Department of Communications and Networking

An Efficient System Orchestrator and a Novel Internet of Things Platform for Unmanned Aerial Systems

Naser Hossein Motlagh

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 18/2018

An Efficient System Orchestrator and a Novel Internet of Things Platform for Unmanned Aerial Systems

Naser Hossein Motlagh

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall AS2 of the school on 23 March 2018 at 12.

Aalto University
School of Electrical Engineering
Department of Communications and Networking

Supervising professor
Prof. Tarik Taleb, Aalto University, Finland

Thesis advisor
Dr. Miloud Bagaa, Aalto University, Finland

Preliminary examiners
Prof. Zhenhui Yuan, Hangzhou Dianzi University, China
Dr. Jaeho Kim, Korea Electronics Technology Institute, Korea

Opponent
Prof. Mika Ylianttila, University of Oulu, Finland

Aalto University publication series
DOCTORAL DISSERTATIONS 18/2018

ISBN 978-952-60-7821-2 (printed)
ISBN 978-952-60-7886-1 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-7886-1

Unigrafia Oy
Helsinki 2018

Finland

NORDIC ECOLABEL
441 697
Printed matter

A? Aalto University

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Abstract

Author
Naser Hossein Motlagh

Name of the doctoral dissertation
An Efficient System Orchestrator and a Novel Internet of Things Platform for Unmanned Aerial Systems

Publisher School of Electrical Engineering

Unit Department of Communications and Networking

Series Aalto University publication series DOCTORAL DISSERTATIONS 18/2018

Field of research Networking Technology

Manuscript submitted 6 November 2017 **Date of the defence** 23 March 2018

Permission to publish granted (date) 6 February 2018 **Language** English

☐ **Monograph** ☒ **Article dissertation** ☐ **Essay dissertation**

Abstract
Unmanned aerial vehicles (UAVs) are used to provide diverse civilian, commercial, and governmental services. In addition to their original tasks, UAVs can also be used to offer numerous value-added internet of things services (VAIoTSs). Sharing UAV infrastructure to provide IoT services can lower both capital and operational expenses to create a novel ecosystem with new stakeholders. The deployment of UAV-based IoT platforms, along with mechanisms to provide VAIoTSs from the sky, comes with a number of challenges, and solving few of them is the objective of this dissertation. To enable VAIoTS delivery, this dissertation introduces an innovative UAV-based IoT platform and envisions a UAV-based communication architecture including a system orchestrator (SO). The SO manages the UAVs and their on board IoT devices and optimally carries out diverse IoT services in an energy-efficient manner, using reliable data communication. Furthermore, to ensure efficient delivery of VAIoTSs using UAVs, a UAV selection mechanism is proposed, which takes into account multiple metrics in the UAV selection process such as UAVs' onboard IoT devices, energy, geographical proximities, and the priority levels of the IoT events. In this light, the following three complementary solutions are proposed: energy-aware UAV selection (EAUS), which aims to reduce UAVs' energy consumption; delay-aware UAV selection (DAUS), which aims to reduce UAVs' operational time; and fair trade-off UAV selection (FTUS), which ensures a fair trade-off between UAV energy consumption and operation time. The results demonstrate the efficiency and robustness of the proposed schemes. In addition, to provide reliable communication for the UAVs, a connection steering mechanism between mobile networks and UAVs is presented. This mechanism selects the network with the strongest radio signal among available networks in order to ensure network coverage and increase transmission rates. The efficiency of this mechanism is examined using two different LTE-4G networks of two different network providers. The results prove the efficiency of the proposed mechanism in terms of an increase in data transmission rate and a reduction in UAV energy consumption. Furthermore, as an envisioned application of the UAV-based IoT platform, a crowd surveillance use case based on face recognition is proposed. To evaluate the use case, the offloading of video data processing to a mobile edge computing (MEC) node is compared to the local processing of video data on board a UAV. The results prove the efficiency of the MEC-based offloading approach in saving the energy of UAVs. In Conclusion, the research presented in this dissertation, through the presented UAV-based IoT platform, the system orchestrator, and the designed mechanisms enables the effective delivery of VAIoTSs from the sky.

