]

The use cases above are yet to be fully implemented and mature. The drone industry is already working towards such a goal. At the moment there are several bottlenecks to be overcome before this goal can become a reality. Such bottlenecks include but are not limited to, as stated by [27] regulation, infrastructure, technical capabilities, public acceptance and economic drivers. Lagkas in [14] also further states that privacy and data security challenges need to be addressed. To realise more of the novel use cases the drone industry would need to be realised as part of the IoT as much of the use cases are IoT related. With this in regard Lagkas et al. [14] also mentions energy conservation and reliability, vandalism and weather challenges to be overcome. Other important challenges include [9] the size, weight and power (SWAP) challenge which is common within IoT devices. The Line Of Sight (LOS) usage of current small UAVs is also a major constraint in achieving drone applications that require BVLOS. Most of these limitations are driven by both legislation and communication link limitations. Addressing the latter is one of the key components to realising eased legislation.

Some but not all of the above challenges are common reasons behind the unrealised drone applications, one important challenge, which is the focus area for this project is communication limitations. Such communication challenges include latency issues, data rate limitations, and LOS communication limitations (coverage area). This will be further discussed in Section 2.2. Most of these limitations are driven by both legislation and communication link limitations. Addressing the latter is one of the key components to realising eased legislation.

2.2 Drone Communication

This sub-section pertains to wireless communication for sUAS, it will discuss the currently used communication systems and technologies for drone command and control as well their predecessors and possible future communications for sUAS to realise new and novel applications. sUAS communication can be grouped into two main categories. The first is mission-critical information to ensure safe and reliable flight and relay this information to several stakeholders such as air traffic control and nearby vehicles. This communication is known as [30] control and non-payload communication (CNPC). The other is payload communication, in which data such as image, video streams and sensor data being relayed to the ground control station and sUAS pilots. The Third Generation Partnership Project (3GPP) has specified the communication requirements for these two types of links shown in Table II adapted from [9]. 3GPP is made up of seven telecommunications standard development organisations, they govern and set all the standards for cellular telecommunications technologies.

Technology	**Description**	**Advantages**	**Disadvantages**
Direct Link	Direct point-to-point communication with ground station or controller	Simple, low cost	Limited range, low data rate, vulnerable to interference and signal jamming, non-scalable
Satellite	Communication and Internet access via satellite	Global coverage	Costly, heavy/bulky/energy consuming (SWAP limitations for sUAVs)communication equipment, high latency, large signal attenuation
Ad-Hoc Network	Dynamically self organizing and infrastructure free	Robust and adaptable, support for high mobility	Costly and time consuming to setup, low spectrum efficiency, intermittent connectivity, complex routing protocol
Cellular Network	Enables sUAS communications by using cellular infrastructure and technologies	Almost ubiquitous accessibility, cost-effective, superior performance and scalability	Unavailable in remote areas, potential interference with terrestrial communications. Costs money to use Alitude dependant performance

Table III: Summary of different UAV communication modes for UAVs [9]

[9] specified four of the most dominant communication technologies in the sUAS industry. Namely 1) direct link; 2) satellite; 3)Ad Hoc network; and 4) cellular network. Direct link communication uses a controller/joystick communication with the drone over the ISM (International Scientific Medical) band of 2.4GHz and 5.8 GHz [31]. The latter is allotted to reduce interference from the several devices already using the 2.4GHz band such as computers and WiFi. The utilised LOS limitation disqualifies this kind of communication as it is not scalable for large-scale deployments of sUAS and cannot be used to realise applications requiring BVLOS. Satellite communication is advantageous in terms of coverage however it comes with the largest delays making it unsuitable for ultra-Reliable Low Latency Communication (URLLC) and unsuitable for relaying delay-sensitive information.

Mobile Ad-Hoc Networks (MANETs) (Mobile referring to its mobility i.e different from the cellular network) can be split into two categories [9] Vehicular Ad-Hoc Networks (VANETs) and Flying Ad-Hoc Networks (FANETs). These are infrastructure-free and dynamically self-organizing networks for enabling peer-to-peer communications among mobile devices, such as laptops, and walkie-talkies without the need for any routers or central nodes. Such devices usually communicate over bandwidth-constrained wireless links using, e.g., IEEE802.11 a/b/g/n [9]. Ad-hoc networks can be set up using wifi, Bluetooth, a cellular network or even a cluster of telemetry radios. This form of communication is costly, especially in terms of time with the need to set up complex and bespoke routing protocols. Due to the lack of central computing/networking infrastructure, each node acts as a router and host. This increases the difficulty in implementation. The frequency spectrum limitation also greatly influences the capabilities of this sort of communication. This form of communication does not look promising for the future as it is difficult to scale and it also has coverage limitations.

The final mentioned technology, Cellular/Mobile Networks (different from MANETs as they utilise the cellular infrastructure, devices on this network are not self-organising and do not act as both router and hosts) seems to be the most promising technology to help realise URLLC and BVLOS. It has almost ubiquitous accessibility, is cost-effective, and is scalable for many drone deployments. To this end, the need for BVLOS usage of drones, the FAA and National Aeronautics and Space Administration (NASA) have initiated a collaborative effort, with the main area of focus being the safe control of UAVs for BVLOS use cases. [22] This ignited the consideration of using mobile networks for drone communication.

2.3 Mobile Networks

This sub-section will narrow down the research to Mobile Networks in particular. It will start by outlining the historical evolution of mobile networks, from the 1st Generation to the 5th Generation. The 4th and 5th Generation are of most notable interest for this research. Most importantly highlighting and drawing out the significance of each with respect to the main topic at hand, which is drone communications, both CNPC and payload communication.

The evolution of Mobile Networks

Since their conception and first deployment of mobile networks back in the early 1980s, mobile networks have seen a lot of changes and evolution over the years. This is mainly due to the increased demand for and adoption of such a technology. Standards had to progress quickly to accommodate the increase in users [32]. Starting with 1G, the characterising protocol being Frequency Division Multiple Access (FDMA). The base stations were circuit-switched and came with many limitations. There was no provision for encryption and no roaming facilities. The radio signals used by 1G were analogue which enabled a lot of background noise. [33] 1G only had support for one user per frequency channel, and frequency channels needed to be separated to avoid interference, this poor frequency utilisation meant that the increasing number of users could not be accommodated for [34].

The second generation mobile networks (2G) brought about some improvement, in addition to wireless voice capabilities 2G introduced Short Message Service (SMS) and were later enhanced with General Packet Radio Service (GPRS) for data communication [35]. The adoption of Time Division Multiple Access (TDMA) technologies allowed for better frequency utilisation as multiple users could be accommodated on one channel [34]. The introduction of Code-Division Multiple Access (CDMA) further improved frequency utilisation.

Following 2G, Third Generation Mobile Networks (3G) brought about a vast improvement in data rates, spectrum usage and the number of services offered. The commercial introduction of Universal Mobile Terrestrial / Telecommunication Systems (UMTS), which saw a data rate of 384kbps and support for video calling for the first time on mobile devices. [36] Further technological improvements such as High Speed Downlink Packet Access (HSDPA), High Speed Uplink Packet Access (HSUPA) and High Speed Packet Access plus (HSPA+) further improved the quality of these services offered [36,37]. The latter evolved to the more powerful LTE, bringing about the advent of Fourth Generation Mobile Networks (4G). Table IV shows the main differences and the evolution of mobile networks.

Technology ⇨ **Feature** ⇩	**1G**	**2G**	**3G**	**4G**	**5G**
Start/ Deployment	1970 – 1980	1990 – 2004	2004-2010	Now	Soon (probably 2020)
Data Bandwidth	2kbps	64kbps	2Mbps	1 Gbps	Higher than 1Gbps
Technology	Analog Cellular Technology	Digital Cellular Technology	CDMA 2000 (1xRTT, EVDO) UMTS, EDGE	Wi-Max LTE Wi-Fi	WWWW(coming soon)
Service	Mobile Telephony (Voice)	Digital voice, SMS, Higher capacity packetized data	Integrated high quality audio, video and data	Dynamic Information access, Wearable devices	Dynamic Information access, Wearable devices with AI Capabilities
Multiplexing	FDMA	TDMA, CDMA	CDMA	CDMA	CDMA
Switching	Circuit	Circuit, Packet	Packet	All Packet	All Packet
Core Network	PSTN	PSTN	Packet N/W	Internet	Internet

Table IV: Table showing a summary of the evolution of Mobile Networks [12]

2.4 Fourth Generation Mobile Networks and 4G LTE

To meet the exponentially growing demand for mobile network services, the introduction of 4G came with many changes. The architectural design became all-IP based, all traffic was now separated between the control plane and the user plane and the adoption of a breakthrough Orthogonal Frequency Division Multiple Access (OFDMA) [38]. OFDMA brought about a major improvement in spectral efficiency, radically lower latency and improved network congestion.

2.4.1 4G Network Architecture

The 4G architecture is made up of two main elements, the Core Network (CN)) commonly known as the Evolved Packet Core (EPC) for LTE systems, and the Evolved-UMTS Terrestrial Radio Access Network (E-UTRAN) which is composed of the Radio Access Network (RAN) and the User Equipment (UE). Shown in Figure 2.1 is the simplified 4G network structure.

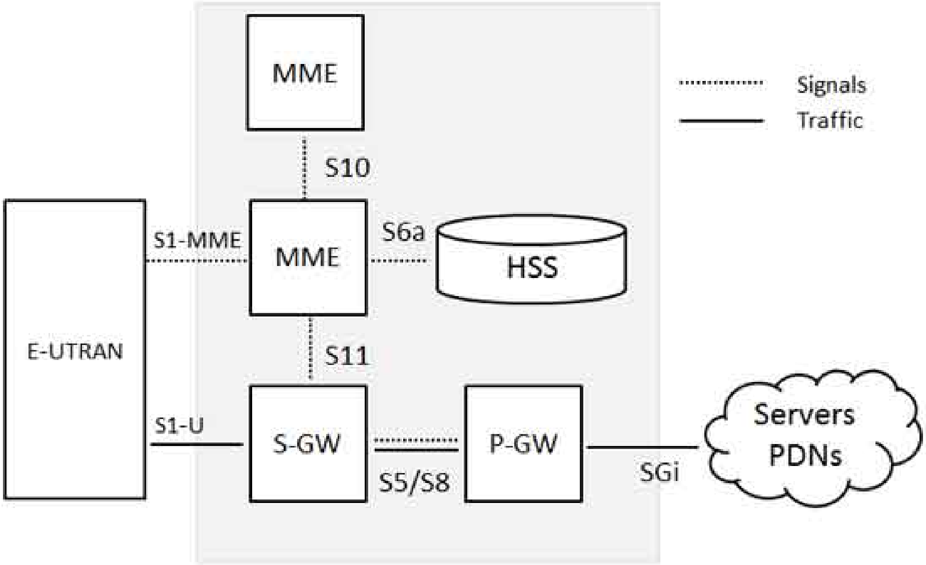


Figure 2.1: 4G LTE Network Architecture [1]

The EPC is made up of the following components, each with its own functions:

- Mobility Management Entity (MME)
- Serving Gateway (S-GW)
- Packet Data Network Gateway (PDN) Gateway (P-GW)
- Home Subscriber Service (HSS)
- Policy Control and Charging Rules Function (PCRF)

The **MME** is responsible for the following functions [39, 40]:

- Authentication: enables UEs to authenticate to the network by exchanging subscriber information between the UE and the HSS.
- Mobility management: allows the subscriber's mobility within the network or across networks.
- Location update: keeps track of the subscriber's location within the network, and of its current state.
- Bearer establishment: establishes bearers by deciding on a gateway router to the Internet if more gateways are available.
- Handover support: enables handover between eNodeBs (for handover on the S1 interface)

The **S-GW** is an anchor for handovers between neighbouring eNBs and routes all the user data packets, it also performs replication of user traffic in the special case of lawful interception and maintains a UEs context in the case of idle UEs [39, 40]. The **P-GW** is responsible for giving UEs access to external networks such as the greater internet, or other networks connected to the EPC.

The **HSS** handles all UE subscriber details and acts as a database for these. On top of this, it also handles user authentication, access authorization, and call session establishment support.

The **PCRF** is a combination of the Charging Rules Function (CRF) and the Policy Decision Function (PDF) and takes care of the Quality of Service (QoS) information. [39]

The **E-UTRAN** is comprised of the eNBs and UEs. Mobile network-capable devices such as mobile phones and tablets are all UEs. The **eNBs** are responsible for [39] radio resource management, packet compression and ciphering as well as message scheduling and transmission.

2.4.2 LTE for sUAS

Lakgas et al., Lin et al. and Yang et al. in [11, 14, 22] have shown agreement that mobile networks can indeed benefit the drone industry and gone on to state how and why the drone industry would benefit from using mobile networks to meet these future requirements. Their findings are listed below.

- Improved coverage enhancements. Thanks to the almost ubiquitous accessibility of cellular networks worldwide, cellular-connected drones make it possible for the ground pilot to remotely command and control the drone with essentially unlimited operation range.

- High data rates with latency and improved bandwidth. This comes with the effective use of available frequency bands. The OFDM protocol can deliver high data rates whilst significantly reducing latency.
- Detection and Resource Management. Detecting rogue drones. Mobile Network operators may provide legal rogue drone detection services to assist with drone traffic management, law enforcement, or other business purposes such as insurance.
- Mobile technologies are based on standards and are evolving, facilitating inter-operability based on industry-wide consensus and are improving with backward compatibility. [11]
- Cost-effectiveness. Reusing the already deployed mobile base stations removes the need for deploying new infrastructure, thus saving up on deployment costs. On the other hand, it may also help save operational costs, via bundling UAVcommand and control and other numerous types of payload communications into cellular systems, which will create new business opportunities for both cellular and UAV operators. Thus, cellular-connected UAVs are conceived to be a win-win technology for both cellular and UAVindustries, which may help facilitate the integration of sUAS into the National Airspace System (NAS) cost-effectively. [41]

Following the requirements set out in Table I and Table II. Guang Yang et al. [11] carried out performance analysis experiments and simulations with 4G LTE to ascertain its capabilities in terms of its Signal-To-Interference plus Noise Ratio (SINR), uplink throughput and latency, the results from these experiments were of notable interest. [11] noted that although the current 4G LTE advanced networks targeting terrestrial usage can already support the initial deployment of drones, these would not be able to support the increasing use case requirements. This conclusion is supported by results from experimentation. With reference to Table II the measured LTE advanced network could not meet the transmission rate requirement for HD video ($\geq$10Mbps). The network also had difficulty providing ubiquitous coverage for drones flying above 100m, this is because as the height increases propagation becomes more and more like LOS. Without further enhancements, the LTE-Advanced network would have difficulty meeting the stringent end-to-end latency requirement of 100ms.

These results proved that the 4G LTE still needs to be improved and upgraded to better serve these cellular-connected drones. [11] Fifth Generation mobile networks provide these upgrades, discussed in the following Sub-Section.

2.5 Fifth Generation mobile networks (5G)

Being the successor of 4G LTE mobile networks, 5G brings about a lot of enhancements and upgrades, new architectural designs, ultimately bringing forth a mobile network that has proved (through theory, simulations and experimentation) to be capable of advancing drone communications to realise new use cases such as BVLOS flights, 8K video streaming. The emergence of Internet of Things (IoT) devices and the increasing number of high-bandwidth applications has been the major agent responsible for the radical shift in technologies used in 5G compared to the previous generations.

This section will highlight the new advancements that 5G mobile networks offer, how they work and what makes them capable of seeing more and more drone applications to maturity. As well as comparing this technology with the previous 4G technology.

There are several new technological advancements that the Fifth Generation New Radio (5G NR) brings forth to mobile networks that the drone communication area can benefit from. Such as massive bandwidths, having a large number of antennas, and with an extreme base station and device densities [42]. These are further elaborated as follows:

- **Network Slicing** 5G introduces three main communication scenarios, these are (**ultra-Reliable Low Latency Communication (uRLLC)**), (**massive Machine-Type Communication (mMTC)**) and (**Enhanced Mobile Broadband (eMBB)**) [43]. Network slicing provides different Quality of Service (QoS) levels to end users depending on the end users' credentials, needs and use cases. This is done by creating a virtual network (i.e. Network slice) to run on top of the shared physical network infrastructure. In the UAV case, a drone would be allocated a slice relevant to its most dominant use case requirements, for example, a drone delivering medical supplies could be given a secure uRLLC slice as low latency communication, in this case, is more important than high data rates and massive-machine-type communication. Figure 2.2 shows some example slices that can be allocated according to use case requirements.

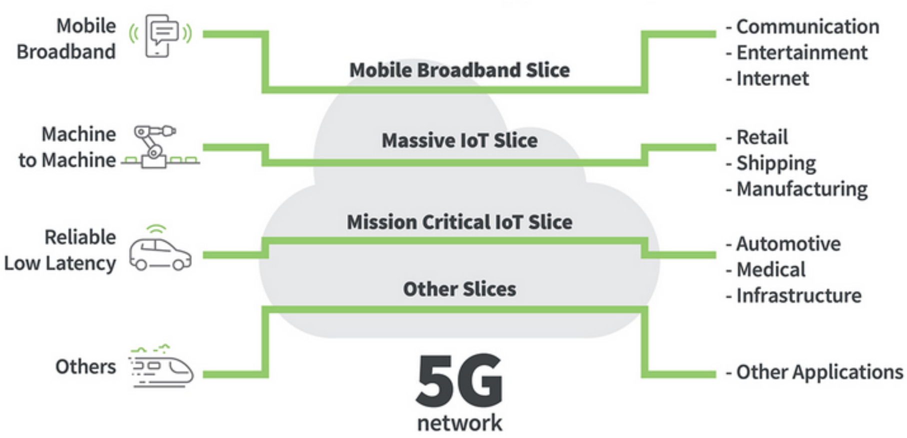


Figure 2.2: Network slice and use case examples [2, 3]

- **Flexible Numerology** As shown in figure 2.3, with 5G sub-carrier spacing is no longer limited to 15kHz like in its predecesor 4G. A resource block (RB), later referred to in this paper as a Physical Resource Block (PRB) is the smallest unit of resources that can be allocated to a user. [44] By exploiting the sub-carrier spacing 14 OFDM symbols can be transmitted in less and less time depending on the numerology used which can be 0,1,2 or 3, representing sub-carrier spacings of 15kHz, 30kHz, 60kHz or 120kHz respectively. Equation 1 describes this where μ is the actual number referred to in the word numerologies. These flexible numerologies are the enabler to realise uRLLC mentioned above and better frequency spectrum utilisation.

$$\Delta f = 15kHz * 2^{\mu} \tag{1}$$

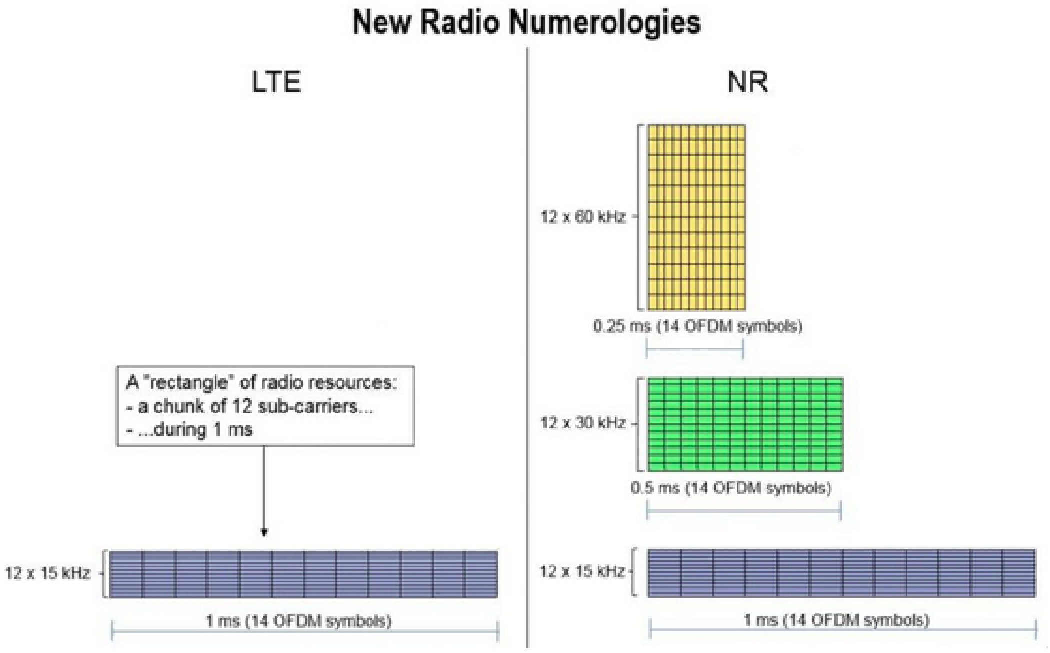


Figure 2.3: 5G Flexible numerology [4]

- **3D Beam-forming** [5, 41] To enhance the signal strength between an open communication channel from 5G network to UE, 5G mobile networks offer a new technology for mobile networks called beam-forming to address the need for a robust low attenuation channel as a drone manoeuvres in the 3D space [43].

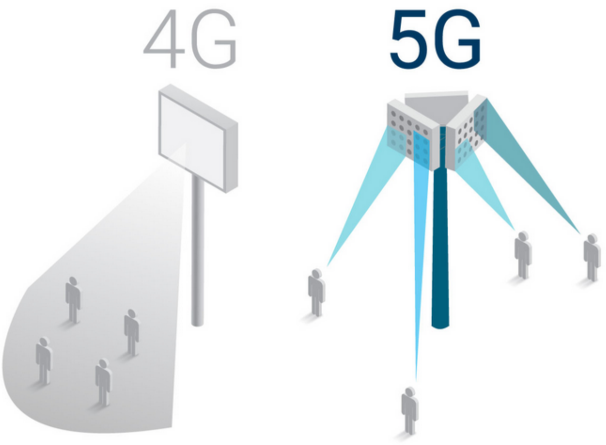


Figure 2.4: 5G Beam-forming signal processing technique [5]

Figure 2.4 shows the new 5G Antenna technologies and how it differs from their predecessors. Multiple Input Multiple Output (MIMO) arrays with thousands of small antennas combined in a single formation can be used to evaluate the most effective transmission path to each end-user unit. By exploiting interference, these arrays can form a more direct beam towards the end user and carry out beam-steering and beam-switching techniques [5].

- **Asymmetric Uplink/Downlink traffic rates** [41] Unlike the current mobile networks, which are more downlink traffic rate dominant (this being from the base station to UE(drone) direction), a cellular-connected UAVis more concerned about the uplink traffic rates. (This being from UE (drone) to the base station/ground control station) This is because payload communications such as video streaming can become resource-demanding. Since 5G utilises a wide frequency spectrum, new technologies can be developed to address this unique requirement more efficiently. For example, [41] suggested using drastically different bands for uplink and downlink communications, such as the conventional sub-6 GHz for UAVdownlink whereas the largely under-utilized millimetre wave (mmWave) spectrum for UAVuplink.
- The 3GPP framework allows **UAS Traffic Management** (UTM) to communicate with the UAS and allows approved tenants to query UAVidentity and UAS meta-data, e.g. for safety and security reasons. [45]
- **Robust Navigation**. As mentioned earlier traditional sUAS use GPS for navigation which is highly susceptible to interference. Cellular-connected drones offer one effective method to mitigate this. By using [41] differential GPS (D-GPS), to achieve more robust UAV navigation by utilizing cellular signals as a complementary for GPS navigation.

The deployment of 5G Networks can take either one of two forms. The first is the NSAt, and the second is the SA. These deployments vary in performance, and the structure and architectures used, for example, Core Networks and Base Stations deployed are different and so are the technologies used to implement these. These are explored further in the upcoming sections.

2.5.1 5G Architecture: Non-Standalone (NSA)

Th 5G non-standalone architecture is not based entirely on 5G specifications as it still uses the 4G EPC. That being said everything else besides this is based on 5G standards including the base station, called gNodeB (gNB) which is the equivalent of the eNB in a 4G network and these are used in tandem. In an NSA 5G RAN, you won't get the much-touted 5G capabilities such as near-zero latency and unparalleled speed, but it's a cost-effective way to deploy a 5G network across the globe. [46]

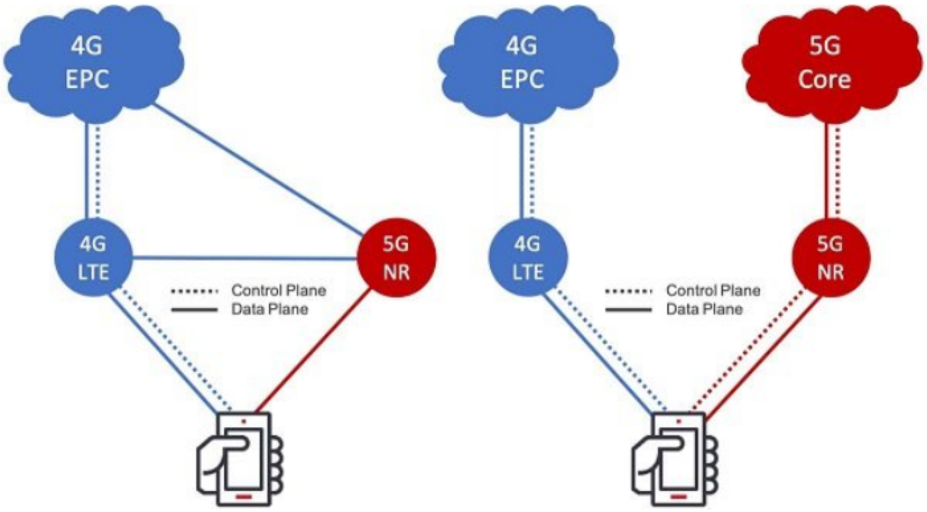


Figure 2.5: 5G NSA *(left)* and SA *(right)* architectures [6]

Figure 2.5 shows the most common 5G NSA architecture which is also the one being currently deployed by commercial mobile networks [46]. The control plane is entirely on 4G standard, and only when a 5G capable UE has successfully attached to the network (via the 4G eNB) is it then transferred to the 5G base station, i.e. the gNB. This 5G connection is only responsible for data plane transport (using 5G standards) but still makes use of the 4G EPC.

Some advantages of 5G NSA are as follows [46]:

- Since it still makes use of the EPC, deploying NSA does not require a complete overhaul, hence it is the quickest and simplest way to deliver 5G.
- It still offers significant data rate improvement as it adopts the new 5G standards and the gNB.
- With Dynamic Spectrum Sharing (DSS), 4G and 5G bands can simultaneously share their spectrum to offer an overall better experience.

2.5.2 5G Architecture Standalone (SA)

5G standalone is a complete shift from 4G, SA essentially means the whole end-to-end network is entirely different and built on 5G specifications. With an entirely new core network (called the 5G Core) the new SA architecture brings about all the technological advancements described above in Section 2.5.

The 5G Core comprises several new network functions/components, each responsible for their specific service, hence the common name, service-based architecture (SBA). These are shown in Figure 2.6.

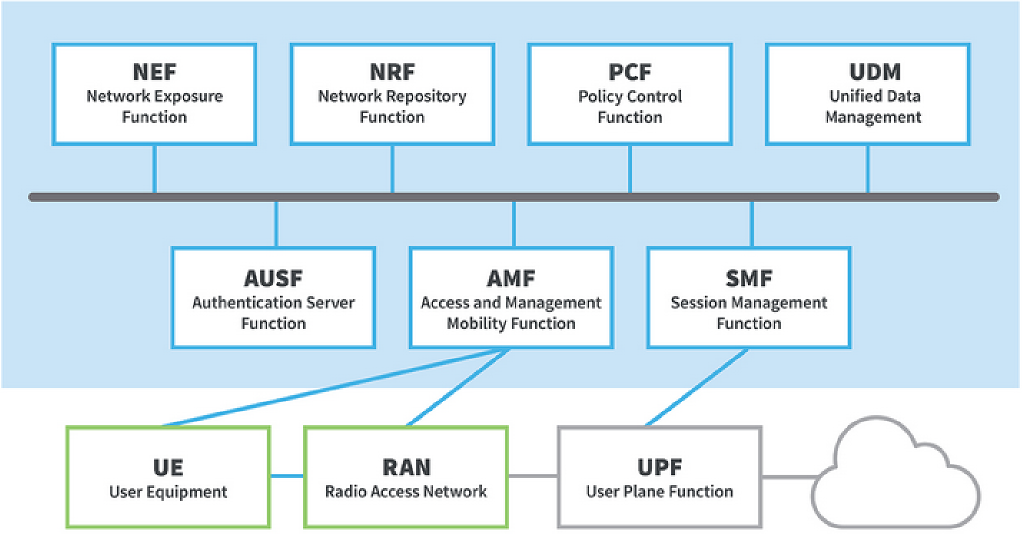


Figure 2.6: 5G Core Network: Service based architecture [3]

The Core Network functions are described below [3]:

- The **Access and Mobility Management Function** (AMF) acts as a single-entry point for the UE connection.
- Based on the service requested by the UE, the AMF selects the respective **Session Management Function** (SMF) for managing the user session.
- The **User Plane Function** (UPF) transports the IP data traffic (user plane) between the User Equipment (UE) and the external networks.
- Other functions like the **Session Management Function** (SMF), the **Policy Control Function** (PCF), the **Application Function** (AF) and the **Unified Data Management** (UDM) function provide the policy control framework, applying policy decisions and accessing subscription information, to govern the network behaviour.

However, it is worth noting that true 5G (5G SA) still needs time to mature. As stated above almost all current commercial deployments of 5G are 5G NSA, based on the 4G EPC. (For example in South Africa, the service provider MTN, uses the deployment provided by Ericsson which NSA) As this is the most cost-effective way of delivering 5G. Reasons being that deploying 5G SA requires completely new base stations and Core Networks. This usage of higher frequencies does not make this easier, as such frequencies are more susceptible to path loss and interference due to more base stations per unit area would be required which increases the deployment costs The costs will decrease over time, but many challenges must be overcome before 5G SA can be truly deployed everywhere.

2.6 Open Source

Following a deeper dive into mobile networks as the most suitable means of communication for this research project, it is important to find ways of working with mobile networks. The cost of implementing your own network could be very high, and acquiring access to an already deployed mobile network was not possible. With that in mind, this section looks at open-source mobile network implementations currently used and developed by both the industry and researchers.

Several open-source mobile networks have been implemented to date. Some organisations mainly focus on the development of the Core Network eg MagmaCore [47], free5GC [48] and open5GS [49]. Other implementations include OpenAirInterface (OAI) [7] and SRS RAN [50], these implement a full end-to-end solution. Ideally, all the open source implementations should be fully interoperable and easy to integrate out of the box as they are all governed by 3GPP specifications and are all currently implemented using Release 16 compliance. However, in reality, this is not the case. Interoperability might require some bit of low-level tweaking for a working end-to-end solution. Looking at all their development timelines, OAI and SRS had managed to implement a working solution for 5G NSA and at the time of reading, they had just recently started developing their 5G standalone software stacks.

Going forward in this research OAI was chosen to carry out the research project, as it was the furthest in development timelines and currently had the best performance and most robust network by comparison. SRS RAN is more lightweight when compared to OAI which requires more computing resources to run, but OAI has the advantage of accessibility. It provides the user access to its low-level source code, enabling the user to tweak and adjust the setup for their specific deployment for the best performance.

2.7 Summary

This section provided insight into the literature covered for this research. Beginning with a formal definition of the drones that will be the focus of this research. It then went on to discuss some of the drone applications and the requirements to meet these applications and went on to further focus on these requirements in terms of communications as it is the main area of focus for this research. Researching the most common conventional communication technologies available for drones and taking into account the new drone applications that are yet to be realised and bottle-necked by current communication technologies. Exploration into mobile networks was carried out, specifically 4G and 5G. The main takeaway from this research in literature was that 4G LTE technology is able to cover some but not all of these requirements to meet the potential and future drone applications, but the advent of 5G, and the myriad of new technological advancements that it provides, these new and some novel applications can be realised.

Although the ubiquitous deployment of 5G is yet to come. It is worthwhile researching how these new technologies can benefit the drone industry. To this end, research was carried into the currently available open-source mobile network implementations to carry out this research. OpenAirInterface was found to be the most suitable, and hence will be used further with research.

3 Implementation and practical work

This chapter aims to provide, in detail an insight and reasoning towards the applications, tools and tasks carried out to realise the implementation of the project from start to completion. It will show the installation, setup and usage of tools used in the order that they were implemented for the project. This section will start by stating the needed equipment, both software and hardware needed to realise an end-to-end communication stack that utilises a private mobile network implementation for drone telemetry transmission. This is described in Section 3.1. Afterwards, this section goes on to show the mobile network test-bed setup and how to implement them, both the 4G LTE and 5G NSA. (3.3, 3.4). Following the test-bed implementation, this section goes on to describe how to ultimately tie everything together, thus the implementation of a software stack to achieve end-to-end (drone to the ground station and back) communication. This final architecture is shown below, Figure 3.1. The process of reaching this final stack is described in this Chapter.

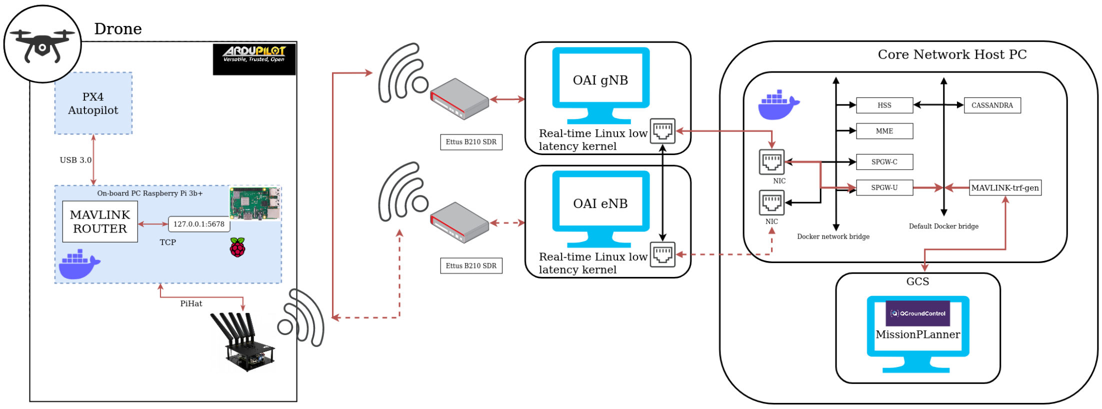


Figure 3.1: Drone to ground station connection

3.1 Prerequisites: Equipment and Software

This section outlines the equipment software and hardware used to realise an end-to-end communication stack that utilises both a private mobile network for drone command and control. These then are the requirements needed to build and implement such a test bed by following this guide. All software and hardware to be used in this are entirely open-source.

3.1.1 Mobile Network test-beds

The OpenAirInterface RAN project[1] is going to be used to implement a private and local Mobile Network. It comprises of both the Core Network and RAN software required to run

---

[1] https://openairinterface.org/

the SDRs (Software Defined Radios) It will be used to realise both the 4G LTE and 5G NSA mobile network.

3.1.2 Software Defined Radios

A software-defined radio (SDR) is a radio communication system that uses software to process various signals by carrying out modulation, demodulation, decoding etc. as opposed to using dedicated hardware (e.g. amplifiers, duplexers). In this case, an NI Ettus B210 SDR[2] is used to realise the Radio Access Network (RAN) and is being operated by the OpenAirInterface RAN software. This SDR can operate across frequencies from 70 MHz to 6GHz, with full duplex MIMO with up to 56 MHz of real-time bandwidth which is more than adequate for the use case at hand.

3.1.3 Antennas

Antenna choice is an important decision to be made as they need to be able to operate at the required frequencies for the respective 4G and 5G operating bands. All antennas with an operating frequency range between 800MHz and sub 6GHz should work fine. Specifically the frequency range (MHz) 1710-2700. Ideally, antennas designed for mobile network usage would be preferred. The Space TV[3] omnidirectional antenna is to be used since it is suitable for outdoor usage, which would prove handy for drone flights.

3.1.4 Host Machines (Core Network and RAN)

OpenAirInterface states the PC specifications for a smooth operation of their stack. The RAN component is more computationally expensive and requires the machine to be running Ubuntu on the low latency kernel, which is better suited for carrying out real-time operations. Minimally a 1.8GHz processor and 8GB RAM are required.

3.1.5 Programmable USIM cards

Programming USIM cards play an important role in the realisation of an end-to-end mobile network. Before a COTS UE can attach to the EPC via the eNB, it is required to have a SIM card with the correct authentication parameters. The MME handles [51] security procedures (i.e. user authentication, ciphering and integrity protection), hence for a COTS UE to attach to the network it should have with it have a USIM card with the correct authentication parameters as those described by and stored in the HSS database.

5G NR capable USIM cards are required, and a SIM card reader/writer is also needed to program the USIM card with parameters that match those defined in the Core Network. The

---

[2] `https://www.ettus.com/all-products/ub210-kit/`

[3] `https://www.spacetv.co.za/`

sysmoISIM-SJA2 USIM cards were used, these were sourced from Sysmocom [4]. They are re-programmable UISM cards and hence can be reconfigured an unlimited number of times, they are also capable of 5G NR. The programming software used was the open-source PySim [5] which is the most compatible with Sysmocom SIM cards. Another alternative to PySim is GRSIMWrite which does not require the use of ADM keys, but this comes at the cost of limited programming options. The Rocketek smart card reader was used to connect, read and write the USIM card.

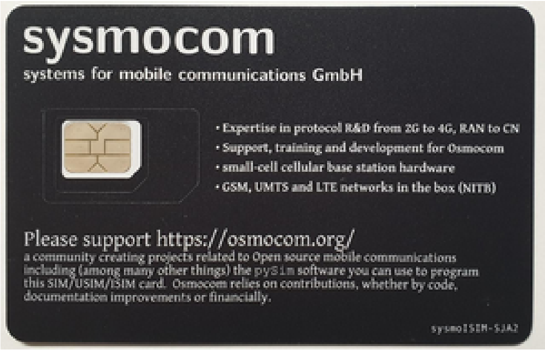


Figure 3.2: sysmoISIM-SJA2 USIM card provided by Sysmocom

Figure 3.3: Rocketek smart card reader

```
~/pysim$ ./pySim-prog.py -p 0 -t sysmoISIM-SJA2 -a 46923892 -x 655 -y 25
-i 655250000000002 -s 8988211000000498439 -o efb5c7f1329f031d72760e7fd95a776a
-k efbfbdefbfbd473f2fefbfbdd094efbf -n openairinterface

% Output
Using PC/SC reader interface
Ready for Programming: Insert card now (or CTRL-C to cancel)
Generated card parameters :
 > Name      : openairinterface
 > SMSP      : e1ffffffffffffffffffffffff0581005155f5ffffffffffff000000
 > ICCID     : 8988211000000498439
 > MCC/MNC   : 655/25
 > IMSI      : 655250000000002
 > Ki        : efbfbdefbfbd473f2fefbfbdd094efbf
 > OPC       : efb5c7f1329f031d72760e7fd95a776a
 > ACC       : None
 > ADM1(hex): 3436393233383932
 > OPMODE    : None
Programming ...
Warning: Programming of the ICCID is not implemented
```

Listing 1: Simcard configuration parameters

---

[4] https://sysmocom.de/index.html

[5] https://github.com/osmocom/pysim

Shown in Listing 1 parameters of main importance that have been programmed are as follows:

- International Mobile Subscriber Identity (IMSI)
- Authentication Key (KI)
- Operator Code (OPC)
- Service Provider Name (SPN)
- Authentication Algorithm

3.1.6 COTS UEs

For this project, two COTS UEs were used. The Huawei P40 lite 5G mobile phone and the Waveshare 5G Pi Hat[6] accompanied by a SIM8200EA-M2 SIM card module. Any other COTS UE that is 5G capable should work equally fine, the SIM8200EA-M2 and the Quectel RM500Q-GL SIM card modules would be preferred as OpenAirInterface is developing and testing their stack with these two in mind as priorities.

3.1.7 Drone

For this project an Open Source Drone was used, and design and documentation are found at [52] on GitHub. Any other drone with the following open-source hardware should work equally well.

3.1.8 Cube Orange

The Cube Orange flight controller is the brains of the drone. It is completely interoperable alongside the 5G Pi Hat accompanied by a Raspberry Pi, in the case the RPi 3B+ was used. Details for this setup are shown below in Section 2.2 . Commands sent to the drone via the Mobile Network are received by the Pi Hat, transferred to the RPi and then to Autopilot using the MAVlink communication protocol. [53]

3.2 Methodology

The following part of the project consists of most of the work to be done. This involves implementing the agreed-upon architecture, testing it and assessing its feasibility. The bigger architectural stack is made up of smaller components that each need to be run and tested. As such the Verification and validation model shown in Figure 3.3 is to be exercised to realise this phase of development. It will provide a structured framework going forward.

---

[6] `https://www.waveshare.com/sim8200ea-m2-5g-hat.htm`

This stage involves testing several open source platforms in as to reveal the one most suitable for this project. Hence this stage will be done in an iterative process.

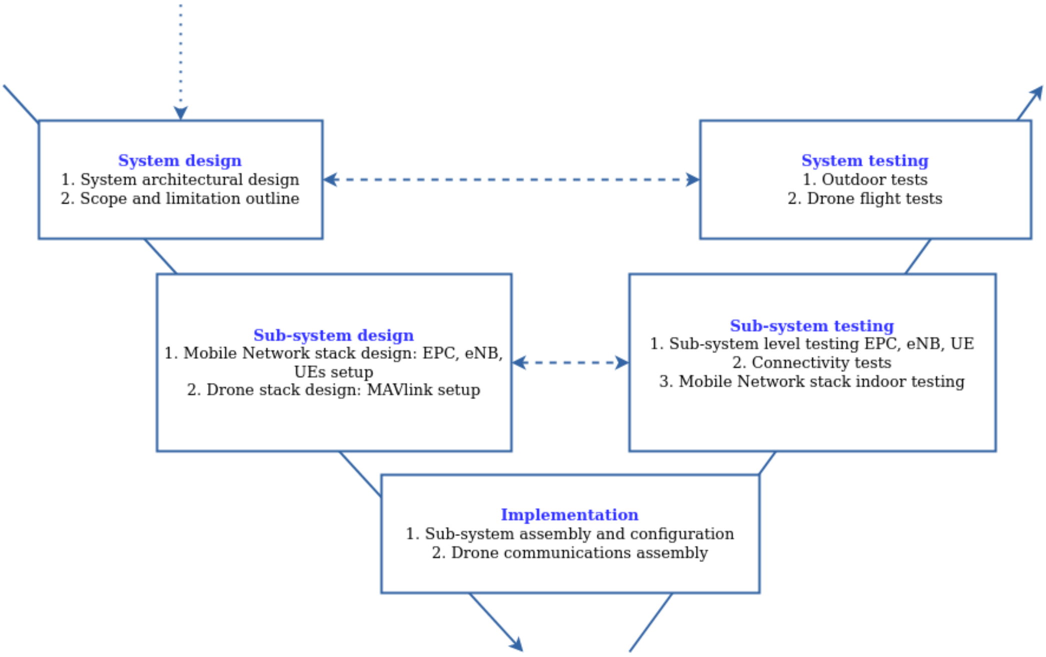


Figure 3.4: Specific V-model to be exercised throughput the implementation section

As shown in figure 3.4 the bigger system design is broken down into its sub-components which are then implemented individually. Along the way, all these sub-components are tested individually before bringing them all together to test the system as a whole. Which then will also be tested.

Testing

Validation and verification are to be done from the component level to the sub-system level all to way to the complete system level.

Component testing: Each component of the EPC, namely the MME, SPGW-C, SPGW-U, HSS, and UE is to be individually tested for connectivity. Performance tests will be carried out afterwards when connectivity has been achieved.

System testing: will be carried out on the entire finished product. Such tests include indoor performance tests and outdoor flight tests. These tests are described in greater detail in Section 4.

3.3 OpenAirInterface (OAI) 4G LTE test-bed Setup

This test bed comprises of the OAI 4G Core Network and the eNB. These run on two machines connected and communicating via an ethernet connection. A 1Gbps ethernet cable is required so as not to be a throughput bottleneck within the system. Figure 3.5 shows the full connection from the mobile network to the user equipment. Instructions on how to set up the Core Network and eNB are given in detail by OpenAirInterface, [54].

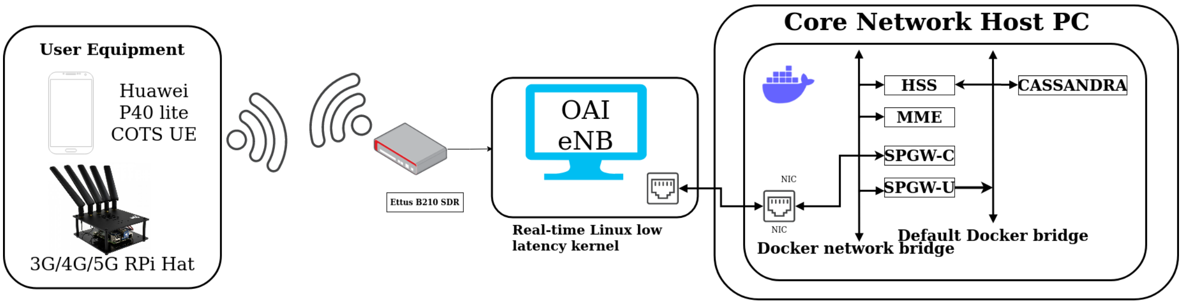


Figure 3.5: 4G LTE test-bed

3.3.1 User provisioning

The HSS container running on the Core Network (EPC) machine carries out the operation of entering and checking user subscriber information to and from the database. (In this case, the Cassandra DB) The MME container receives user subscriber information as soon as an UE attempts to connect and does the authentication by prompting the HSS to verify the existence of such a user. Therefore for the user subscriber to be able to connect to the network details of the UE need to be found in the database, and match the security credentials.