Keywords Unmanned Aerial Vehicle, UAV, Unmanned Aerial System, UAS, Internet of Things, IoT, UAV Selection Mechanism, UAV Networks.

ISBN (printed) 978-952-60-7821-2 **ISBN (pdf)** 978-952-60-7886-1

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2018

Pages 147 **urn** http://urn.fi/URN:ISBN:978-952-60-7886-1

Preface

The research for this doctoral dissertation has been carried out in the Department of Communications and Networking (ComNet), School of Electrical Engineering, Aalto University, Finland.

I express my deepest gratitude to Prof. Tarik Taleb, who has given me the opportunity to pursue the doctoral degree under his supervision with his endless guidance and support. I would like to thank my advisor, Dr. Miloud Bagaa, for all his instructions and encouragement. In addition, I wish to thank Prof. Song Jaeseung for offering me a research visit at the Department of Computer and Information Security, Sejong University, Korea. Special thanks go to my pre-examiners, Prof. Zhenhui Yuan and Dr. Jaeho Kim, for their valuable remarks and constructive comments. Thank you to my colleagues at MOSA!C Lab, who created a fruitful science and technology environment for me to work in. I would also like to acknowledge the assistance I received from the ComNet laboratory, and I would like to thank laboratory engineer Mr. Viktor Nässi for his readiness in providing any required laboratory equipment in a short time. Thanks also go to the Aalto IT Service Desk members for their continuous service and support. I am grateful to my dear friends in Finland, both in the Helsinki region and in Vaasa, who have supported me with their wisdom and kindness during my studies. Most of all, I would like to thank my parents and my family, who have always given me strength and freedom to follow my dreams.

Lastly, I am deeply grateful to my wife, Somayeh, for her tremendous support, endless patience, and understanding during my doctoral studies.

Helsinki, February 28, 2018

Naser Hossein Motlagh

Contents

List of Publications

This dissertation consists of an overview of the following publications, which are referred to in the text by their roman numerals.

I N. H. Motlagh, T. Taleb, and O. Arouk, "Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899-922, Dec 2016.

II N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV Selection for a UAV-based Integrative IoT Platform," In *2016 IEEE Global Communications Conference (GLOBECOM)*, Washington, USA, pp. 1-6, Dec 2016.

III N. H. Motlagh, M. Bagaa, and T. Taleb, "UAVs Task Assignment Mechanism for UAV-based IoT Platform," *submitted to IEEE Transactions on Vehicular Technology (TVT)*, 2018.

IV N. H. Motlagh, M. Bagaa, T. Taleb, and J. Song, "Connection Steering Mechanism between Mobile Networks for reliable UAVs IoT Platform," In *2017 IEEE International Conference on Communications (ICC)*, Paris, France, pp. 1-6, May 2017.

V N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-Based IoT Platform: A Crowd Surveillance Use Case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128-134, Feb 2017.

Author's Contributions

Publication I: "Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives"

The author prepared the content of the manuscript, provided the literature review, and wrote the paper. Dr. Arouk assisted in writing the manuscript. Prof. Taleb supervised the work during the research process and contributed to the manuscript by making revisions and enhancing the quality of the paper.

Publication II: "UAV Selection for a UAV-based Integrative IoT Platform"

The author had the main responsibility of writing the paper and developing the idea. The author made the problem formulation and, together with Dr. Bagaa, defined the mathematical optimizations and made the simulations. The author analyzed and interpreted the results. The final manuscript took advantage of the valuable comments given by Prof. Taleb.

Publication III: "UAVs Task Assignment Mechanism for UAV-based IoT Platform"

The author developed the idea and defined the research questions. The author established the system modeling, performed the data analysis, interpreted the results, and drafted the manuscript. Dr. Bagaa assisted in defining the optimization section and the simulation of the proposed solutions. Prof. Taleb supervised the work. He also contributed to enhancing the quality of the manuscript by providing valuable suggestions and making revisions.