Important parameters for setting up the HSS are shown in Listing 4. These are parameters associated with devices to be connected to the Core Network via eNB. These parameters are stored in the HSS database and allow the core network to recognise devices that request a network attach and allow them to connect, giving them access to the internet and via SPGWC service. NB: The first 5 digits of the IMSI number should be made up of the ”MCC|MNC” parameters programmed in the USIM Card.

Listing 5 shows the parameters to be set for the MME container. As the MME is the main signalling node in the EPC, much of the parameters are to allow communication to and from the rest of the EPC. The MME is mainly responsible for authenticating mobile devices, hence the Mobile Country Code (MMC), Mobile Network Code (MNC) and Tracking Area Code (TAC), these are to be the same as the ones set up when programming the USIM Cards. In this case, 655, 25, 1 respectively were used. These represent the geographical area corresponding to South Africa.

The USIM cards need to be programmed with the correct corresponding parameters. Shown in Listing 1 parameters of main importance that have been programmed are as follows (Including the command to program using the pysim program):

- International Mobile Subscriber Identity (IMSI)
- Authentication Key (KI)
- Operator Code (OPC)
- Service Provider Name (SPN)
- Authentication Algorithm

3.3.2 Connecting the eNB and Core Network machines

The machines are connected together using ethernet cables, each machine is connected to every other machine. The eNB and gNB machines will need to know the EPC IP address and the MME IP address. In the eNB and gNB configuration files, specify the MME IP address.

```
////////// MME parameters:
   mme_ip_address      = ( { ipv4       = "**YOUR_EPC_IP_ADDR**";
                             ipv6       = "192:168:30::17";
                             active     = "yes";
                             preference = "ipv4";
                           });
```

Listing 2: eNB to EPC configurations

On the eNB machine the following parameters shown in listing 3 need to be configured in the correct configuration file. (i.e eNB.conf)

```
    NETWORK_INTERFACES :
   {
       ENB_INTERFACE_NAME_FOR_S1_MME            = "eth0";
       ENB_IPV4_ADDRESS_FOR_S1_MME              = "**YOUR_ENB_IP_ADDR**";
       ENB_INTERFACE_NAME_FOR_S1U               = "eth0";
       ENB_IPV4_ADDRESS_FOR_S1U                 = "**YOUR_ENB_IP_ADDR**";
       ENB_PORT_FOR_S1U                         = 2152; # Spec 2152
   };
```

Listing 3: eNB configuration parameters

An IP route needs to be configured between the two machines to enable communication via the IP protocol. This is done by creating a route to the Core Network docker network bridge running on the Core Network machine. This is done as follows:

```
sudo ip route add 192.168.61.128/26 via 16.0.0.1 dev enx00e04e69659b
```

- Where 192.168.61.128 is the IP address of the Docker Host
- Where enx00e04e69659b is the Network Interface Controller(NIC) of the eNB machine

To ensure this works you should be able to ping the Core Network Containers attached to the main docker network bridge successfully.

3.3.3 OAI EPC setup

The OpenAirInterface EPC can be found on their Github [7], along with installation steps and instructions. This sub-section will only show important configuration parameters for the experiment carried out as well as running the Core Network. The OAI EPC is as shown in Figure 3.6 below. The EPC comprises of 4 main components each offering different services, namely HSS, MME and SPGW, split into SPGWC and SPGWU-tiny. The components have all been containerised using Docker. They can be configured and deployed using Docker-compose. Instructions are available under this project GitHub, [54].

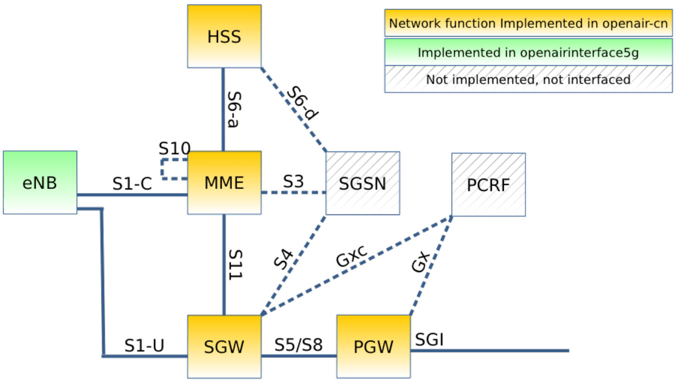


Figure 3.6: OpenAirInterface Core Network (EPC) structure

```
    --apn1=apn1.carrier.com
    --imsi=655250000000001
    --ltek=efbfbdefbfbd473f2fefbfbdd094efbf
    --op=00000000000000000000000000000000
```

Listing 4: HSS set parameters

```
    --hss_s6a=${HSS_IP}
    --mme_s6a=${MME_IP}
    --mme_s1c_IP=${MME_IP}
    --mme_s1c_name=eth0
    --mme_s10_IP=$"127.0.0.10"
    --mme_s10_name=lo
    --mme_s11_IP=${MME_IP}
    --mme_s11_name=eth0
```

```
    --spgwc0_s11_IP=${SPGW0_IP}
    --mcc=655 --mnc=25 --tac_list="1 2 3"
```

Listing 5: MME set parameters

```
    --s11c=eth0 --sxc=eth0
    --apn=apn1.carrier.com
    --dns1_ip=137.158.xxx.xx
    --dns2_ip=8.8.4.4
    --sxc_ip_addr=${SPGWC_IP}
    --sxu=eth0
    --s1u=eth0
```

Listing 6: SPGWC and SPGWU-tiny set parameters

3.3.4 Execution

Firstly you need to run the EPC. This is done by initialising the Cassandra database and then deploying of the Core Network function containers. Run the following commands from the folder containing the Docker-compose file.

```
$ docker-compose up -d db_init
```

Listing 7: Initialise Cassandra database

```
$ docker-compose up -d oai_spgwu
demo-cassandra is up-to-date
Creating demo-redis   ... done
Creating demo-oai-hss ... done
Creating demo-magma-mme ... done
Creating demo-oai-spgwc ... done
Creating demo-oai-spgwu-tiny ... done
```

Listing 8: Deploy Core Network functions

Afterwards, run the eNB on the other machine, code listing 9 shows how to build the eNB executables. Listing 10 shows the command to run the eNB executables, this command will prompt the SDR to switch on and be on standby for any UE attachments.

```

cd <your oai installation directory>/openairinterface5g/
source oaienv
cd cmake_targets/
./build_oai -I -w USRP --eNB --UE
```

Listing 9: Build eNB excutables

From the build folder:

```
sudo ./lte-softmodem -O **YOUR_ENB_CONF_FILE** | tee **YOUR_LOG_FILE**
```

Listing 10: Run eNB

3.4 OpenAirInterface 5G NSA test-bed setup

There are various ways of implementing a 5G NSA test bed. OAI has implemented the ”EPC and two base stations” format. (Often referred to as Option 2) That is using the 4G Evolved Packet Core and two base stations. One being the eNB and one being the gNB (5G base station). The NodeBs all have their own separate SDRs corresponding to the eNB and gNB respectively, running on separate machines.

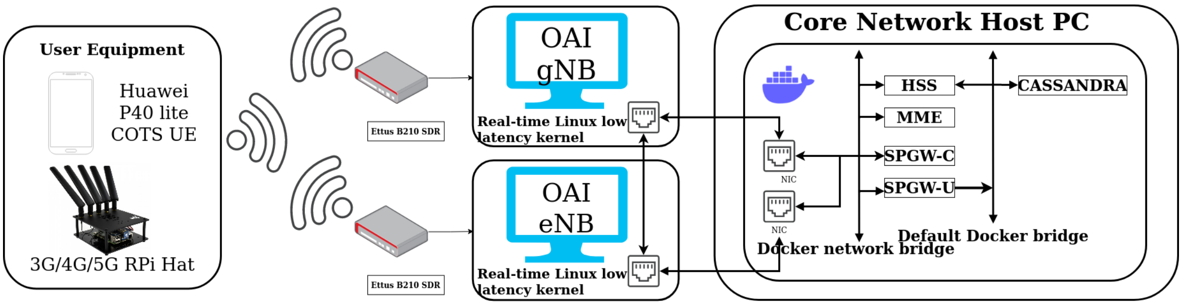


Figure 3.7: 5G OAI LTE Test-bed

A UE attaches to the mobile network via the 4G LTE eNB and if its user parameters show that it is 5G capable and compatible, that connection is passed on to the 5G base station (gNB). Figure 3.7 shows the setup, quite similar to the 4G test-bed with the addition of a gNB. User provisioning follows the same procedure as that outlined above for the 4G test bed, as it is still the same core network being used.

For the NSA deployment, besides this connection, the eNB and gNB need to be connected to each other as well. They communicate via the X2 interface, this is done by enabling X2 and setting up the appropriate IP addresses in the configuration files. Shown below in listing 11 are the X2 interface settings needed as well as a description of their actual value.

```
///X2
   enable_x2 = "yes";
   t_reloc_prep      = 1000;      /* unit: millisecond */
   tx2_reloc_overall = 2000;      /* unit: millisecond */
   target_enb_x2_ip_address      = (
                                    { ipv4       = "**YOUR_ENB_IP_ADDR**";
                                      ipv6       = "192:168:30::17";
                                      preference = "ipv4";
                                    }
                                   );

```

```
    NETWORK_INTERFACES :
    {

        GNB_INTERFACE_NAME_FOR_S1_MME            = "eth0";
        GNB_IPV4_ADDRESS_FOR_S1_MME              = "**YOUR_GNB_IP_ADDR**";
        GNB_INTERFACE_NAME_FOR_S1U               = "eth0";
        GNB_IPV4_ADDRESS_FOR_S1U                 = "**YOUR_GNB_IP_ADDR**";
        GNB_PORT_FOR_S1U                         = 2152; # Spec 2152
        GNB_IPV4_ADDRESS_FOR_X2C                 = "**YOUR_GNB_IP_ADDR**";
        GNB_PORT_FOR_X2C                         = 36422; # Spec 36422
    };
```

Listing 11: Enabling X2 interface from gNB

Execution

Building the 5G NSA network, following the same procedure as the 4G LTE network described above with the one addition of a gNB running another SDR.

Build the gNB executables:

```
cd <your oai installation directory>/openairinterface5g/
source oaienv
cd cmake_targets/
./build_oai -I -w USRP --gNB
```

Listing 12: Build gNB excutables

Run the gNB from the build folder:

```
sudo ./nr-softmodem -O **YOUR_GNB_CONF_FILE** -E | tee **YOUR_LOG_FILE**
```

Listing 13: Run the gNB

In summary, to spin up the 5G NSA network:

1. Setup and execute the EPC as described in Section 3.3.3
2. Run the eNB executable and ensure connection to the EPC
3. Run the gNB executable and ensure connection to both the EPC and the eNB.
4. Remove COTS UEs from airplane mode, the base stations should now be ready to receive connections.

Figure 3.8: 4G and 5G indoor Setup

Figure 3.8 shows the indoor setup used to realise these testbeds. Two laptops running The eNB, and EPC. The machine running the EPC is also running the GCS (Mission Planner) for drone communication. A third machine is shown (CPU) for running the gNB for 5G NSA deployments. The LANDRS Nyala HexaQuad is also shown, this time in hexa-rotor configuration.

3.5 sUAS communication

Following the 4G LTE test-bed setup and running experiments to discover Key Performance Indicators (KPIs), the next step is to set up the sUAS communication. This subsection shows the currently used conventional communication architecture used for sUAS 's and DIY drones which use opensource hardware such as the Pixhawk [55] as well as opensource software stacks such as Ardupilot, [56] and QGroundControl [57]. This sub-section then goes on to show some of the drone communication architectures developed and implemented. Their advantages and limitations are also considered so as to carry out flight experiments with the best model.

3.5.1 Conventional drone communication link

The conventional communication link between the ground user and sUAS uses one or both of the following options: a direct communication link to the drone via a controller to the drone at 2.5GHz or a PC running opensource autopilot software such as QGroundControl and Mission Planner with attached attached telemetry antennas 866Mhz. This link is shown in Figure 3.9.

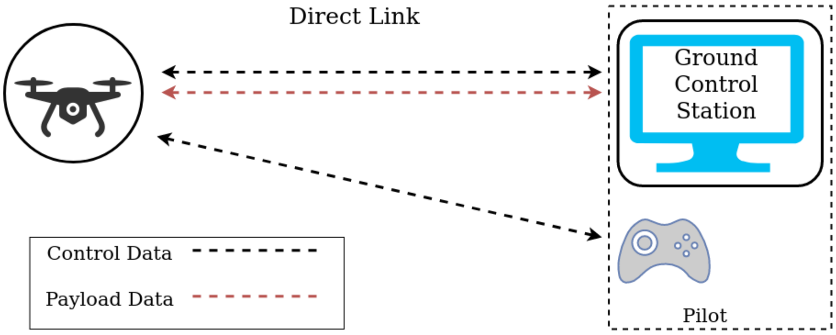


Figure 3.9: Conventional Communication link

The drone to be used for these experiments is the LANDRS Science Drone (Nyala HexaQuad) [52]. This is an opensource drone design re-configurable between being a quad copter or a hex-copter and meeting all the requirements to be categorised as a drone as defined earlier. It is equipped with all the necessary equipment to realise the conventional communication link, annotated in Figure 3.10. It is to be noted that this is the default mode of communication on this drone, as with most drones.

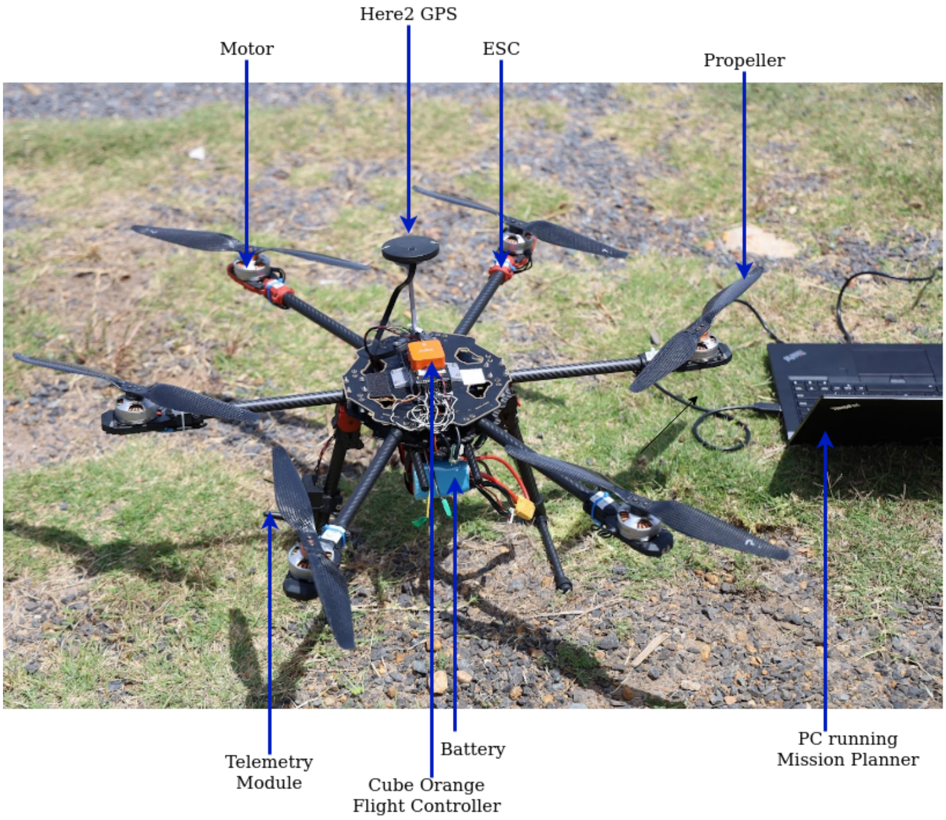


Figure 3.10: Hex-copter used for experiments with conventional communication equipment

The HexaQuad is equipped with the following:

- **Flight Controller**- The Cube Orange [58] was used. This is the "brain" of the drone. It comes with an H7 SOC with 1 GB RAM, and combines all the incoming data (from the user and from sensors etc) and sends the appropriate electric signals to the motors for a stable and secure flight. The Inertial Measurement Device (IMU) consists of an accelerometer, a magnetometer and a gyroscope for knowledge of motion and location and autonomous flight capabilities. [43] The flight controller communicates with the GroundControl Station (GCS) (i.e.Mission Planner, QGroundControl) sending MAVLink protocol messages [53].
- **Telemetry module**- The UAV and the GCS at the edge utilize a dedicated telemetry link in 433MHz/866MHz for transmitting flight and real-time system configuration data. The first of the two paired modules are serially connected to the flight controller, while the second one is USB connected to the computing node at the GCS.
- **Electronic Speed Controllers** (ESCs)- The ESCs are connected both to the PMS (for electric power supply) and to the flight controller (for control signalling) and are used for regulating motor spin [43].

- **Radio Control Receiver**- The remote pilot can control the drone using a special remote-control transmitter (Tx) sending PPM signals in 2.4 GHz, which is used for security reasons or just manual flying. The flight controller receives the signals using a radio receiver module (Rx).
- **Global Positioning System** (GPS) module- This module combines a GPS antenna and a compass. It is placed away from the UAV electronics to avoid interference and provides the autopilot with all the necessary positioning data for autonomous navigation. [43]
- **Power Management System** (PMS)- This electric circuit is responsible for managing the electric power coming from the UAV main battery, regulating and distributing it to all the onboard devices (e.g., motors, flight controller etc.).

Figure 3.11: Landrs Nyala Hexaquad drone used for experiments (In quad-rotor configuration)

In addition to the above conventional equipment, additional equipment is added on to the drone to implement this project, i.e Raspberry Pi 3B+ (RPi) which can be seen in Figure 3.11 with four attached antennas connected to a 5G capable RPi Hat. The functionality of this is described later in this Section.

3.5.2 Initial Proposed Mobile Network Communication Architecture

The architecture initially used to realise drone command and control by leveraging mobile networks is adapted from [17]. The architecture is shown in Figure 3.12 and explained below. In this scenario the Mobile network in use can be commecial mobile networks (For example MTN/Vodacom in South Africa) or our on private open-source mobile network implemented in Section 3.3, with the capability of internet access. This can be realised as 4G LTE or 5G as long as the mobile phone used is both LTE and 5G-NR capable.

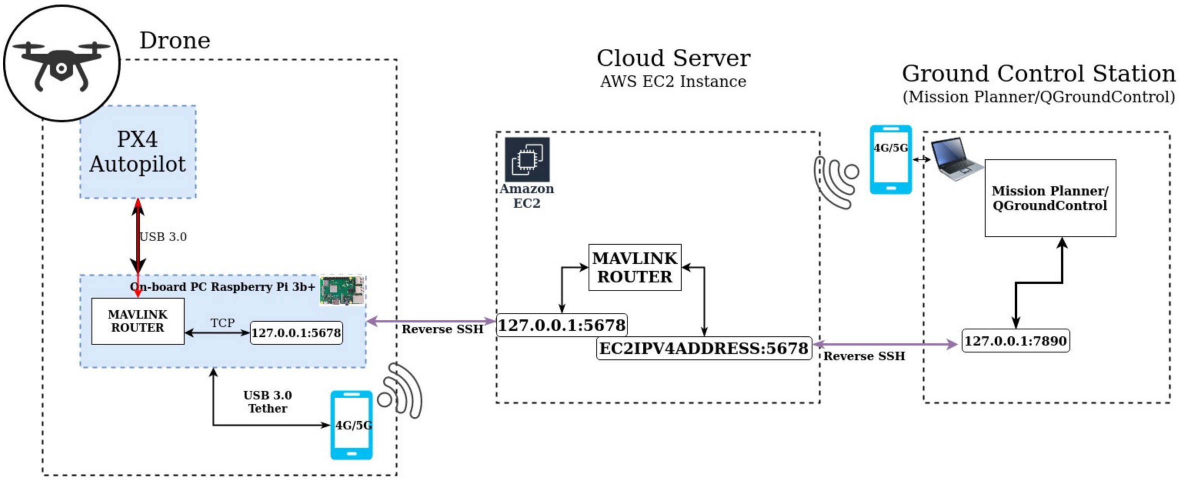


Figure 3.12: Design architecture used to realise the drone communication leveraging both the non public and public Mobile Networks realised

The architecture uses a remote server, in this case, the AWS EC2 [59] instance solely running Mavlink Router [60]. The drone and the ground control station will thus communicate via the EC2 instance via reverse ssh. Internet connectivity will be provided by the Mobile Network via the SPGWU-tiny container of the Core Network for 4G LTE, 5G-NR. (As the OAI 5GCore still utilizes the same container for this functionality)

On the drone side, the Raspberry Pi 3B+ was used to communicate with autopilot, internet connection is supplied by the mobile phone, tethered to the Pi hence providing internet access and allowing IP communication and hence reverse SSH to remote servers. The same is true for the Ground Control Station, also tethered by the phone communicates with the server using reverse SSH.

This solution does however come with some drawbacks. These are listed below,

- Using a remote server introduces some overhead latency. Depending on the location the deployed server communication becomes less and less real-time.
- Since a remote server is being used and communication is IP-based, an internet

connection will be required at all times.

This design's major advantage is that communication is possible without needing a private mobile network. Any 4G LTE/5G-NR capable handset equipped with a SIM card registered to any network provider in any country will work. Hence allowing Drone to Ground Control Station communication to be possible and not limited to the physical geographical location. Another advantage of the EC2 instance is that it has very high bandwidth and very low latencies for communication in the same geographical area. To the extent that it would not bottleneck any communication performance within this architecture.

3.5.3 Proposed Mobile Network Communication Architecture

The proposed architecture more suitable for this project utilises the same analogy and reasoning as the solution above but introduces some novelty to the Core Network. This is done by introducing a new Core Network function whose sole purpose is routing Mavlink data via IP.

Instead of using the remote AWS EC2 server, the SPGW-U container gets added functionality by installing within it, the MAVlink communication protocol and MAVlink-Router. This allows the Core Network machine to communicate with the drone. This being the only change to the Core Network, the remaining step is to make the drone Mobile Network capable.

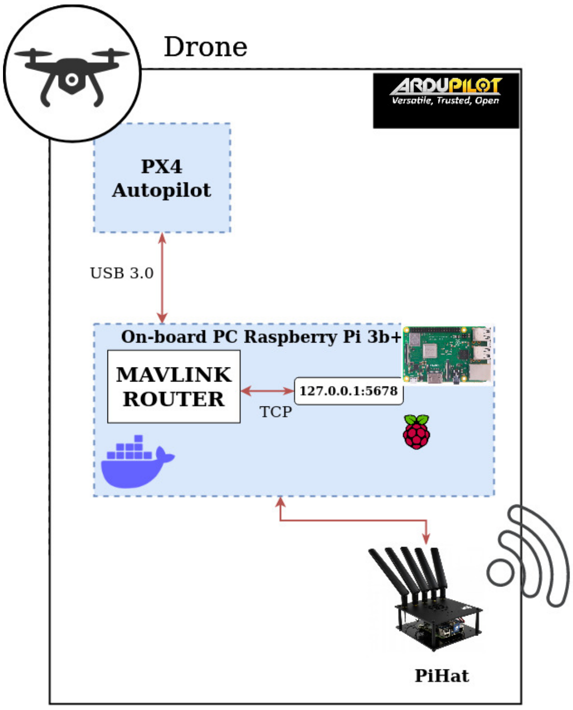


Figure 3.13: Drone as a Mobile Network User Equipment

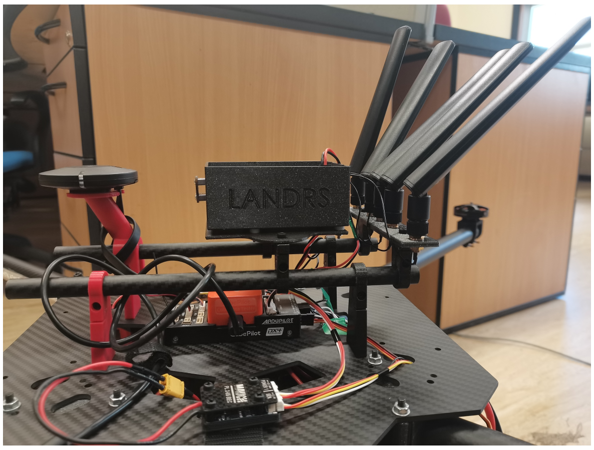


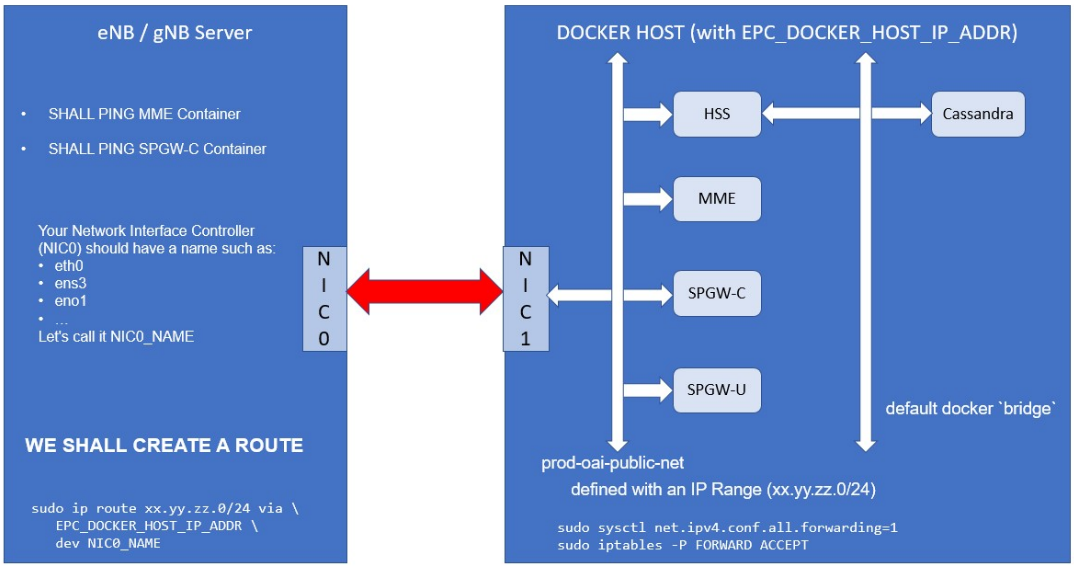
Figure 3.14: Drone as a Mobile Network User Equipment

Figure 3.13 shows how to make this possible. Attaching the Raspberry Pi to the drone would make it a companion computer for the drone. It was connected via USB 3.0 to the autopilot and communication between the two can be established using the MAVlink protocol. The PiHat allows the Raspberry Pi to connect to the mobile network hence the autopilot automatically inherits the mobile network connection. Thus the drone becomes a mobile network UE. Figure 3.14 shows the equipment mentioned above with the Raspberry Pi and Pi hat enclosed in the black 3D printed casing as crash protection.

The Architecture shown in Figure 3.12 now evolves to that shown at the beginning of the chapter, Figure 3.1. Such a change addresses the disadvantages of the previous architecture, the main one being the high latency. The fact that the MAVlink router container is now localised, and in the same IP network, time to travel is greatly reduced thus greatly reducing the latency. This however comes at a trade-off and that's the fact that operation is now solely localised, hence drone operation is limited to the specific geographical area covered by the mobile network and commercial mobile networks can no longer be used since they do not contain this bespoke CNF (Core Network Function i.e MAVlink protocol) within their Core Networks.

3.6 Expected behavior

To ascertain that the mobile network has been installed, configured and running correctly, subsystem-level testing had to be done. This was done during and after setting up the mobile network. This subsection entails the details of how these tests were carried out. Firstly component testing was done to verify that all of the mobile network components that were to communicate with each other could do so. The deployed mobile network architecture was designed by OpenAirInterface and is shown below in figure 3.15. Figure 3.8 shows the layout of the deployment in the lab.



- SHALL PING MME Container
- SHALL PING SPGW-C Container

Your Network Interface Controller (NIC0) should have a name such as:
- eth0
- ens3
- eno1
- ...

Let's call it NIC0_NAME

WE SHALL CREATE A ROUTE

```
sudo ip route xx.yy.zz.0/24 via \
   EPC_DOCKER_HOST_IP_ADDR \
   dev NIC0_NAME
```

prod-oai-public-net defined with an IP Range (xx.yy.zz.0/24)

```
sudo sysctl net.ipv4.conf.all.forwarding=1
sudo iptables -P FORWARD ACCEPT
```

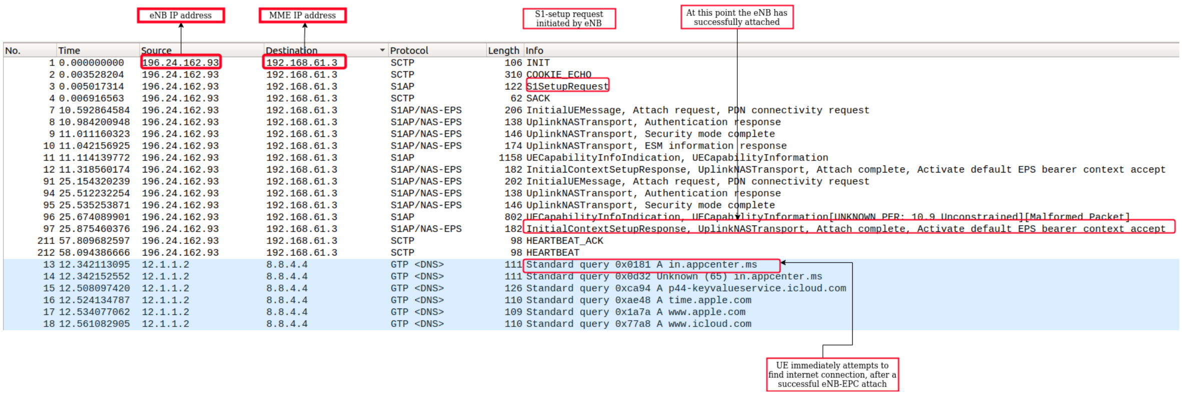
Figure 3.15: Deployed mobile network architecture (Both EPC and eNB) [7]

Component connectivity

As shown in Figure 3.15 the eNB Server/PC should be able to ping the EPC host server/PC. Also, the eNB should be able to ping the SPGW-C and the SPGW-U components. The ”ping” network test transmits data packets to a specific IP address and either confirms or denies there is connectivity between IP-networked devices. Wireshark was also used to show more information about the network traffic, such as Source, Destination, Protocol and the information contained in data packets.

EPC ↔ eNodeB Pings were used to determine whether a communication channel is available between the EPC and eNB, shown below in Figure **??**. A successful ping shows that the two PCs are connected together (in this case via Ethernet) and can communicate, however, it does not prove that the eNB running can successfully attach to the Core Network.

A surefire way to ascertain this is to monitor the channel using Wireshark and observe the data packets transmitted as the connection procedure occurs between the eNB and MME container of the Core Network. Figure 3.16



```
No.   Time          Source          Destination     Protocol      Length Info
    1 0.000000000   196.24.162.93   192.168.61.3    SCTP            106 INIT
    2 0.003528204   196.24.162.93   192.168.61.3    SCTP            310 COOKIE_ECHO
    3 0.005017314   196.24.162.93   192.168.61.3    S1AP            122 S1SetupRequest
    4 0.006916563   196.24.162.93   192.168.61.3    SCTP             62 SACK
    7 10.592864584  196.24.162.93   192.168.61.3    S1AP/NAS-EPS    206 InitialUEMessage, Attach request, PDN connectivity request
    8 10.984200948  196.24.162.93   192.168.61.3    S1AP/NAS-EPS    138 UplinkNASTransport, Authentication response
    9 11.011160323  196.24.162.93   192.168.61.3    S1AP/NAS-EPS    146 UplinkNASTransport, Security mode complete
   10 11.042156925  196.24.162.93   192.168.61.3    S1AP/NAS-EPS    174 UplinkNASTransport, ESM information response
   11 11.114139772  196.24.162.93   192.168.61.3    S1AP           1158 UECapabilityInfoIndication, UECapabilityInformation
   12 11.318560174  196.24.162.93   192.168.61.3    S1AP/NAS-EPS    182 InitialContextSetupResponse, UplinkNASTransport, Attach complete, Activate default EPS bearer context accept
   91 25.154320239  196.24.162.93   192.168.61.3    S1AP/NAS-EPS    202 InitialUEMessage, Attach request, PDN connectivity request
   94 25.512232254  196.24.162.93   192.168.61.3    S1AP/NAS-EPS    138 UplinkNASTransport, Authentication response
   95 25.535253871  196.24.162.93   192.168.61.3    S1AP/NAS-EPS    146 UplinkNASTransport, Security mode complete
   96 25.674089901  196.24.162.93   192.168.61.3    S1AP            802 UECapabilityInfoIndication, UECapabilityInformation[UNKNOWN PER: 10.9 Unconstrained][Malformed Packet]
   97 25.875460376  196.24.162.93   192.168.61.3    S1AP/NAS-EPS    182 InitialContextSetupResponse, UplinkNASTransport, Attach complete, Activate default EPS bearer context accept
  211 57.809682597  196.24.162.93   192.168.61.3    SCTP             98 HEARTBEAT_ACK
  212 58.094386666  196.24.162.93   192.168.61.3    SCTP             98 HEARTBEAT
   13 12.342113095  12.1.1.2        8.8.4.4         GTP <DNS>       111 Standard query 0x0181 A in.appcenter.ms
   14 12.342152552  12.1.1.2        8.8.4.4         GTP <DNS>       111 Standard query 0x0d32 Unknown (65) in.appcenter.ms
   15 12.508097420  12.1.1.2        8.8.4.4         GTP <DNS>       126 Standard query 0xca94 A p44-keyvalueservice.icloud.com
   16 12.524134787  12.1.1.2        8.8.4.4         GTP <DNS>       110 Standard query 0xae48 A time.apple.com
   17 12.534077062  12.1.1.2        8.8.4.4         GTP <DNS>       109 Standard query 0x1a7a A www.apple.com
   18 12.561082905  12.1.1.2        8.8.4.4         GTP <DNS>       110 Standard query 0x77a8 A www.icloud.com
```

Figure 3.16: eNB to EPC connection procedure via the S1 interface

eNodeB ↔ COTS UE To show that COTS UE has successfully connected to the base station eNB, the bearer name should appear on the COTS UE, monitoring the base station via the terminal will show the connection procedure taking place. In this case the bearer name is ”65525 Openairinterface” and the screenshots from the COTS UEs verify this. Figure 3.17, 3.18.

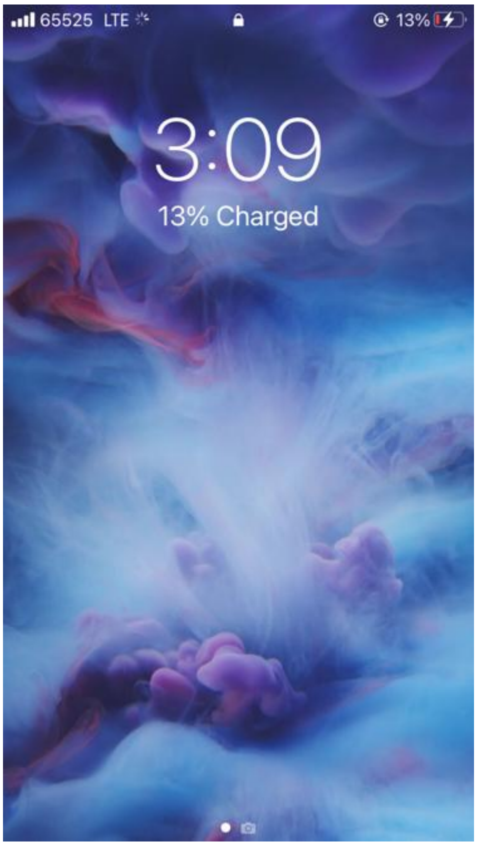
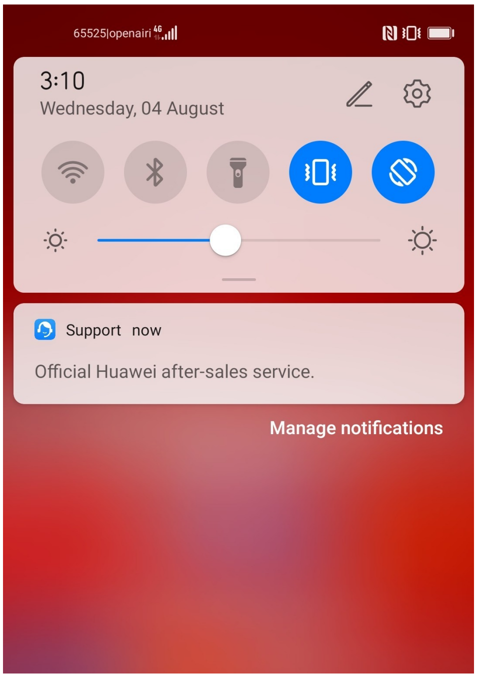


Figure 3.17: Iphone COTS UE connected to eNB

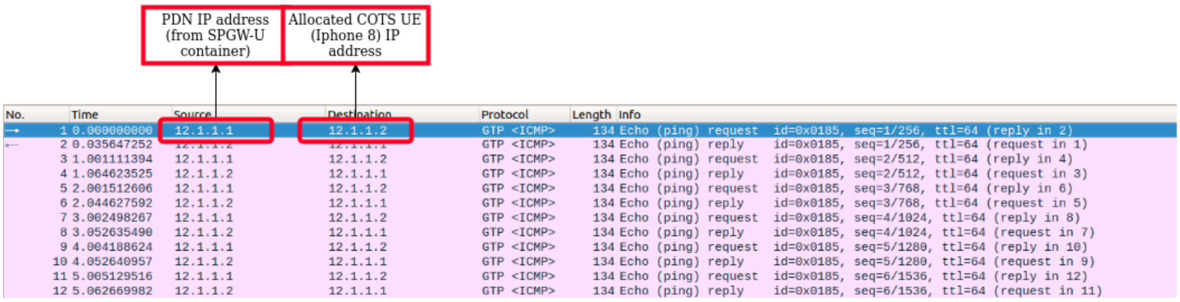


Support now

Official Huawei after-sales service.

Figure 3.18: Huawei COTS UE connected to eNB

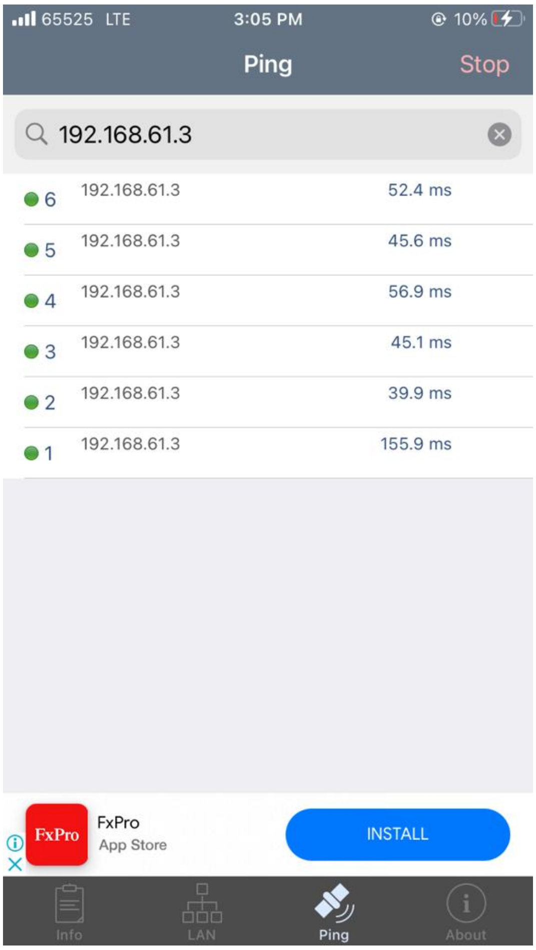
EPC ↔ COTS UE COTS UEs that attach to the network are assigned IP address in the range "12.1.1.0/24" The SPGWC and SPGWU containers handles UEs and external networks. Specifically, with OpenAirInterface the SPGW-U container handles the PDN (Packet Data Network) and IP prefix allocation hence pings to the SPGW-U container to and from UEs would prove a successful communication channel between UEs and the EPC. The PDN has IP address 12.1.1.1.



```
No.     Time         Source      Destination     Protocol   Length Info
  1 0.000000000   12.1.1.1      12.1.1.2        GTP <ICMP>    134 Echo (ping) request  id=0x0185, seq=1/256, ttl=64 (reply in 2)
  2 0.035647252   12.1.1.2      12.1.1.1        GTP <ICMP>    134 Echo (ping) reply    id=0x0185, seq=1/256, ttl=64 (request in 1)
  3 1.001111394   12.1.1.1      12.1.1.2        GTP <ICMP>    134 Echo (ping) request  id=0x0185, seq=2/512, ttl=64 (reply in 4)
  4 1.064623525   12.1.1.2      12.1.1.1        GTP <ICMP>    134 Echo (ping) reply    id=0x0185, seq=2/512, ttl=64 (request in 3)
  5 2.001512606   12.1.1.1      12.1.1.2        GTP <ICMP>    134 Echo (ping) request  id=0x0185, seq=3/768, ttl=64 (reply in 6)
  6 2.044627592   12.1.1.2      12.1.1.1        GTP <ICMP>    134 Echo (ping) reply    id=0x0185, seq=3/768, ttl=64 (request in 5)
  7 3.002498267   12.1.1.1      12.1.1.2        GTP <ICMP>    134 Echo (ping) request  id=0x0185, seq=4/1024, ttl=64 (reply in 8)
  8 3.052635490   12.1.1.2      12.1.1.1        GTP <ICMP>    134 Echo (ping) reply    id=0x0185, seq=4/1024, ttl=64 (request in 7)
  9 4.004188624   12.1.1.1      12.1.1.2        GTP <ICMP>    134 Echo (ping) request  id=0x0185, seq=5/1280, ttl=64 (reply in 10)
 10 4.052640957   12.1.1.2      12.1.1.1        GTP <ICMP>    134 Echo (ping) reply    id=0x0185, seq=5/1280, ttl=64 (request in 9)
 11 5.005129516   12.1.1.1      12.1.1.2        GTP <ICMP>    134 Echo (ping) request  id=0x0185, seq=6/1536, ttl=64 (reply in 12)
 12 5.062669982   12.1.1.2      12.1.1.1        GTP <ICMP>    134 Echo (ping) reply    id=0x0185, seq=6/1536, ttl=64 (request in 11)
```

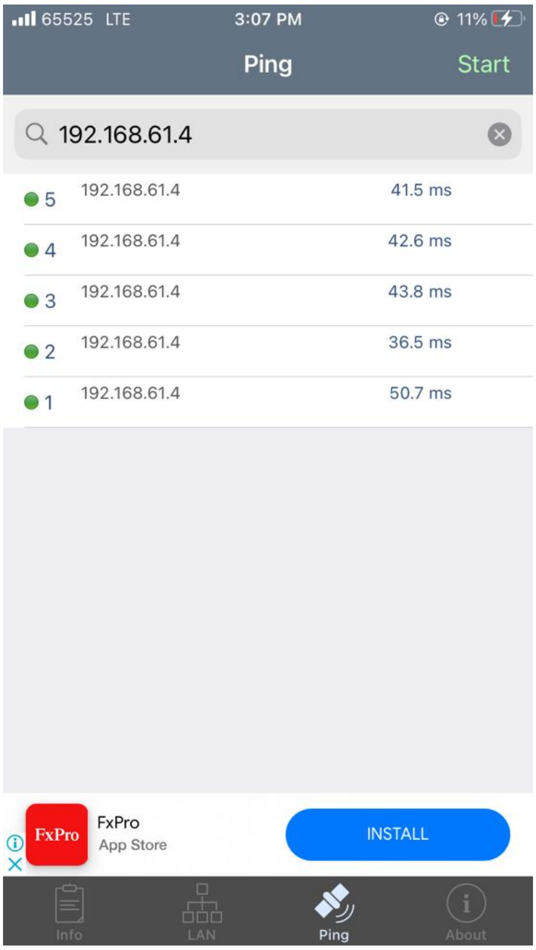
Figure 3.19: Successful ping requests and responses from the PDN to COTS UE (Iphone 8)

COTS UE ↔ EPC A free mobile application (NetAnalyser) was installed on the COTS UEs, to ensure that the mobile phones were configured and connected correctly to the EPC via the eNB. The structure of the OAI EPC allows the eNB to ping certain containers of the EPC (namely the MME, SPGWC, SPGWU with IP addresses 192.168.61.3, 192.168.61.4, 192.168.61.5 respectively). Other containers cannot be pinged directly from the eNB (i.e HSS and Cassandra). As shown in Figure 3.16 the eNB connected to the EPC via the S1 interface, meaning if this is the case any COTS UEs attached to this base station inherit the connection to the EPC as well. The screenshots below confirm this. Figure 3.20, 3.21, 3.22. A ping to Google's second public DNS IP address to verify that the SPGWU container is routing traffic correctly to external IP addresses. In this case, this shows a successful internet connection, 3.23.