Publication IV: "Connection Steering Mechanism between Mobile Networks for reliable UAVs IoT Platform"

The author had the main responsibility of developing the test-bed. He made the laboratory experiments and collected the data. Dr. Bagaa helped the author develop the steering mechanism. The author collected and analyzed the data and wrote the paper, while Dr. Bagaa advised the writing process. Prof. JaeSeung Song revised the paper. Prof. Taleb supervised the research and provided valuable comments on the paper.

Publication V: "UAV-Based IoT Platform: A Crowd Surveillance Use Case"

This article is the result of collaboration between all authors. The idea for the article was proposed by Prof. Taleb. The test-bed developments, measurements, and simulations were conducted by the author with the assistance of Dr. Bagaa. The author wrote the manuscript, while Dr. Bagaa contributed to the writing process. Prof. Taleb supervised the work and contributed to enhancing the quality of the manuscript by giving valuable suggestions and making revisions.

List of Abbreviations

ARQ	Automatic Repeat Request
BS	Base Station
DAUS	Delay Aware UAV Selection
DTMC	Discrete Time Markov Chain
EAUS	Energy Aware UAV Selection
GCS	Ground Control Station
FANET	Flying Ad-hoc Network
FC	Flight Controller
FTUS	Fair Trade-off UAV Selection
IoT	Internet of Things
LoS	Line of Sight
LBPH	Local Binary Pattern Histogram
LIP	Linear Integer Problem
MEC	Mobile Edge Computing
NASA	National Aeronautics and Space Administration
NBM	Nash Bargaining Model
OpenCV	Open Source Computer Vision
OP	Optimization Problem
PC	Personal Computer
QoS	Quality of Service
RAT	Radio Access Technology
RF	Radio Frequency
RPi	Raspberry Pi
RSSI	Received Signal Strength Indicator
TCP	Transport Control Protocol
UAV	Unmanned Aerial Vehicle
UAS	Unmanned Aerial System
UTM	Unmanned Traffic Management
VAIoTS	Value-Added IoT Service

List of Symbols

a	The Air
b	UAV's body
C	Capacitance
f	Switching frequency
I_{leak}	Leakage current
K	A unitless constant
L	Priority level
M	Maximum number of retransmissions
$n_{\mathcal{B}}$	Gaussian distributed random variable
P_{CPU}	CPUs power consumption
P_{charge}	Required power to charge the capacitors
P_{leak}	Leakage power
P_{short}	Short-circuit power
P_r	Received power at a UAV
P_t	Transmitted power at a UAV
P_u	Power consumption per retransmission
R_b^w	UAV's turning radius (rotation)
T_F	Required time for a single transmission
$T_u^{Transmit}$	Sojourn time of a packet before its transmission
u	A UAV
V	Voltage swing across capacitor
V_a	UAV's air speed
V_{dd}	Supply voltage of circuit
V_u^a	UAV's air relative velocity
w	The wind

α	Side-slip angle
β	Angle of attack
Γ_u	Maximum allowed altitude for a UAV
γ_{th}	Target minimum received power level
Δ_t	Time duration that the device is active
η	Scaling factor (effects of short-circuit power)
λ	Energy consumption per meter
μ	Amount of systems active mode (a constant)
$\xi_u^{Battery}$	UAV's full battery amount
$\xi_u^{SenseProcess}$	Energy consumption for sensing-and-processing
ξ_u^{Travel}	Energy consumption for traveling
$\xi_u^{Transmit}$	Energy consumption for packet transmissions
ξ_{th}	Energy threshold
$\sigma_{\psi_{dB}}^2$	Variance
Υ_{th}	Time threshold
Υ_u^{Travel}	Traveling time between two positions
$\Upsilon_u^{Transmit}$	Average transmission time
$\Upsilon_u^{Endurance}$	Maximum flight time that a UAV u can fly
Φ	Payoffs vector
φ	Path loss exponent
ω	Pair of utility functions
$\mathcal{D}_0$	Reference distance (typically one meter)
$\mathcal{D}_{u,\mathcal{E}}$	Distance from the UAV to the event $\mathcal{E}$
$\mathcal{E}$	An event
$\mathcal{E}_L$	Incoming events
$\mathcal{E}_\Gamma$	Scheduled events
$\mathbb{E}(N_{u,\mathcal{B}})$	Average number of retransmissions
$\mathcal{R}$	Rule
$\mathcal{K}$	Number of data packets
$\mathcal{N}$	Set of UAVs
$\ddot{\mathcal{N}}$	Set of eligible UAVs
$\mathcal{P}_{u,\mathcal{B}}$	Outage probability
$\bar{P}$	Average consumed power
$\mathcal{Q}_{th}$	Threshold level for the accomplished task
$\mathcal{Q}_{e_\Gamma}$	Amount of accomplished task from an event
$\mathcal{S}_\mathcal{E}$	Set of IoT devices required by an event
$\mathcal{S}_u$	Set of IoT devices on board a UAV
$\mathcal{U}$	Selected subset of eligible UAVs
$\mathcal{X}$	Set of players strategies

List of Figures

1. Introduction

1.1 Literature and motivation

Unmanned aerial vehicles (UAVs), known as drones, are deployed in various sectors of human life to provide diverse civilian, commercial, and governmental services [1]. Many UAV use cases include, but are not limited to, public safety, homeland security, environmental monitoring, farming, architecture, surveillance, Internet delivery, and goods transportation such as by Amazon Prime Air [2]. For instance, a crowd surveillance use case [3] aims to provide public safety through surveillance of people. In this case, UAVs are used to identify criminals by detecting any suspicious human activities. In fact, UAV use cases are growing fast, yet the market for UAVs is in its infancy. From a technology perspective, UAVs are foreseen as an important component of an advanced cyber-physical Internet of Things (IoT) ecosystem [4]. Based on its definition, IoT aims to allow things to be connected anytime, anywhere, with anything, ideally using any network and providing any service. Using the concept of the IoT for UAVs allows them to become an integral part of the IoT infrastructure. Therefore, UAVs possess unique characteristics in being dynamic, easy to deploy, easy to reprogram during run time, and capable of measuring anything anywhere [5]. In addition, smart UAV management systems are capable of planning and controlling UAV flights with a high degree of autonomy [6].

In fact, UAVs consist of different parts such as avionics. They can also be equipped with diverse types of sensors, cameras, software tools, and communication technologies to provide IoT services from a given height. Thus, the UAV body with on board hardware devices can be considered the physical entity, and the UAV controller system with the employed software tools can be considered the virtual entity. These two parts jointly form a smart object (UAV or thing) [7], or alternatively, a UAVIoT platform. Therefore, using this platform, in addition to doing their designed tasks, e.g. mail delivery, UAVs can be simultaneously exploited for other value-added services (VASs) [8], particularly in the IoT domain. Consequently, this service type can be called a value-added IoT service (VAIoTS). A UAV that is designed to perform a specific task can be equipped with diverse IoT devices such as sensors and cameras. These devices can be triggered on and off whenever needed. For example, suppose a transport safety agency is interested in knowing the current traffic status on a specific street. A UAV flying near that area, equipped with a suitable camera on board, can provide live video streams from that particular street. In this regard, without investing in UAVs, various stakeholders can use them for diverse IoT services. Therefore, these VAIoTSs would potentially generate additional sources of revenue for the actual owners of the UAVs. Moreover, sharing the infrastructure of this potential unmanned aerial system (UAS), which consists of multiple UAVs, to provide diverse IoT services would lower both capital and operational expenses and create a novel ecosystem with new stakeholders.