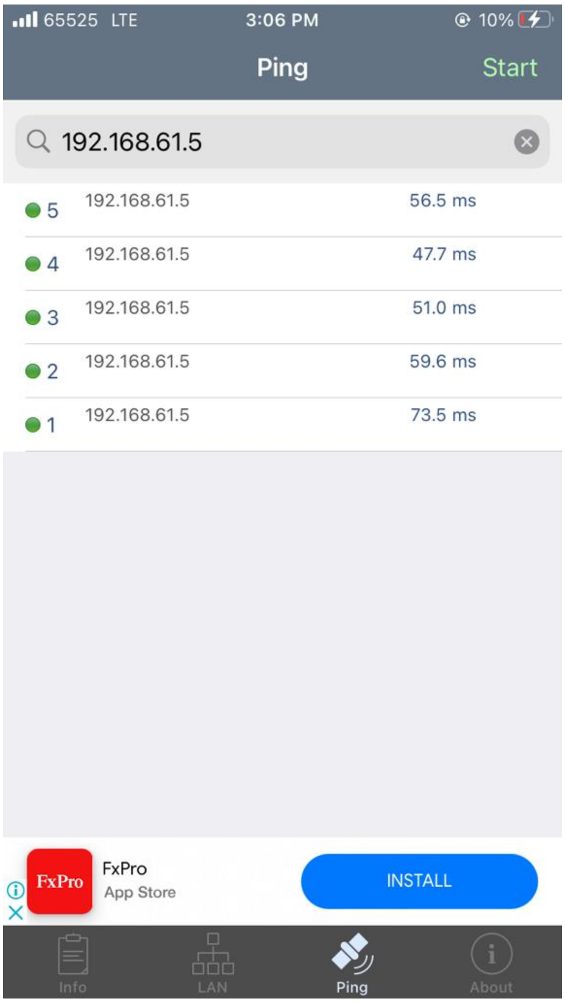
```
6   192.168.61.3    52.4 ms
5   192.168.61.3    45.6 ms
4   192.168.61.3    56.9 ms
3   192.168.61.3    45.1 ms
2   192.168.61.3    39.9 ms
1   192.168.61.3   155.9 ms
```

Figure 3.20: Iphone COTS UE to MME ping



```
5   192.168.61.4    41.5 ms
4   192.168.61.4    42.6 ms
3   192.168.61.4    43.8 ms
2   192.168.61.4    36.5 ms
1   192.168.61.4    50.7 ms
```

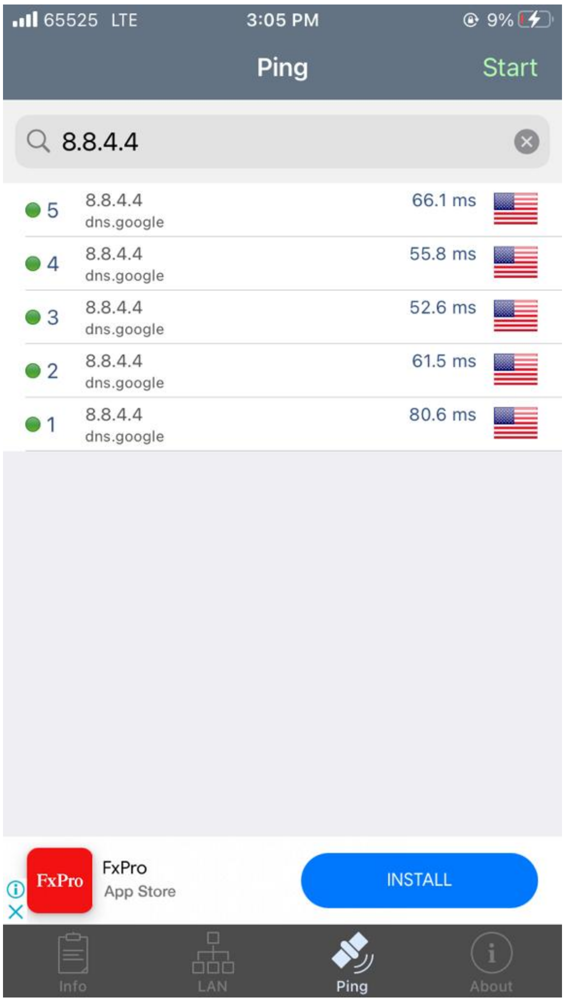
Figure 3.21: Iphone COTS UE to SPGWC ping



```
192.168.61.5

● 5   192.168.61.5      56.5 ms
● 4   192.168.61.5      47.7 ms
● 3   192.168.61.5      51.0 ms
● 2   192.168.61.5      59.6 ms
● 1   192.168.61.5      73.5 ms
```

Figure 3.22: Iphone COTS UE to SPGWU ping



```
8.8.4.4

● 5   8.8.4.4       66.1 ms
      dns.google
● 4   8.8.4.4       55.8 ms
      dns.google
● 3   8.8.4.4       52.6 ms
      dns.google
● 2   8.8.4.4       61.5 ms
      dns.google
● 1   8.8.4.4       80.6 ms
      dns.google
```

Figure 3.23: Iphone COTS UE to the Internet

3.7 Summary

This chapter described in detail the steps taken to see full working architecture. This working solution was presented at the very beginning and the subsequent sections show how it came to be. It starts by stating the prerequisites for the implementation and then goes on to show how to implement the 4G LTE test bed and the 5G testbed. It then showed the implementation of the sUAS communication architecture which can use both 4G and 5G NSA for drone communication. Lastly, the chapter showed the expected behaviour for a mobile network implementation that is working correctly and ready to be a part of the complete solution.

4 Performance Testing and Experiments

Performance testing was carried out to ascertain the capability of the mobile network implementation. Knowing its performance would help decide if it can feasibly be taken outside for a flight test and give a rough idea of what to expect when everything has been connected and flights are being carried out.

4.1 Indoor tests (IT1, IT2, IT3, IT4)

Indoor tests were tests carried out on the system whilst in the lab. These are the tests listed below. The performance tests were carried out on several different COTS UEs, namely the Huawei p40 lite 5G, iPhone 8, and the SIMCOM 5G capable module on an RPiHat development board. Tests were done using the OAI 4G setup, OAI 5G NSA setup and 5G standalone setup. It is to be noted that further research and experiments were done to explore other open-source mobile network implementations. The results of these experiments can be found in an upcoming paper in IEEE, by M.Chepkoech, N.Mombeshora et al. [8] and also presented in Section 5.

1. **Throughput tests (IT1)**. Throughput tests were carried out using `iPerf`. These tests include an investigation into the maximum bandwidth that the testbed is capable of using TCP protocol. The TCP protocol is suitable for large and bulk data transfers which are also loss intolerant. TCP tests will provide the maximum bit rate information for both Downlink (eNB to COTS UE/Emulated UE) and Uplink(COTS UE/Emulated UE to eNB) directions. Also, the UDP protocol will be tested, the UDP protocol is more suitable for data traffic which is loss tolerant and is more optimised for latency within the Linux kernels. UDP tests will provide useful information such as packet loss information, jitter and transfer rates.

   Throughput testing for the UDP protocol using iPerf requires that the bandwidth be specific, therefore to test the maximum throughput it is important to specify the ”b” flag with a ”0”. However, this will cause timing errors with the SDR as iPerf will attempt throughputs in the Gbps region which is too much for OAI to process. Another way of achieving this is to specify an arbitrary bandwidth value that is reasonably above the capability of the channel, in this case, 100Mbits was used. This way the iPerf will attempt to use the maximum throughput available on the channel. The command to run the downlink TCP throughput test from the SPGW-U TINY container is shown in listing 14, and for UDP listing 15.

```
iperf -c 12.1.1.2 -t 60 -i 1 -y C
```

Listing 14: TCP throughput test command

```
iperf -c 12.1.1.2 -t 60 -i 1 -u -y C -b 100
```

Listing 15: UDP throughput test command

2. **Latency and jitter (IT2)** This was tested using `iPerf` and `netPerf`. This provides insights into the travel times between the Core Network components and other networked devices including the drone.
3. **RSSI (IT3)** (Received Signal Strength Indicator) A measurement of cellular signal strength. RSSI is the measurement of received power from the cell site combined with noise and interference. It is represented as a negative number of deciBels per milliwatt (-dBm), with a good RSSI above -90 and a poor reading generally below -110. This was measured on the mobile phone COTS UEs using `Cellular Z` The RpiHat module provides functionality to measure this via AT commands. A python script has been set up to communicate these commands via serial as well as to log the results, this can be found at [61]. The AT command for measuring this is listed below. The AT Command documentation can be found at [13].

```
    ’AT+CSQ’, ’OK’,1
```

Listing 16: RSSI read command

4. **Base resource utilisation (IT4)** These tests were taken to gain insights into how ”heavy” and resource-consuming OAI is. This allows us to decide what machine to use and how the architecture can be improved, e.g. running the EPC on the drone companion computer. Changes in CPU and RAM were monitored while spinning up the Mobile Network.

4.2 Flight tests (FT1, FT2, FT3, FT4, FT5)

In addition to the above-mentioned parameters being measured, the ultimate goal is to take the drone to flight and analyse how the mobile network and communication stack implemented perform in a real flight. The following is going to be measured

1. **Channel throughput with distance (FT1)** Throughput will be measured during the flight. This will show how the transfer rate varies with distance from the ground station.
2. **Channel throughput with speed (FT2)** This provides an insight into whether or not the aircraft’s speed affects throughput and if so in what way it does. By measuring throughputs along different pre-configured flight paths and at pre-configured speeds, one can gain insight into this behaviour.

3. **RSSI with distance (FT3)** With the same analogy as above, one can discern whether or not the distance between the drone and base station has any impact on the received signal strength.
4. **RSSI with speed (FT4)** This helps gain insight into the effect the speed of the drone has on the received signal strength. Effects such as the Doppler frequency shift will be considered.

4.3 Distance calculations

In drone autopilot logs, longitude, latitude, and altitude values are recorded with time. As soon as the drone is armed. The drone logs the altitude of the drone from the take-off point, but unfortunately not the absolute distance from the base station or take-off point. Finding the absolute distance of the drone from the take-off position was done as follows.

Using the longitude and latitude values, one can calculate the distance between two GPS coordinates by using the most commonly used method, which incorporates the Haversine formula [62]. This formula was implemented by a Python script to calculate the distance. Below is the equation to do so, where Φ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km)

$$\begin{aligned} a &= \sin^2(\phi B - \phi A/2) + \cos\phi A * \cos\phi B * \sin^2(\lambda B - \lambda A/2) \\ c &= 2 * \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)}) \\ d &= R \cdot c \end{aligned} \tag{2}$$

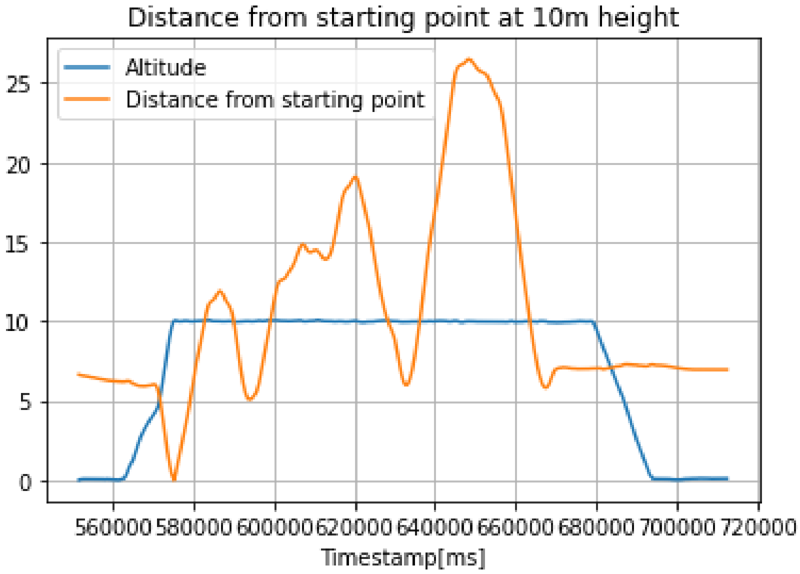


Figure 4.1: Distance from takeoff during one flight plan at 10m altitude

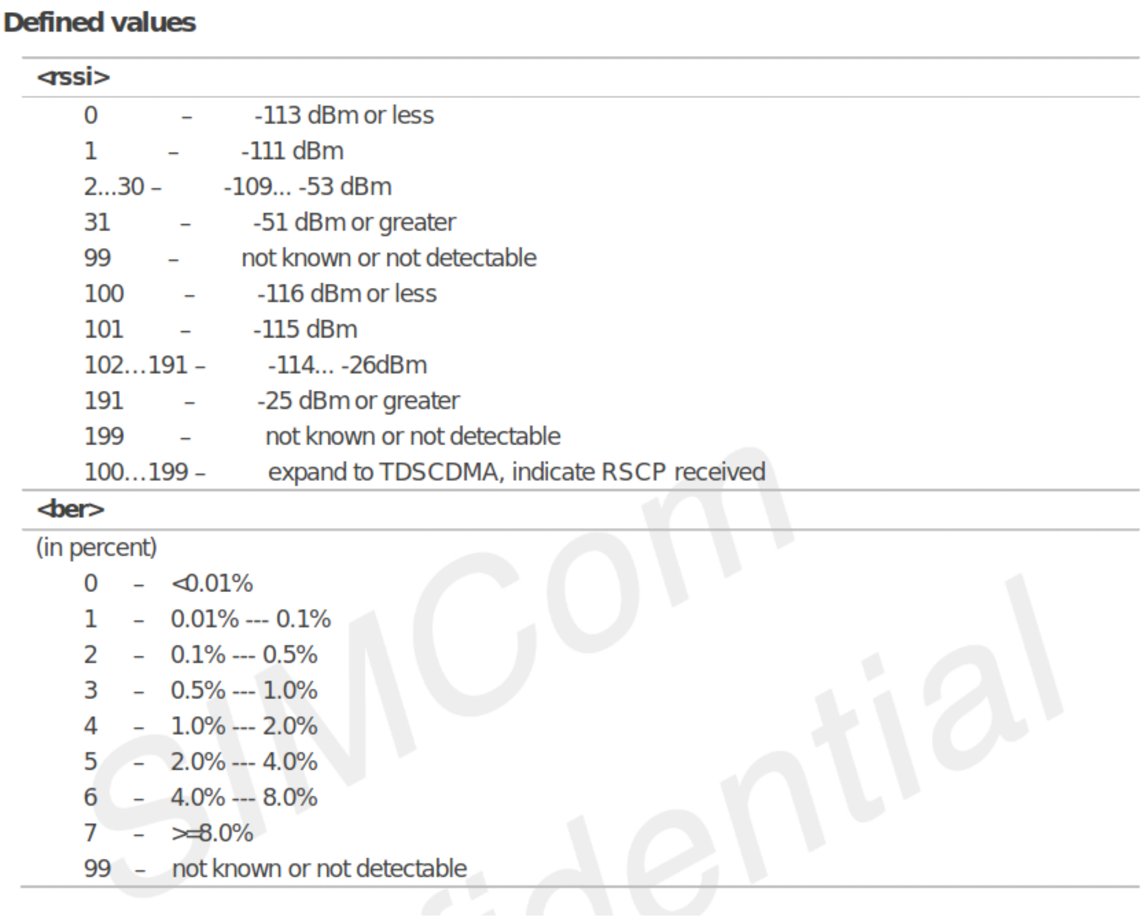
Figure 4.1 shows how the distance from the starting position varies with time. (N.B: The starting position in this case is at 10m height) It is clear from the concentric circles flown. From here onwards, getting the absolute distance is a matter of using the Pythagoras theorem, using the altitude/height (h) and distance from the starting point (d) to get the absolute distance from the take-off point(x). (NB: The take-off point was relatively close to the base station)

$$distance_{from\ base\ station}(x) = \sqrt{h^2 + d^2} \tag{3}$$

4.4 Monitoring and logging commands

The logging and monitoring script used to read the above KPI parameters is a Python script, it executes AT-Commands and communicates with the PiHat via serial interface.

`AT+CSQ` (Query Signal Strength command) Reads and returns the `<RSSI>` and `<ber>` (Bit Error Rate). The PiHat returns arbitrary RSSI values ranging from 0 to 199, these can then be translated to dBm values. The same is true for the BER values being translated to percentage values. This is shown in Table V.



Defined values

<rssi>

0 – -113 dBm or less
1 – -111 dBm
2...30 – -109... -53 dBm
31 – -51 dBm or greater
99 – not known or not detectable
100 – -116 dBm or less
101 – -115 dBm
102…191 – -114... -26dBm
191 – -25 dBm or greater
199 – not known or not detectable
100…199 – expand to TDSCDMA, indicate RSCP received

<ber>

(in percent)

0 – <0.01%
1 – 0.01% --- 0.1%
2 – 0.1% --- 0.5%
3 – 0.5% --- 1.0%
4 – 1.0% --- 2.0%
5 – 2.0% --- 4.0%
6 – 4.0% --- 8.0%
7 – >=8.0%
99 – not known or not detectable

Table V: Returned RSSI and BER values [13]

The information from the data above is limited. Getting the specific dBm values is not so straightforward. Advice from the PiHat SIM card module manufacturers (SIMCom) suggests using extrapolation to get an estimate of the actual RSSI dBm value. (Assuming a linear trend). Therefore Figure 4.2 was created for this. However, there is no way of knowing certain values (i.e. 31 and 191) marked as blue dots. The only information is that the RSSI is equal to greater than -53dBm and equal to or greater than -25dBm respectively. (Marked as blue dots) The red dot represents 100 (i.e. -116 dBm or less)

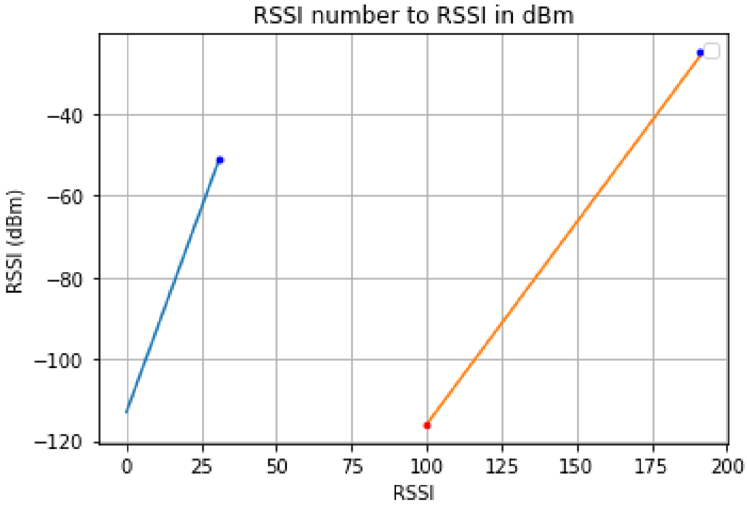


Figure 4.2: TableV presented graphically for extrapolation of RSSI values

4.5 Flight planning (FT5)

Before planning flights to carry out investigations on the above mentioned, some initial flights were carried out. For all these flights telemetry data is being streamed over the mobile network implemented as according to the proposed above in Section 3.5.3.

These initial flights were done as follows. Manual flights using the remote controller were carried out to get a rough estimate of the mobile network area coverage. Flying the drone horizontally and vertically away from the base station until the point of disconnection allows us to get a rough idea of the coverage distance we have to deal with. This will be essential for flight planning, as all the flights will be auto-mission.

Figure 4.3 shows the first three initial auto missions flown to gain an insight into expected behaviour. It shows two concentric rings at 10m in altitude and a ring flown at 15m in altitude.

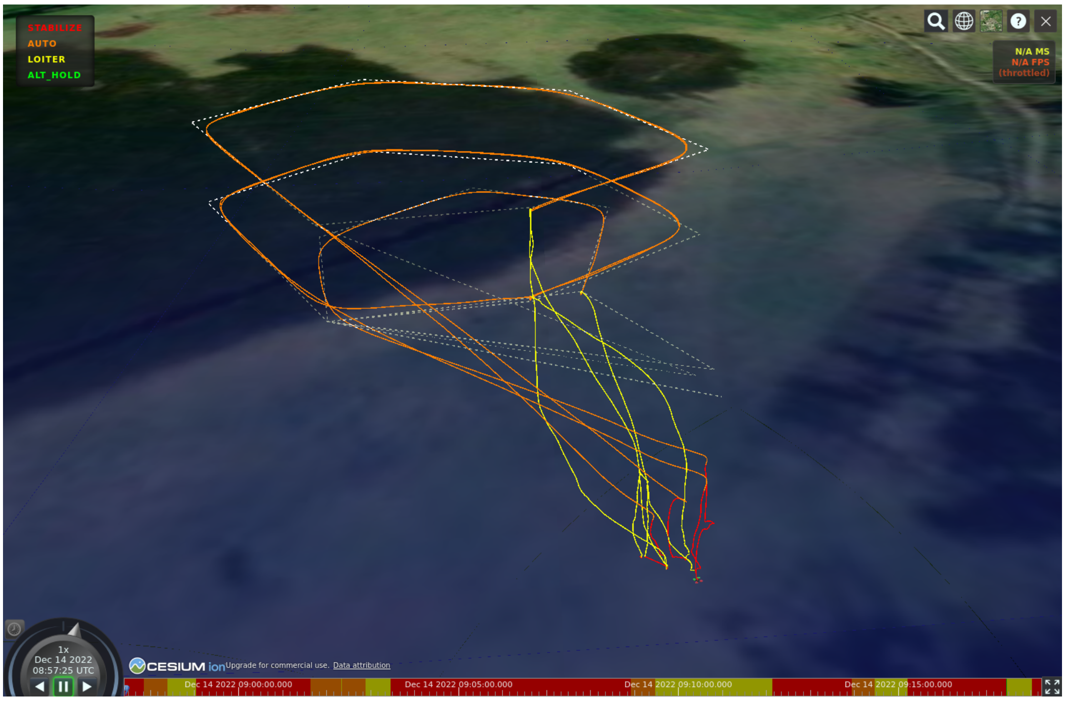


Figure 4.3: Preliminary flight auto-missions completed

Automated mission flight tests

Automated missions were set as follows:

- Fly concentric rings as automated missions, rings getting bigger in radius, thus varying the distance from the base station.
- Repeat the same flight path at that given height 3 times.
- Height was varied in 5m increments i.e 5m, 10m, 15m, 20m and 25m.

Figures 4.4, 4.5 and 7.1 show the flight paths used and sample flight altitude logs. Automated monitoring scrips will be running on the Raspberry Pi, logging the above-mentioned data in Section 4.2.

Figure 4.4: Flight plan carried out at different heights (Birds eye view)

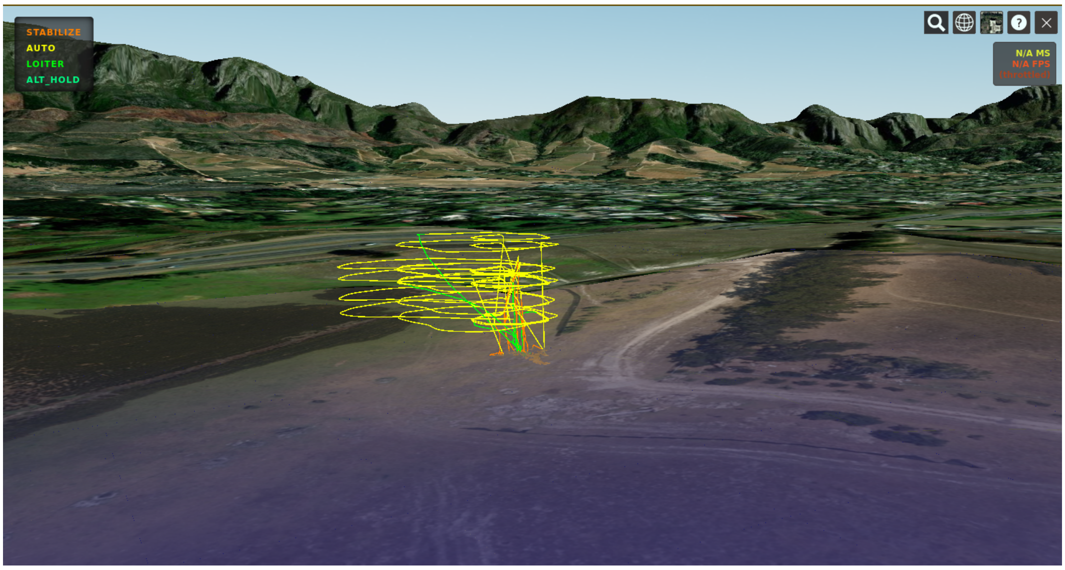


Figure 4.5: Flight plan carried out at different heights

Figure 4.6: Outdoor Setup

4.6 Summary

This section outlined the performance tests to be carried out. Starting with the indoor test, which is constituted of testbed implementations. It shows the exact KPIs tested and how the tests were carried out. Similar procedures were used to show the flight test performance, including showing the designed flight paths. Also shown were insights into how reading and understanding some of the logs that'll be returned from experiments.

5 Results and Discussion

This section presents the results obtained from the experiments described in Section 4. It shows the results of the several indoor test and the outdoor flight tests performed, whilst at the same time explaining the results and discussing the key findings gained from these results. Testbed implementation performance was compared against the performance of other mobile networks including commercial ones. Lastly resource utilisation information is presented, to evaluate how running the mobile networks affects CPU and memory usage.

5.1 4G Throughput performance (IR1, IR2, IR3)

Throughput performance tests were carried out using `iPerf` as described in Section 4. TCP and UDP tests were carried out to determine the maximum achievable throughput across the communication channel, i.e from the Core Network to the COTS UEs. The results are shown below.

5.1.1 Huawei COTS UE

The deployment scenario for the following tests is as shown in Figure 3.5 using the Huawei COTS UE.

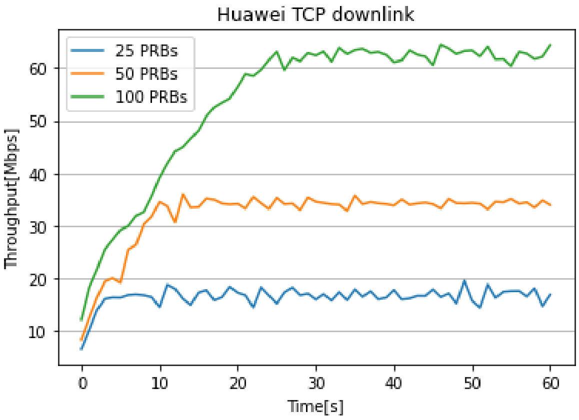


Figure 5.1: Huawei TCP downlink

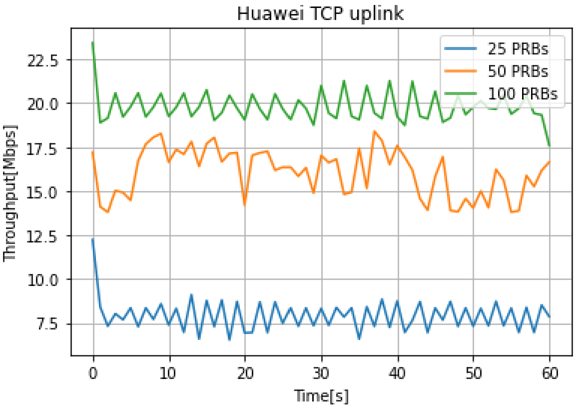


Figure 5.2: Huawei TCP Uplink

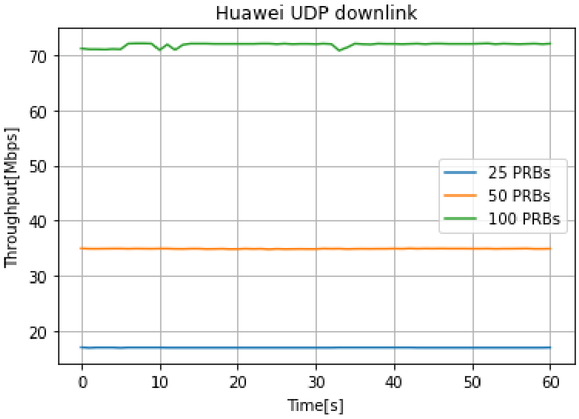


Figure 5.3: Huawei UDP downlink

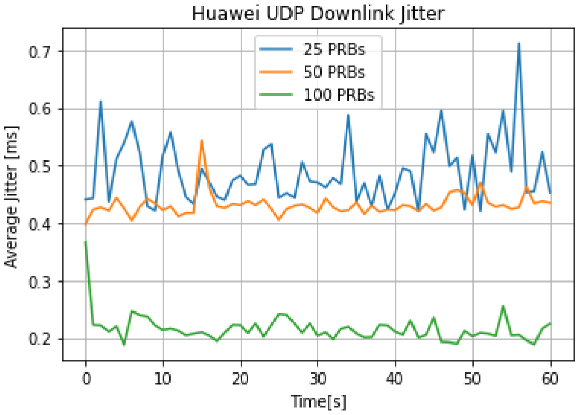


Figure 5.4: Huawei UDP Downlink Jitter

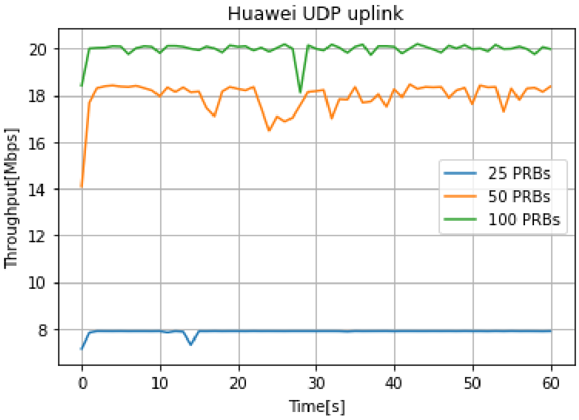


Figure 5.5: Huawei UDP Uplink

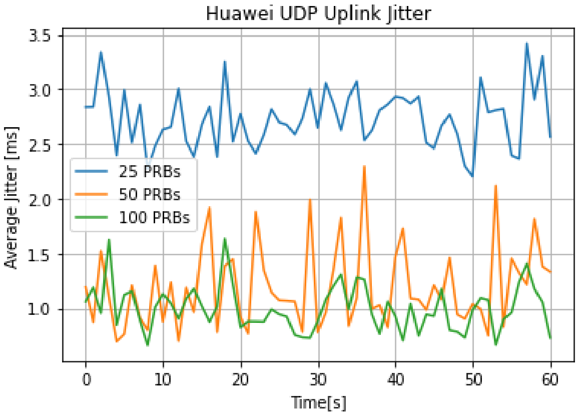


Figure 5.6: Huawei UDP Uplink Jitter

The results above are from throughput experiments carried out on the Huawei P40 lite 5G COTS UE. All the throughputs were as expected, from the figures provided by OpenAirInterface with some experiments performing even better, namely throughput experiments carried out using 100 physical resource blocks. The expected throughputs provided by OpenAirInterface are provided in [63].

Figure 5.1 and Figure 5.2 show the maximum TCP protocol downlink throughput the testbed is capable of performing at 25 PRBs, 50 PRBs and 100 PRBs respectively. The graphs show that doubling the number of resource blocks used approximately doubles the throughput.

Figure 5.1 shows an initial rise in throughput and as the experiment carries on, throughput finally settles down (levels out) at a value that is the maximum throughput the channel is capable of. The reason for this is due to the TCP ”Slow start”. [64] This is a congestion control algorithm that the TCP protocol initiates at the start of data transfer. To ensure no data packets are lost the TCP slow start congestion control algorithm operates by observing that the rate at which new packets should be injected into the network is the rate at which the acknowledgements (ACK) are returned by the other end. Hence slowly increasing the bandwidth for each transmission if an ACK is received and reducing the bandwidth if an ACK is not received until the optimal throughput is reached.

A resource block in 4G LTE system consists of 12 sub-carriers, hence increasing these would increase the bandwidth available on the channel, thus the increase in throughput observed for experiments with higher resource blocks.

It should be noted that the uplink throughputs are approximately half of the downlink link values at the same resource blocks. This is largely due to the modulation schemes employed. In LTE systems, the downlink (eNB to UE direction) modulation scheme is OFDMA which allows more sub-carriers to fit within a given bandwidth and improves spectral efficiency. However, in the uplink direction (UE to eNB) the modulation scheme is Single Carrier Orthogonal Frequency Division Multiple Access (SC-OFDMA). To reduce the cost of production, Mobile phones utilise the SC-OFDMA modulation scheme which achieves a lower PAPR (Peak-to-average-power ratio) than OFDMA, however, this comes at the cost of throughput. Hence the Uplink will always have lower throughputs than the downlink counterpart using the same number of resource blocks.

The same trend is again observed for the downlink UDP experiments, Figure 5.3 and Figure 5.3. The average jitter (figure 5.4, 5.6) is reduced by increasing the number of resource blocks used. It is to be noted that in the uplink direction, the jump to 100 PRBs did not exhibit the expected throughput which would be approximately double that at 50 PRBs. The expected theoretical uplink is 35Mbits/s however only a maximum of 20Mbits/s was achieved. This is due to the OpenAirInterface not being fully configured and optimised for the 100 PRBs uplink transmission.

5.1.2 iPhone COTS UE

The deployment scenario for the following tests is as shown in Figure 3.5 using the iPhone COTS UE.

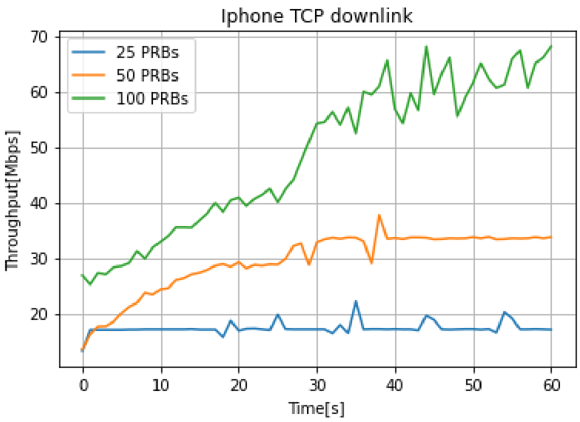


Figure 5.7: iPhone TCP downlink

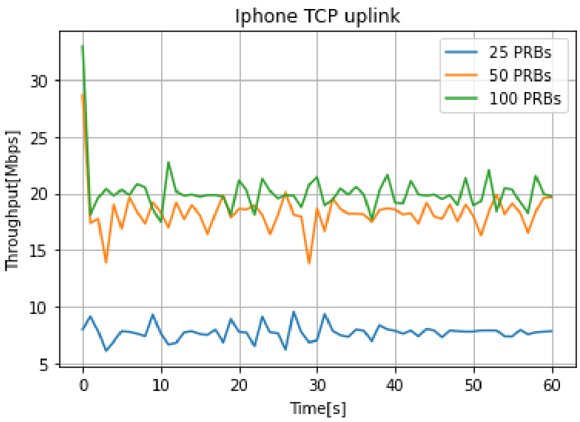


Figure 5.8: iPhone TCP Uplink

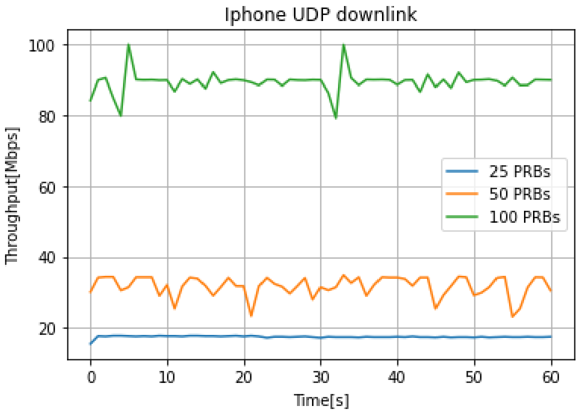


Figure 5.9: iPhone UDP downlink

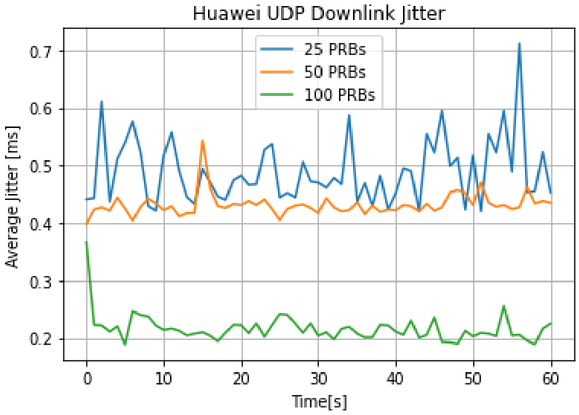


Figure 5.10: iPhone UDP Downlink Jitter

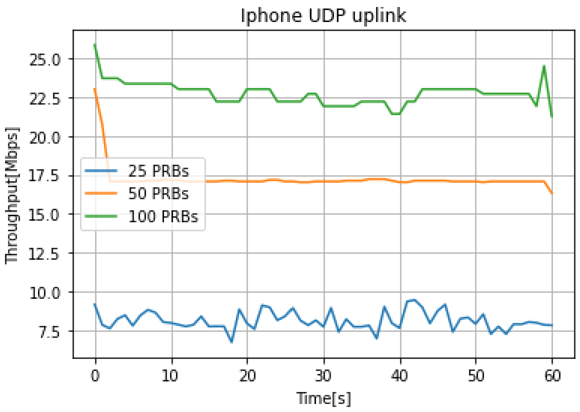


Figure 5.11: iPhone UDP Uplink

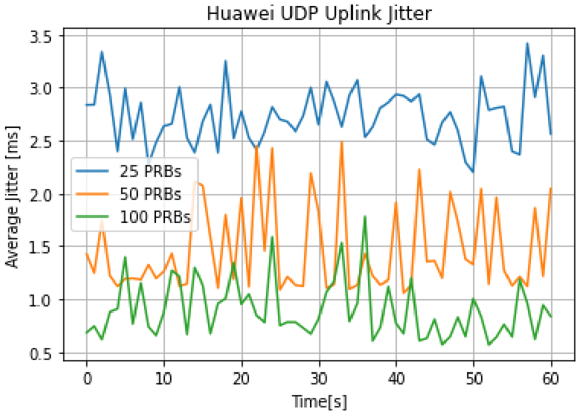


Figure 5.12: iPhone UDP Uplink Jitter

As seen by Figures 5.7 and 5.12, the iPhone COTS UE followed a similar trend to the

Huawei COTS UE in terms of throughput and jitter behaviours. Both exhibiting the slow start characteristic for TCP transmission and an almost doubling in throughput by doubling the number of resource blocks deployed (PRBs), as well as decreased jitter associated with more PRBs. Also similar to the Huawei COTS UE, the downlink throughput is approximately half the uplink throughput, this was expected due to the different modulation schemes used in the two scenarios as explained in section 5.1.1.

Another notable observation is the jitter comparison between the uplink and downlink directions. Downlink jitter ranged from 0.5ms at 25 PRBs to 0.2ms at 100PRBs which is multiple times less than the uplink jitter. Uplink jitter ranged from an average of 2.8ms at 25PRBs to 0.5ms and averaged just under 1ms at 100PRBs. The effect of jitter is on latency, essentially the jitter value is a reflection on the latency variation.

5.1.3 Raspberry PiHat UE

The Raspberry Pi hat UE is the decided upon COTS UE for drone flight tests. For this reason, it was used to perform further performance tests for 5G NSA and 5G SA. The results shown are for experiments run on the OAI test bed and the SRS RAN test bed. They also go on to further explore the performance of these implemented open source test-bed compared to the current commercial mobile network deployments in South Africa.

To standardize the experiments for comparison between SRS and OAI, these experiments were carried out with 50 PRBs as this was the only option currently offered by SRS RAN.

KPI	**MTN**	**SRS RAN**	**OAI**
DL Throughput [Mbps]	40	43.1	34.66
UL Throughput [Mbps]	30	9.8	18.0
E2e latency [ms]	23	11.6	16
RSSI [dBm]	-51 to -25	-51	-43

Table VI: 4G LTE Throughput results: Comparison between SRS, OAI and MTN (commercial), RPIHat COTS UE [8]

Compared to SRS RAN and MTN (commercial), OpenAirInterface lagged in the downlink throughput. Its performance on the RPiHat showed similar trends mentioned above. The Uplink throughput is half that of the downlink due to the the different modulation schemes used. However overall the best performing deployment was the MTN (commercial) solution. This is unsurprising as this mobile network has been fully implemented and optimised, and uses actual commercial eNBs (specialised hardware) which have better performance than the antennas used in the lab. The large latency exhibited by MTN is due to the distance from the eNB (Actual cell phone tower in this case.) This is also reflected in the large RSSI variation.

5.2 5G NSA Performance tests

KPI Metrics	MTN	SRS RAN	OAI
DL Throughput [Mbps]	252	38.5	84.5
UL Throughput [Mbps]	30	9.8	34.9
E2e latency [ms]	10	11.8	5
RSSI [dBm]	-51 to -25	-57	-51 to -25

Table VII: 5G NSA results comparison between MTN, SRS, OAI [8]

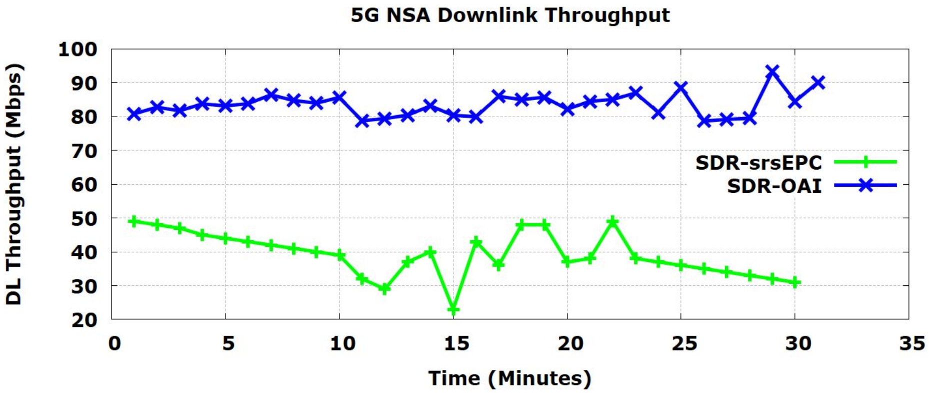


Figure 5.13: 5G NSA downlink throughput results [8]

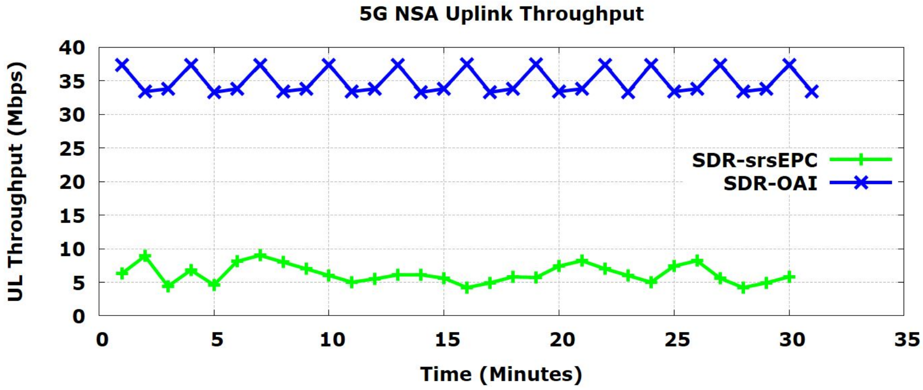


Figure 5.14: 5G NSA uplink throughput results [8]

Table VII, shows the superior performance improvement offered by 5G NSA over 4G LTE. Starting with OAI, uplink and downlink throughputs were doubled and the latency was reduced by approximately a factor of 3, going from 16ms to only 5ms. The latency is better than SRS and even the commercial MTN. Overall SRS RAN offered the least performance and least improvement from its 4G LTE deployments. The reason being it's still under development and currently behind OAI in the implementation of their 5G NSA architecture, especially on the gNB RAN side. It is also to be noted that these KPIs are to improve in the near future as they are nowhere close to the expected 5G NSA throughput KPIs (which are DL:1000Mbps, UL:500Mbps), a lot of development is still taking place with new updates being pushed every month.

5.3 5G SA Performance tests

KPI Metrics	**srsRAN, Open5GS**
DL Throughput [Mbps]	29.04
UL Throughput [Mbps]	5.04
E2e latency [ms]	12
RSSI [dBm]	-71

Table VIII: 5G SA measured KPIs [8]

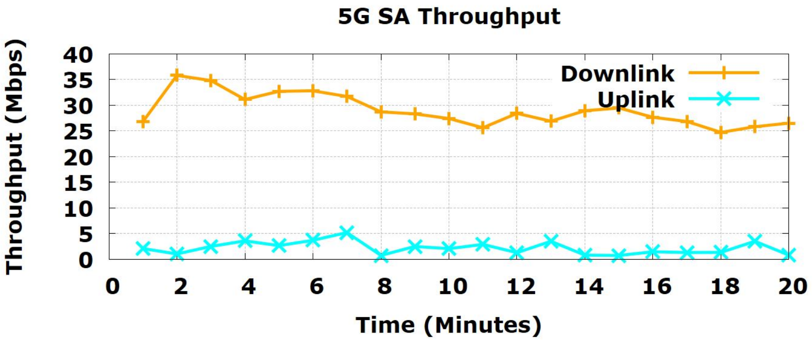


Figure 5.15: 5G SA downlink and uplink throughput results [8]

These ”True 5G” results are far from the expected results. Not only this but the 5G SA offered worse performance than 5G NSA and even the 4G deployments. For these 5G SA performance tests, srsRAN and Open5GS were used, for the gNB and 5GCore respectively. This was due to the fact that 5G SA is still in its beginning phases in terms of implementation, especially in the open-source industry. This combinational deployment was the most fruitful in 5G SA. 5G SA by OAI would work only for a few seconds before disconnection occurs. A lot of features are yet to be deployed for reasonable performance tests to be carried out. This is also the same for the implementation tabulated in Table VIII.