Offering VAIoTSs on a heterogeneous platform of UAVs originally destined for other tasks is a challenging issue. Let us assume widespread flying UAVs, where each one is equipped with a different IoT device (e.g. a sensor or camera) with a specific functionality, and each obtains a limited amount of energy to perform its task. Selecting a UAV or a suitable set of UAVs to perform an IoT task first requires obtaining complete information about their on board IoT devices, their residual battery amount, and their geographic locations. Second, there need to be employed efficient methods for selecting the right set of UAVs to have an IoT task assigned to them. This UAV selection requires designing a central computing and orchestration system that holds updated information about the UAVs. Collecting data from the remote locations using these heterogeneous IoT devices is another challenge. Since, the UAVs need to fly over different environments (e.g. sea, mountains, or forest), they need to access the locations properly. One important challenge is that the UAVs need to connect to the network (a ground control station or a base station) using reliable access technology. This is a vital factor for commanding and controlling the UAVs for safe and successful flight and operation. Another challenge relates to the processing of the collected data, because the battery resources of the UAVs are limited. Actually, in heavy computations, e.g. video data processing, there is need for a considerable amount of energy resources. Thus, the UAVs need to offload their heavy processing tasks to the ground stations or edge computing nodes.

Moreover, keeping the IoT devices constantly operating and connecting to the network drains their batteries, and the UAVs are not able to complete the IoT tasks. For instance, a UAV may be programmed to deliver two objects to two distant places. After the delivery of the first object, the battery level becomes low and the UAV may not be able to deliver the second object and return safely. In this case, on board sensors and cameras should be turned on and off as needed. Considering the above stated issues, many criteria are of importance when selecting a UAV, or a set of UAVs, to carry out a particular IoT task. Therefore, efficient methods and algorithms are required *i*) to select a UAV or the right set of UAVs to perform a specific IoT service, *ii*) to keep steady and reliable communication links with the ground control stations (GCSs)/base stations (BSs) to deliver the data, and *iii*) to collect the data and decide whether to process them locally on board the UAV or to offload the computations to a mobile edge node. In order to deliver VAIoTSs, since the UAVs are highly dynamic and highly mobile objects, there is a need to ensure reliable data links (i.e. good coverage and stable connectivity) in UAV communications. Therefore, the type of communication technology to be used on board a UAV becomes a significant issue.

In addition, advanced mobile communication systems, such as LTE-4G and 5G networks, are the communication technologies that support the dynamic and mobility features of UAVs. These mobile networks support UAV flights in long distance, in high altitude, and beyond the visual line of sight (BVLoS) [9, 10]. These networks enable a UAV to transfer data between a GCS/BS and other exiting UAVs in the network with very high quality of service (QoS). Actually, current LTE-4G systems are used to increase network expandability to up to hundreds of thousands of connections for low-cost, long-range, and low-power IoT devices. In addition, 5G networks are planned to offer high data speed exceeding 10 Gbps and extremely low latency, i.e. 1 ms [11]. In this regard, many studies [3, 12–19] have examined the performance of the LTE-4G networks in terms of different QoS metrics, such as throughput, loss rate, delay, multipath propagation, shadowing, and fading models. These studies confirm that the LTE-4G networks are promising candidates for supporting the high mobility of UAVs. To offer VAIoTSs, depending on the energy required for the data computation, the collected IoT data can be processed locally on board a UAV or it can be delivered to a mobile edge computer (MEC) node [20]. In fact, MEC can solve the computing restrictions of UAVs by enabling them to offload intensive computations to an MEC node. The outstanding characteristics of MECs are their service mobility support, their closeness to end users, and their dense geographical deployment [21]. In addition, obtaining such a UAV-IoT platform assists the unmanned traffic management (UTM) system, envisioned by the USA National Aeronautics and Space Administration (NASA) [22] in managing UAV networks.

1.2 Research environment

The research performed for this dissertation was carried out at the Department of Communications and Networking (COMNET) [23], School of Electrical Engineering, Aalto University. The department is equipped with a modern research environment, laboratory equipment, and advanced telecommunication facilities as shown in Fig. 1.1. The underlying LTE network (eNodeB) is exclusively used for research, and it offers low latency and a high bit rate, as well as extended coverage, to support a variety of scenarios, where measurements can be carried out horizontally, vertically, at higher altitudes, within line of sight (LoS), and beyond LoS.