5.4 Initial flight tests (FR5)

This sub-section presents the results and main takeaways from the initial flight tests described in Section 3.

Range tests

By manually flying the drone using the remote controller and monitoring the eNB, and ground control station activity, one can pinpoint the time a connection is lost from the mobile network.

The maximum range was found to be between 25m and 35m. At which point the drone would disconnect from the mobile network. This event is evident from the logs on the RPi, when iPerf and RSSI are no longer being logged. The Mission Planner and its HUD (Heads Up Display) no longer gets updated and shows only the last known values until the connection is re-established completely. Flying the drone close to the base station $\leq$ 5m would allow the mobile network connection to be established. Figure 7.2 and 7.3 shows the drone UE disconnecting and reconnecting to the eNB respectively.

Further manual flying showed a connection between the flying speed and the maximum distance at which the connection could be maintained. The above-mentioned range of 25m to 35m was obtained by flying at lower speeds, less than 5 meters per second ($5ms^{-1}$). Speed greater than $10m^{-1}$ not only resulted in less range (20m to 25m), but also in a weaker channel connection as seen by several disconnection and re-connection attempts, even at close range.

One important takeaway from these initial flights to take note of is that there is a limited range within which the following automated missions can be conducted. Not only that but also speed plays an important role in our results. It would be ideal to fly at slower, more connection-robust speeds going forward. It will also be worthwhile to note the performance of the communication link at different speeds.

5.5 Flight Tests

After carrying out the initial flight tests, with the takeaways from that further, more organised tests were carried out as described in Section 3 and results were recorded and reported below.

5.5.1 Throughput, distance and speed (FR1, FR2)

As described in the implementation, Section 3 throughput tests were carried out during flights. TCP thoroughput tests showed similar results to those presented in Section 5.1.3. The reason is that the maximum throughput the connection is capable of is independent of the distance of the UE from the base station. The capability remains the same, however, RSSI results show that the connection quality does reduce. Therefore, UDP tests were carried out to monitor the connection channel quality. One can discern this information by running UDP test specifying the throughput as the maximum throughput reported by TCP iPerf tests (in this case 70Mbps) and monitoring the `Lost/Total Datagrams` information reported as a percentage. These results were aggregated and reported in Figure 5.16.

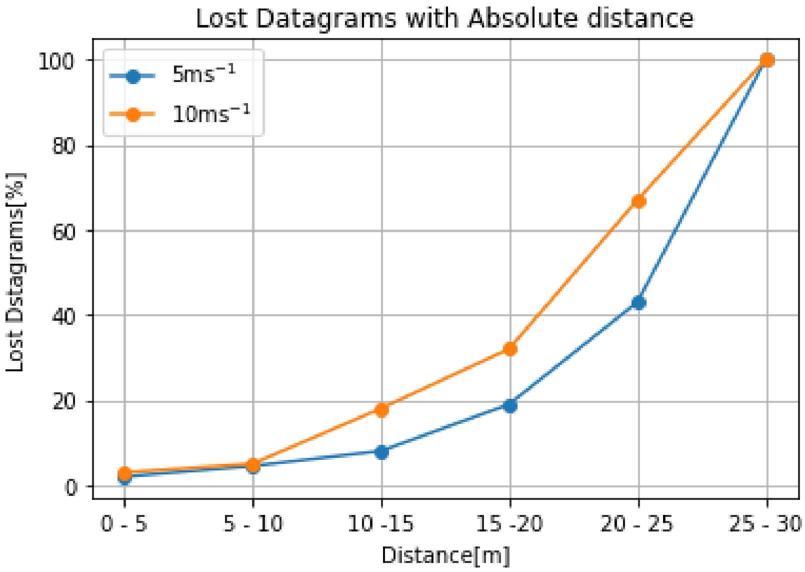


Figure 5.16: Lost Datagrams with absolute distance from the base station to Drone UE

The results show an increasing loss in data packets, the further one moves away from the base station. This is consistent with the reduction in RSSI as well. Poorer channel conditions mean more data packets are lost. As the Drone UE disconnected at distances greater than 25m, this is reported as 100% data packet loss.

Regarding speed, the same trend is visible, but at a distance above 10m the loss rate is way higher for 10ms^{-1} than for 5ms^{-1} flights. This is also coherent with the exponential RSSI reduction. The channel quality is relatively good between 0m and 10m and then reduces drastically after that.

5.5.2 RSSI and height (FR3)

As described above, flights were carried out at different heights. At 5m, 10m, 15m, 20m and 25m. The data was observed, monitoring the change in RSSI with height.

Figure 5.17 shows the RSSI values read from the PiHat during flights carried at 5m heights. Of course, these values need to be converted to RSSI values in dBm using the graph created in 4.2. Similar graphs were obtained for flights at 10m and 15m altitudes and likewise converted to RSSI in dBm values. These results are summarised below in Figure 5.18.

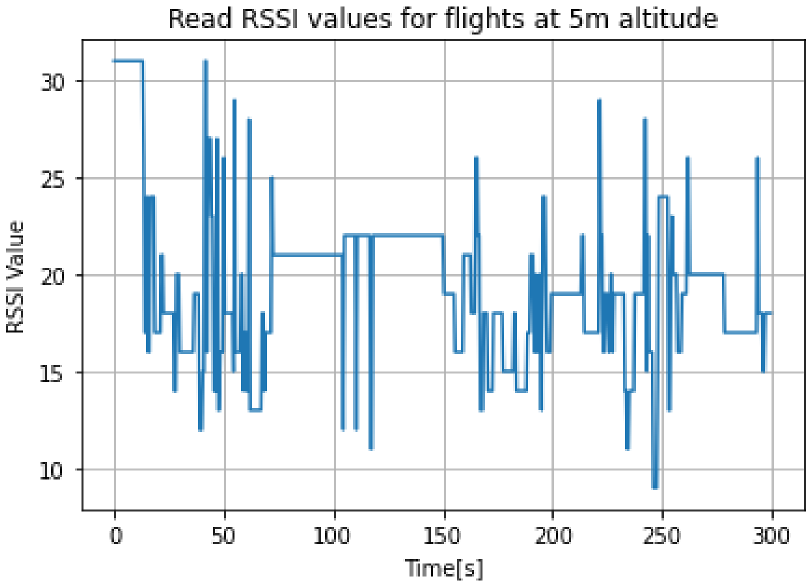


Figure 5.17: Read RSSI values at 5m altitude

From Figure 5.17 it is evident that the RSSI values are highly fluctuating with time (between an unknown highly negative $< -133dBm$ and an unknown less negative $> -56dBm$) and hence quite difficult to visualise this variation with distance flown. Grouping the data together makes visualising this a bit easier.

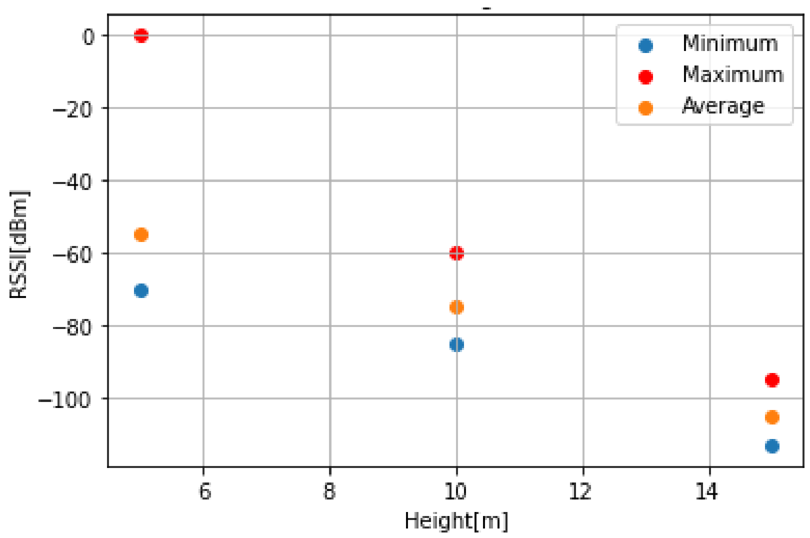


Figure 5.18: RSSI values for flights at 5m, 10m and 15m. Flight speed 5ms^{-1}

Figure 5.18 shows the Minimum, Maximum and average RSSI values for flights carried out at different heights,(5m, 10m, 15m). The set waypoint speed for these missions was 5ms^{-1}.

It is evident that although the RSSI value fluctuates every second this happened within a certain range of RSSI values. And the general trend shows that the higher the flight altitude, the RSSI became more negative. Indicating poorer channel conditions. This makes sense because the higher the altitude, the further away you were from the base station, which was placed on the ground.

Automated flights at 20m were carried out several times, and no data is available for these. The reason being the drone UE would disconnect from the mobile network. This is due to the poor channel conditions further but this is not coherent with other data obtained. For example, RSSI values at a distance greater than 20m were recorded, so this disconnection was not entirely due to the large distance.

Upon further investigations and log analysis. A trend confirmed that these disconnections were due to the flight speed of the set auto-mission flights. More disconnections were realised at 10ms^{-1} as compared to 5ms^{-1} flights.

5.5.3 RSSI and speed (FR4)

The exact same automated missions were carried out at the different flight speeds by changing the way-point speed (Parameter: `wp_speed` in Mission Planner).

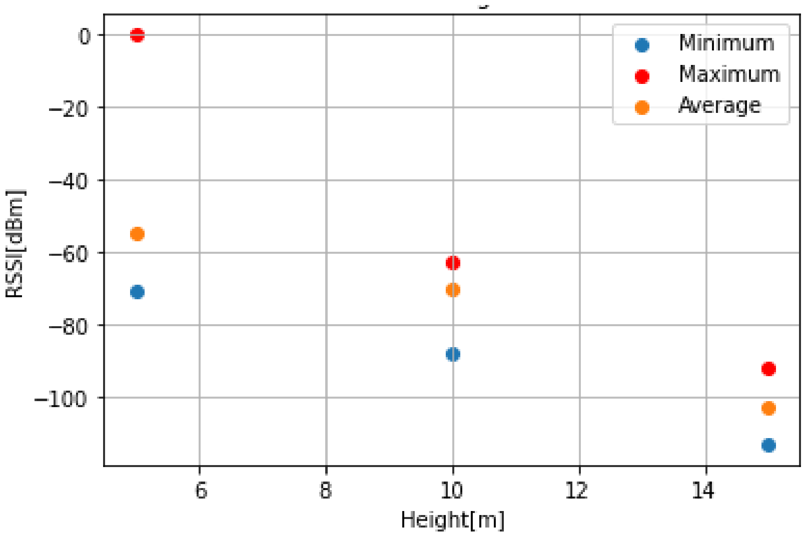


Figure 5.19: RSSI values for flights at 5m, 10m and 15m, Flight speed 10ms^{-1}

Figure 5.19 shows the RSSI values at a speed of 10ms^{-1}. It can be seen that these values are similar and not significantly different from the 5ms^{-1} flights in Figure 5.18. Automated flights were also carried out at 15ms^{-1}, and data for these is unavailable as the drone UE would disconnect, before even starting the auto missions. The takeaway from this is that at such speeds, interference, signal strength and the Doppler frequency shift have a significant part to play in this Base Station to UE connection, especially when compounded together.

Further probing into the logs and some manual flights suggests that the parameters `wp_acc` (Waypoint acceleration) had an important effect on the mobile network connection. Accelerating to the starting point of an auto mission causes a current spike. (The ESCs draw more current from the batteries to meet the speed-up requirements) This higher current draw definitely creates interference, the extent of this interference on the mobile network connection requires further investigation. Antenna positioning was done to reduce this effect. (Antennas were placed as away as possible from the propellers and the ESCs) Antenna orientation also needs further investigation.

5.5.4 RSSI and absolute distance (FR3)

Results for the RSSI and absolute distance (straight line distance from the base to the drone) showed similar results to those shown in Figure 5.18. The greater the distance from the base station the greater the loss in the received strength (RSSI). This data has been grouped at 5m distance intervals and is summarised in the table below, Table IX.

	RSSI[dBm]		
Distance	**Minimum**	**Maximum**	**Average**
0 - 5	-70	-56	-56
5 - 10	-85	-60	-75
10 -15	-103	-82	-100
15 -20	-113	-95	-105
20 - 25	-113	-100	-112
25 - 30	🟥		

Table IX: RSSI with absolute distance from base station

The table above shows the mentioned trend, consistent at all heights. It is to be noted that, according to the extrapolation table some of the actual RSSI values are not known. The only information known is the range. (-56 dBM colour filled green, indicates a range of -56 dBm to 0 dBm. -113 dBm colour-filled orange indicates a range of -113 dBm or less) No data was available for distances above 25m due to disconnections from the mobile network. No substantial data could be logged. The mentioned trend can be seen in Figure 5.20.

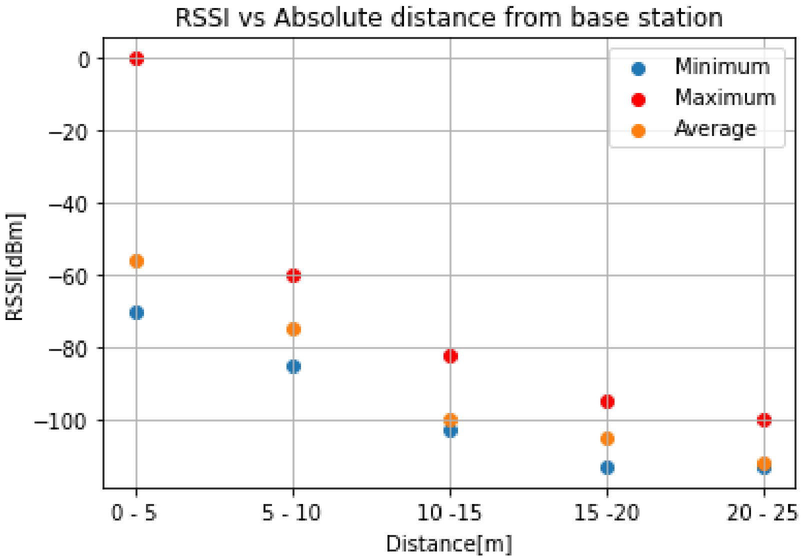


Figure 5.20: RSSI vs Absolute distance

The graph indicates the expected exponential drop in RSSI with an increase in distance. (Especially for 0m to 15m distance) After 15m the exponential drop cannot be seen in the graph, this is because of the above-mentioned reason that the PiHat does not report RSSI below -133 dBm, it is likely possible that this trend still holds up until disconnection at distances $\geq 25m$.

5.6 Typical use case applications

With reference to Table I, this sub-section discusses the currently possible use cases that the above-implemented solution can achieve, based on the results obtained from the experiments.

5.6.1 4G LTE testbed and flight tests

Performing at 100 PRBS OAI, with an average uplink rate of 30 Mbps, (roughly double the uplink rate at 50 PRBs. This setup is capable of 4K HD Video which is up to Level 3 on the data rates scale. For latency, both Level 1 and Level 2 have been achieved, this translates to image/video transmission and remote real-time control application. Flight tests showed that there is a need for improvement as applications requiring 50m or more of coverage, both altitude and wide area coverage cannot be achieved.

5.6.2 5G NSA testbed and flight tests

Given the better performance over 4G, in terms of use case applications, with reference to Table II all the specified use case applications could be achieved. Including 4K drone filming, which was not achieved by 4G. However, this was not translated to flights, as it was not possible to take this 5G NSA setup to flight. Having a fragile connection channel, the range and robustness of the channel are challenges that need to be overcome.

5.6.3 5G SA

The state of 5G SA is that it is still in its very immature. As of right now, one can only expect a working connection, uplink throughput is good enough for applications that do not require much data to be sent, and telemetry-only data can be handled. The end-to-end latency is still much larger than 5G NSA OAI and roughly the same as MTN and SRS.

5.7 Baseline Resource Utilisation (IR4)

Baseline resource utilisation measurements were taken to gain insights into how "heavy" an application Openairinterface can be. This information is useful so as to know which processes can be more power and memory-intensive. As show in Figure 3.15, the Openairinterface EPC is comprised of five containers, to run the complete EPC these docker containers have to be spun up separately in a specific order. This sub-section shows the effect of each container on the CPU and memory usage (RAM) and ultimately the CPU and RAM usage of the entire EPC.

CPU and RAM usages were monitored at and around two instances in time, (i) when starting the docker containers on the EPC machine, (ii) running the corresponding EPC service within each Docker container, (iii) eNB initialisation and finally (iv) at each UE attachment. The results are shown below. Figure 5.21 shows the baseline CPU and RAM usage of the EPC machine.

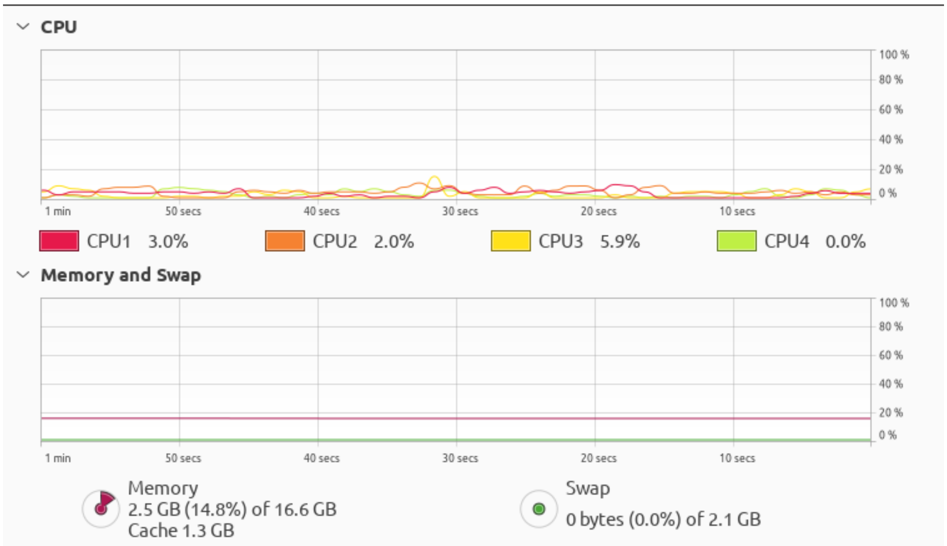


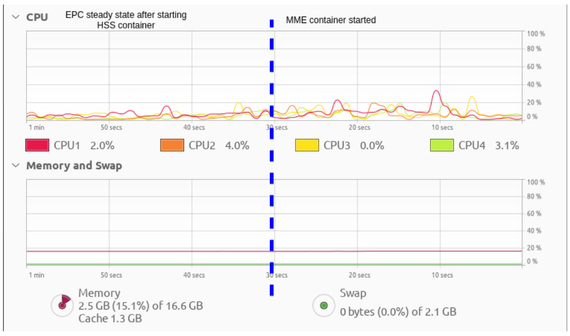
Figure 5.21: Baseline CPU and RAM usage of the EPC machine at steady state.

5.7.1 EPC container initialisation

EPC steady state

HSS container started

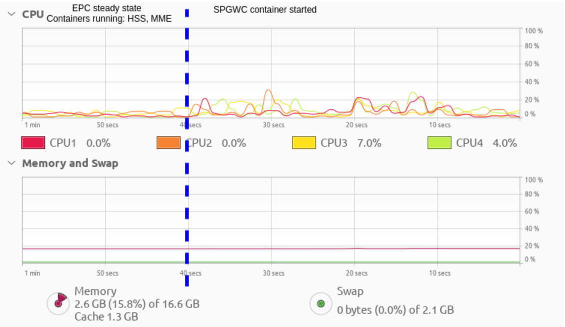
Figure 5.22: Starting HSS container



EPC steady state after starting HSS container

MME container started

Figure 5.23: Starting MME container after HSS



EPC steady state
Containers running: HSS, MME

SPGWC container started

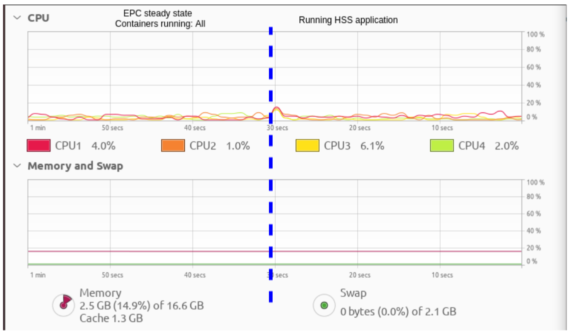
Figure 5.24: Starting SPGWC container

EPC steady state
Containers running: HSS, MME, SPGWC

SPGWU container started

Figure 5.25: Starting SPGWU container

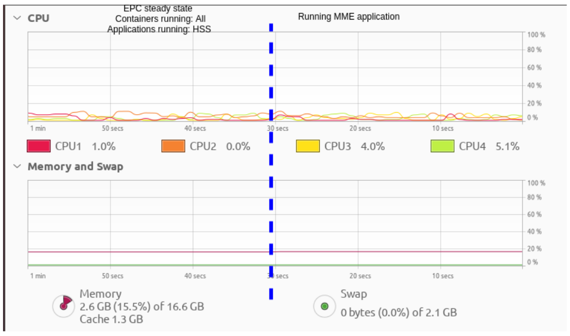
5.7.2 EPC applications initialisation



EPC steady state
Containers running: All

Running HSS application

Figure 5.26: Running HSS application



EPC steady state
Containers running: All
Applications running: HSS

Running MME application

Figure 5.27: Running MME application

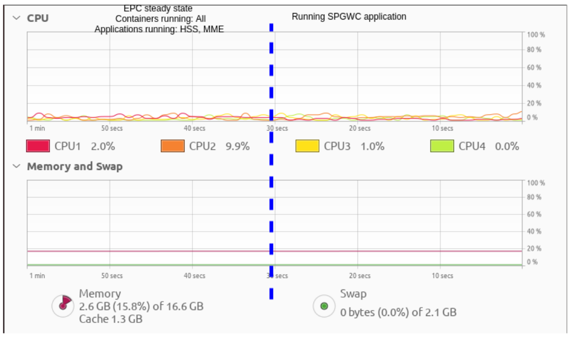


Figure 5.28: Running SPGWC application

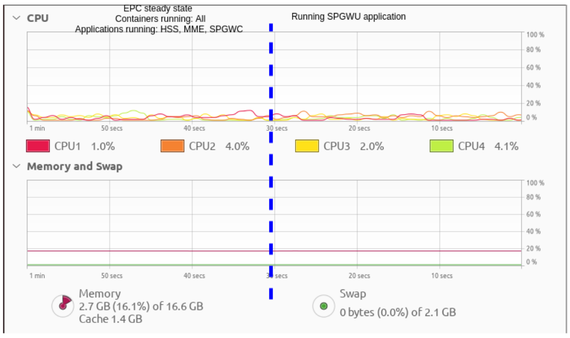


Figure 5.29: Running SPGWU application

Figures 5.21 to 5.29 above show the EPC resource usage from steady state (the baseline usage) to docker container start-ups to the application start-ups. From the images above it can be deduced that the OpenAirInterfaces' EPC is resource-efficient. Much of the resource utilisation took place in the start-ups of the docker containers, showing a slight increase in the usage of CPU threads (at most 25% CPU usage). Running the actual application had an almost negligible effect on the resource usage and cache usage remained constant throughout with no change from the baseline resource usage.

6 Conclusions and Recomendations for future work

The main objective of this research was to deliver a proof-of-concept communication stack that leverages the use of Mobile Networks as the mode of communication for drone telemetry and payload data. This work provided insight into the possibility of the drone industry adopting such a solution, what may be gained and the potential challenges that need to be overcome to ultimately address the unmet needs in the industry.

Following the results discussed in Section 5 this section discusses the extent to which the project was successful and unsuccessful, regarding implementation and takeaways from the indoor and flights carried out. The contributions made, as well as the limitations involved.

- The designed software stack was able to manage drone communication for flights. Using 4G LTE the flights carried out provide information on the performance of such a solution. This architecture design is transferable to 5G NSA and 5G SA. Not only can it be used for drone flights but also for many other applications that require mobile network communication.
- A telemetry connecting between a drone and ground control station was successfully deployed using OpenAirInterface (As the open source mobile network) and the commonly used drone autopilot, the cube orange alongside a Raspberry Pi as the on board computer. This connection was able to maintain sufficiently low latencies and bandwidths to maintain operational flight provided the drone remained with 25m in distance and 10ms^{-1} in speed.
- The operational 4G telemetry connection technical stack is documented and all supporting code published on GitHub. (Link in the Appendix)
- It was determined that speed of flight has a significant impact on the network connection. Specifically fly at speeds greater than 0ms^{-1} causes a lot disconnections to happen. These disconnections form the mobile network could be caused by acceleration and further research needs to be conducted to ascertain this.

Some limitations of the study were that flight experiments were only carried out using 4G LTE. 5G NSA and 5G SA were still too early in the development phase to carry out any flights, once finished the software stack can be easily integrated with these for further experiments. The 5G Mobile network stack still needs improvements to make the connections more stable and improve the range.

6.1 Recommendations for future work

This section discusses additional future work and experimentation which can be done to improve the design presented and experiments carried out and not carried in this dissertation.

These suggestions are the next steps forward following this research.

6.1.1 Testbeds

Following 4G LTE and 5G NSA test-bed implementations, the 5G SA could not be realised as it is still under development. Once this is done further experiments and performance testing can be done on the 5G SA testbed. The results from these experiments can be compared to the drone requirements and see if these requirements are met. Theoretically, they should but that's not always the case in practical implementations.

These experiments will make way for investigations into the new technologies that 5G has to offer, including uRLLC, eMBB and mMTC and the effect these will have on the drone industry as well the feasibility of use and or challenges that may need to be overcome to realise the use of such technologies.

Initial proposed solution

This solution, proposed in Section 3.5.2 was implemented and found to be working. However, it was not ideal for this research as this research is focused more on a self-contained mobile network. (i.e with no access to the outside world/ other networks)

Further research and experiments using this Initial Proposed architecture would help gain insight into drone command and control without any geographical limitations and should provide interesting insight into the use cases of such an application.

6.1.2 Flights

As it was not possible to carry out any practical flight tests with 5G NSA due to the fact that the current state of its implementation is highly fragile. This needs to be further explored.

This includes looking at alternative hardware, such as antennas or using dedicated gNB, eNB base stations for better channel conditions and RAN processing so as to make the channel robust enough to carry out drone flights similar to the ones done in this research.

Following a successful 5G SA testbed setup, drone flight would need to be carried out as well so as to assess the performance of 5G SA in flight.

Acceleration

Drawing from the conclusions above, the effect of drone acceleration needs to be investigated, with special attention to the interference it causes to the communication channel due to the current draw from the propellers.

References

[1] B. Tingnan, "The architectural differences between lte and wimax," 2013.

[2] What is 5g network slicing? (accessed: 29/01/2023). [Online]. Available: https://www.sdxcentral.com/5g/definitions/key-elements-5g-network/5g-network-slicing/

[3] H. Remmert. What is 5g network architecture? (accessed: 29/01/2023). [Online]. Available: https://www.digi.com/blog/post/5g-network-architecture

[4] What is a numerology? (accessed: 08/02/2022). [Online]. Available: https://apistraining.com/5g-numerology/

[5] Rf and 5g new radio: top 5 questions answered. (accessed: 27/01/2023). [Online]. Available: https://www.exfo.com/es/recursos/blog/rf-5g-new-radio-top-5-questions/

[6] M. Bello, "5g network: Expectation vs. reality," 02 2021.

[7] Openairinterface home github. (accessed: 12/07/2021). [Online]. Available: https://github.com/OPENAIRINTERFACE/openair-epc-fed

[8] M. Chepkoech, N. Mombeshora, B. Malila, and J. Mwangama, "Evaluation of open-source mobile network software stacks: A guide to low-cost deployment of 5g testbeds," in *2023 18th Wireless On-Demand Network Systems and Services Conference (WONS)*, 2023, pp. 56–63.

[9] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on uav communications for 5g and beyond," *Proceedings of the IEEE*, vol. 107, no. 12, pp. 2327–2375, 2019.

[10] Y. Zeng, I. Guvenc, R. Zhang, G. Geraci, and D. W. Matolak, *UAV Communications for 5G and Beyond*. John Wiley & Sons, 2020.

[11] G. Yang, X. Lin, Y. Li, H. Cui, M. Xu, D. Wu, H. Rydén, and S. B. Redhwan, "A telecom perspective on the internet of drones: From lte-advanced to 5g," 2018.

[12] P. K. Sharma, "Evolution of mobile wireless communication networks-1g to 5g as well as future prospective of next generation communication network," 2013.

[13] At command munual. (accessed: 02/02/2023). [Online]. Available: https://www.waveshare.com/w/upload/1/17/SIM8200_Series_AT_Command_Manual_V1.00.01_0515.pdf

[14] T. Lagkas, V. Argyriou, S. Bibi, and P. Sarigiannidis, "Uav iot framework views and challenges: Towards protecting drones as "things"," *Sensors*, vol. 18, no. 11, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/11/4015

[15] Insider. Drone technology uses and applications for commercial, industrial and military drones in 2021 and the future? (accessed: 19/03/2021). [Online]. Available: https://www.businessinsider.com/drone-technology-uses-applications?IR=T

[16] M. Marchese, A. Moheddine, and F. Patrone, "Iot and uav integration in 5g hybrid terrestrial-satellite networks," *Sensors*, vol. 19, no. 17, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/17/3704

[17] P. J. Burke, "A safe, open source, 4g connected self-flying plane with 1 hour flight time and all up weight (auw) lt;300 g: Towards a new class of internet enabled uavs," *IEEE Access*, vol. 7, pp. 67 833–67 855, 2019.

[18] N. Mombeshora. Leveraging next generation cellular networks for drone telemetry and payload communication ngonidzashe mombeshora. (accessed: 09/02/2023). [Online]. Available: https://www.youtube.com/watch?v=nm0Vuc32Q_I

[19] The evolution of drones: From military to hobby commercial. (accessed: 07/07/2021). [Online]. Available: https://percepto.co/the-evolution-of-drones-from-military-to-hobby-commercial/

[20] Law enforcement guidance for suspected unauthorized uas operations. (accessed: 25/01/2022). [Online]. Available: https://www.faa.gov/uas/resources/policy_library/media/Sec_331_336_UAS.pdf

[21] Drone regulations in south africa. (accessed: 07/09/2023). [Online]. Available: https://www.starliteaviation.com/drones/drone-regulations-in-south-africa/

[22] X. Lin, V. Yajnanarayana, S. D. Muruganathan, S. Gao, H. Asplund, H. Maattanen, M. Bergstrom, S. Euler, and Y. . E. Wang, "The sky is not the limit: Lte for unmanned aerial vehicles," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 204–210, 2018.

[23] A. Bitar, A. Jamal, H. Sultan, N. Alkandari, and M. El-Abd, "Medical drones system for amusement parks," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, 2017, pp. 1–10.

[24] A. Mukherjee, N. Dey, and D. De, "Edgedrone: Qos aware mqtt middleware for mobile edge computing in opportunistic internet of drone things," *Computer Communications*, vol. 152, pp. 93–108, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366419315750

[25] D. C. Campbell. The potential of drones in research. (accessed: 26/01/2022). [Online]. Available: https://www.environmentalbiophysics.org/the-potential-of-drones-in-research/

[26] P. A. Nakao. Beyond 5g/6g telecommunication ensuring continuity in business, research and education. (accessed: 26/01/2022). [Online]. Available: https://www.itu.int/en/ITU-T/academia/kaleidoscope/2020/Documents/Presentations/Keynote%203_Akihiro%20Nakao.pdf

[27] M. L. Pamela Cohn, Alastair Green and M. Roller. Commercial drones are here: The future of unmanned aerial systems. (accessed: 26/01/2022). [Online]. Available: https://www.mckinsey.com/industries/travel-logistics-and-infrastructure/our-insights/commercial-drones-are-here-the-future-of-unmanned-aerial-systems

[28] B. Dash, M. F. Ansari, and S. Swayamsiddha, "Fusion of artificial intelligence and 5g in defining future uav technologies - a review," in *2023 International Conference on Device Intelligence, Computing and Communication Technologies, (DICCT)*, 2023, pp. 312–316.

[29] Y. Arjoune and S. Faruque, "Artificial intelligence for 5g wireless systems: Opportunities, challenges, and future research directions," 09 2020.

[30] I. T. U. R. communication Study Group *et al.*, "Characteristics of unmanned aircraft systems and spectrum requirements to support their safe operation in non-segregated airspace (report itu-r m. 2171)," *International Telecommunication Union Publications*, 2009.

[31] A. I. Sehgal. Technology on wings- drone regulation in india. (accessed: 31/01/2022). [Online]. Available: https://www.lawyered.in/legal-disrupt/articles/technology-wings-drone-regulation-india/

[32] Difference between 1g and 2g protocols. (accessed: 26/01/2023). [Online]. Available: https://www.tutorialspoint.com/difference-between-1g-and-2g-protocols

[33] 1g mobile phones features , uses , advantages and disadvantages. (accessed: 26/01/2023). [Online]. Available: https://www.online-sciences.com/technology/1g-mobile-phones-features-uses-advantages-and-disadvantages/

[34] A. H. Khan, M. A. Qadeer, J. A. Ansari, and S. Waheed, "4g as a next generation wireless network," in *2009 International conference on future computer and communication*. IEEE, 2009, pp. 334–338.

[35] Evolution of mobile technology. (accessed: 19/01/2023). [Online]. Available: https://iot.telenor.com/technologies/evolution-mobile-technology/

[36] Evolution of wireless technologies 1g to 5g in mobile communication. (accessed: 26/01/2023). [Online]. Available: https://www.rfpage.com/evolution-of-wireless-technologies-1g-to-5g-in-mobile-communication/

[37] A. Kumar, “3g networks: Opportunities and challenges,” *Bulletin of Mathematical Sciences and Applications*, vol. 3, pp. 28–36, 04 2013.

[38] J. Blumenstein, J. C. Ikuno, J. Prokopec, and M. Rupp, “Simulating the long term evolution uplink physical layer,” in *Proceedings ELMAR-2011*, 2011, pp. 141–144.

[39] Lte architecture concepts. (accessed: 27/01/2023). [Online]. Available: https://yatebts.com/documentation/concepts/lte-concepts/

[40] Y. Chen and X. Lagrange, “Architecture and Protocols of EPC-LTE with relay,” p. 25, 2013. [Online]. Available: https://hal.science/hal-00830621

[41] Y. Zeng, J. Lyu, and R. Zhang, “Cellular-connected uav: Potential, challenges and promising technologies,” *IEEE Wireless Communications*, vol. PP, 04 2018.

[42] H. Ullah, N. Gopalakrishnan 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.

[43] H. Koumaras, G. Makropoulos, M. Batistatos, S. Kolometsos, A. Gogos, G. Xilouris, A. Sarlas, and M.-A. Kourtis, “5g-enabled uavs with command and control software component at the edge for supporting energy efficient opportunistic networks,” *Energies*, vol. 14, no. 5, p. 1480, 2021.

[44] Lte physical layer overview. (accessed: 27/01/2023). [Online]. Available: https://rfmw.em.keysight.com/wireless/helpfiles/89600b/webhelp/subsystems/lte/content/lte_overview.htm

[45] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, “Survey on uav cellular communications: Practical aspects, standardization advancements, regulation, and security challenges,” *IEEE Communications surveys & tutorials*, vol. 21, no. 4, pp. 3417–3442, 2019.

[46] 5g sa vs nsa: Difference between standalone and non-standalone 5g architecture. (accessed: 27/01/2023). [Online]. Available: https://beebom.com/sa-vs-nsa-5g/

[47] A modern mobile core network solution. (accessed: 08/02/2023). [Online]. Available: https://magmacore.org/

[48] What is free5gc?: Homepage. (accessed: 08/02/2023). [Online]. Available: https://www.free5gc.org/

[49] Open source implementation for 5g core and epc, i.e. the core network of lte/nr network (release-16). (accessed: 08/02/2023). [Online]. Available: https://open5gs.org/

[50] srsran homepage. (accessed: 12/07/2021). [Online]. Available: https://www.srslte.com/

[51] S. V. Ahamed, "7 - ngmns, 3g, and 4g networks," in *Intelligent Networks*, S. V. Ahamed, Ed. Oxford: Elsevier, 2013, pp. 127–140. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780124166301000078

[52] Landrs science drone. (accessed: 07/11/2022). [Online]. Available: https://github.com/landrs-toolkit/LANDRs-Science-Drone

[53] Mavlink protocol. (accessed: 02/11/2022). [Online]. Available: https://mavlink.io/en/

[54] Openairinterface epc fed. (accessed: 07/07/2021). [Online]. Available: https://github.com/OPENAIRINTERFACE/openair-epc-fed

[55] Pixhawk overview. (accessed: 08/11/2021). [Online]. Available: https://ardupilot.org/copter/docs/common-pixhawk-overview.html

[56] "Mission home planner," (accessed: 29/03/2021). [Online]. Available: https://ardupilot.org/planner/

[57] "Qgroundcontrol," (accessed: 29/03/2021). [Online]. Available: http://qgroundcontrol.com/

[58] The cube orange with adsb-in overview¶. (accessed: 02/02/2023). [Online]. Available: https://ardupilot.org/copter/docs/common-thecubeorange-overview.html

[59] Amazon aws ec2. (accessed: 10/01/2022). [Online]. Available: https://aws.amazon.com/ec2/?did=ft_card&trk=ft_card

[60] Mavlink-router github. (accessed: 10/01/2022). [Online]. Available: https://github.com/mavlink-router/mavlink-router

[61] N. Mombeshora and J. Wyngaard, "Drone NR." [Online]. Available: https://github.com/landrs-toolkit/Drone-NR

[62] Haversine formula to find distance between two points on a sphere. (accessed: 03/02/2023). [Online]. Available: https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere/

[63] Openairinterface featrue set. (accessed: 07/10/2021). [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/blob/develop/doc/FEATURE_SET.md

[64] W. Stevens *et al.*, "Tcp slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," 1997.

7 Appendix

7.1 Flights

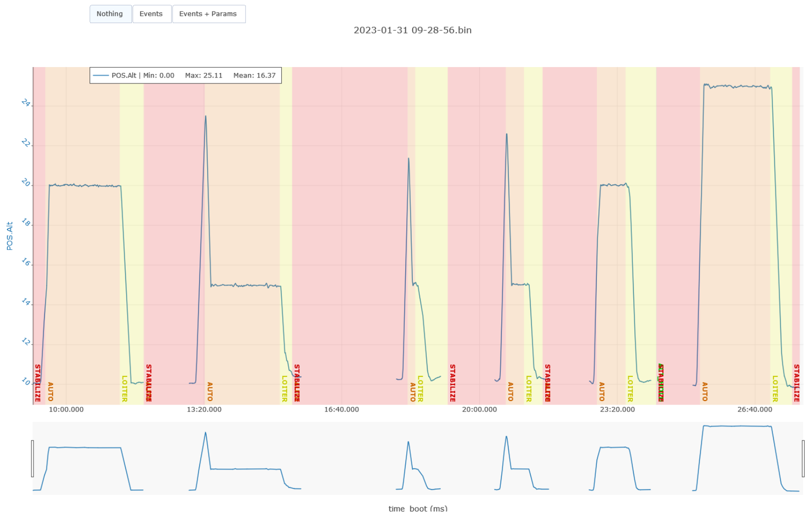
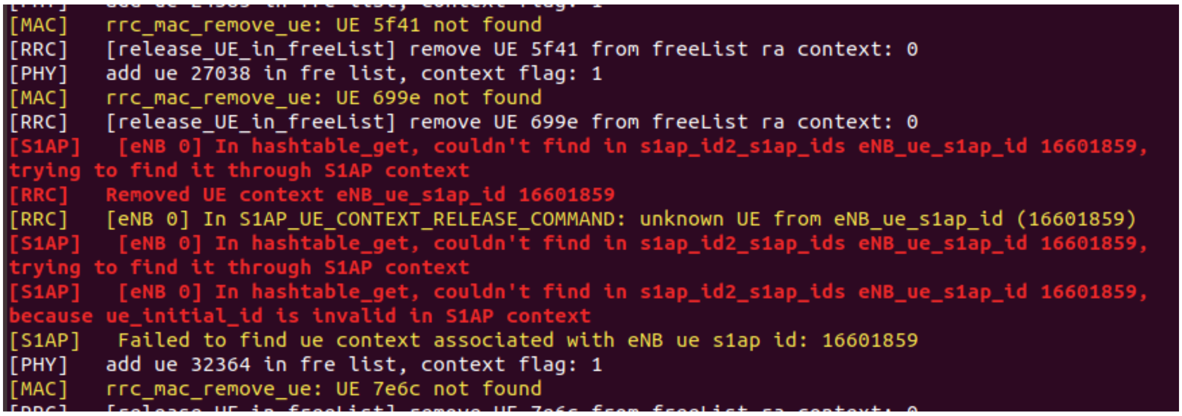


Figure 7.1: Flight altitude with time showing several flights carried out. In this case at 15m, 20, and 25m.

Height information essentially gives the time period when the drone was in air. This data can now be separated and analysed according to the specific height being flown.

7.2 Results



```
[MAC]   rrc_mac_remove_ue: UE 5f41 not found
[RRC]   [release_UE_in_freeList] remove UE 5f41 from freeList ra context: 0
[PHY]   add ue 27038 in fre list, context flag: 1
[MAC]   rrc_mac_remove_ue: UE 699e not found
[RRC]   [release_UE_in_freeList] remove UE 699e from freeList ra context: 0
[S1AP]   [eNB 0] In hashtable_get, couldn't find in s1ap_id2_s1ap_ids eNB_ue_s1ap_id 16601859,
trying to find it through S1AP context
[RRC]   Removed UE context eNB_ue_s1ap_id 16601859
[RRC]   [eNB 0] In S1AP_UE_CONTEXT_RELEASE_COMMAND: unknown UE from eNB_ue_s1ap_id (16601859)
[S1AP]   [eNB 0] In hashtable_get, couldn't find in s1ap_id2_s1ap_ids eNB_ue_s1ap_id 16601859,
trying to find it through S1AP context
[S1AP]   [eNB 0] In hashtable_get, couldn't find in s1ap_id2_s1ap_ids eNB_ue_s1ap_id 16601859,
because ue_initial_id is invalid in S1AP context
[S1AP]   Failed to find ue context associated with eNB ue s1ap id: 16601859
[PHY]   add ue 32364 in fre list, context flag: 1
[MAC]   rrc_mac_remove_ue: UE 7e6c not found
```

Figure 7.2: Mobile network disconnection, as seen at the eNB machine

```
ity index 0 MCC 655 MNC 25
[S1AP]   Found usable eNB_ue_s1ap_id: 0xf514fb 16061691(10)
[S1AP]   S_TMSI_PRESENT
[RRC]   [eNB 0] Received S1AP_INITIAL_CONTEXT_SETUP_REQ: ue_initial_id 16, eNB_ue_s1ap_id 16061
691, nb_of_e_rabs 1
[GTPU]   [102] Created tunnel for RNTI 90ac, teid for DL: a6283a80, teid for UL 1 to remote IPv
4: 192.168.61.133, IPv6 ::
[RRC]   [FRAME 00000][eNB][MOD 00][RNTI 90ac] rrc_eNB_process_GTPV1U_CREATE_TUNNEL_RESP tunnel
(2787654272, 2787654272) bearer UE context index 0, msg index 0, id 5, gtp addr len 4
[RRC]   [eNB 0][UE 90ac] Selected security algorithms (0x7f79cc00c570): 0, 2, changed
[RRC]   [eNB 0][UE 90ac] Saved security key DD67921B001144E7FCBBD4624498C8DB0F19FFD520ABF0EE446
2B7C733992843
[RRC]   [FRAME 00000][eNB][MOD 00][RNTI 90ac] Logical Channel DL-DCCH, Generate SecurityModeCom
mand (bytes 3)
[RRC]   calling rrc_data_req :securityModeCommand
[RRC]   sent RRC_DCCH_DATA_REQ to TASK_PDCP_ENB
[PHY]   add ue 52380 in fre list, context flag: 1
[MAC]   rrc_mac_remove_ue: UE cc9c not found
[RRC]   [release_UE_in_freeList] remove UE cc9c from freeList ra context: 0
[RRC]   [FRAME 00000][eNB][MOD 00][RNTI 90ac] received securityModeComplete on UL-DCCH 1 from U
E
<FreqBandList>
        <bandInformationEUTRA>
            <bandEUTRA>7</bandEUTRA>
        </bandInformationEUTRA>

        <bandInformationNR>
            <bandNR>78</bandNR>
        </bandInformationNR>

</FreqBandList>
[RRC]   [FRAME 00000][eNB][MOD 00][RNTI 90ac] Logical Channel DL-DCCH, Generate UECapabilityEnq
uiry (bytes 10)
[RRC]   sent RRC_DCCH_DATA_REQ to TASK_PDCP_ENB
[PHY]   add ue 6316 in fre list, context flag: 1
[MAC]   rrc_mac_remove_ue: UE 18ac not found
[RRC]   [release_UE_in_freeList] remove UE 18ac from freeList ra context: 0
[RRC]   [FRAME 00000][eNB][MOD 00][RNTI 90ac] received ueCapabilityInformation on UL-DCCH 1 fro
m UE
[RRC]   got UE capabilities for UE 90ac
```

Figure 7.3: Mobile network reconnection, as seen at the eNB machine

7.3 Link to Drone UE Github

`https://github.com/landrs-toolkit/Drone-NR` : GitHub

7.4 Link to Linux Open Source Summit Presentation

`https://www.youtube.com/watch?v=nm0Vuc32Q_I` : YouTube