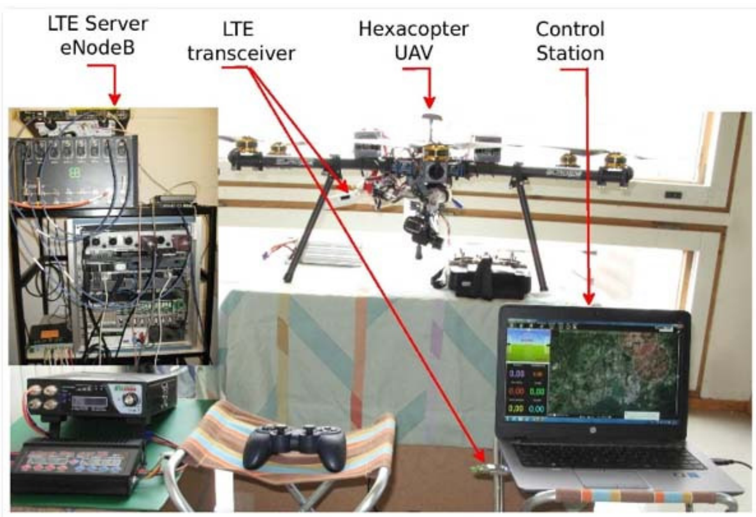


Figure 1.1. Aalto's UAV setup.

The UAV employed in the research is a hexa-copter equipped with an LTE modem, a gimbal with a high-resolution digital camera, and a processing unit. The hexa-copter includes a flight controller (FC) module for a stable flight, accelerometers, a barometer, and an embedded Linux gateway (i.e. a Raspberry Pi) interconnecting the LTE modem to the FC. To set up an LTE connection, any personal computer (PC) can be used as a GCS. Flight control software such as Mission Planner is installed on the PC. The PC is used for controlling the FC via a connected LTE modem. The hexa-copter can carry 1.5 kg of payload, including laboratory equipment and metering devices. With a completely charged battery, its flight time is around 30 minutes with the full payload. It also has a safe landing scheme to cope with unlikely motor failure situations.

1.3 Scope and structure

Deploying UAV-based IoT platforms, along with obtaining mechanisms to provide VAIoTSs from the sky, comes with a number of challenges. In this regard, this dissertation aims to introduce a comprehensive UAV-based IoT platform to enable IoT services and a system orchestrator to manage the UAVs and the delivery of their IoT services. Here, the system orchestration is planned to optimally carry out diverse IoT services in an energy-efficient manner, using reliable data communication technologies. Therefore, the dissertation attempts to find acceptable answers to the following research questions:

1. What is a comprehensive system orchestrator–based architecture model for the delivery of UAV-based IoT services?
2. What mechanism is used to select appropriate UAVs to perform IoT tasks in an energy- and time-efficient manner?
3. What mechanism is used to provide a reliable communication link to the GCS or BS?
4. What is the benefit of MEC-based computation offloading toward local processing on board UAVs in IoT services?

This dissertation uses the concepts, methods, and experiments presented in the associated publications and analyzes the results achieved from those methods and experiments. Table 1.1 shows the relationship between the research questions, publications, and the dissertation chapters.

The envisioned architecture for UAV communications is discussed in different sections of Publications I, II, and V and covered in Chapter 2 of this dissertation. Publications II and III define, model, simulate, and analyze the UAV selection mechanism, as discussed in Chapter 3. The reliable communication of UAVs is explored in Publication IV through a test-bed experiment and in Chapter 4 here. Finally, UAV computation offloading vs. local data processing is evaluated through a small-scale test-bed that is discussed in Publication V and in Chapter 5.

Table 1.1. Relationship between research questions, publications, and chapters in this dissertation.

Question	Main Theme	Publication	Chapter
Q1	UAVs' envisioned architecture	I, II, V	2
Q2	UAV selection mechanism	II, III	3
Q3	UAVs' reliable communications	IV	4
Q4	UAVs' MEC-based computation offloading	V	5

1.4 Contribution of the dissertation

The research presented in this dissertation is centered on advancing the current state of knowledge about UAVs. The results and findings of this dissertation serve the existing literature by offering the following specific advancements in this field. Fig. 1.2 shows the linkage between the core innovation and the contributions of this dissertation.

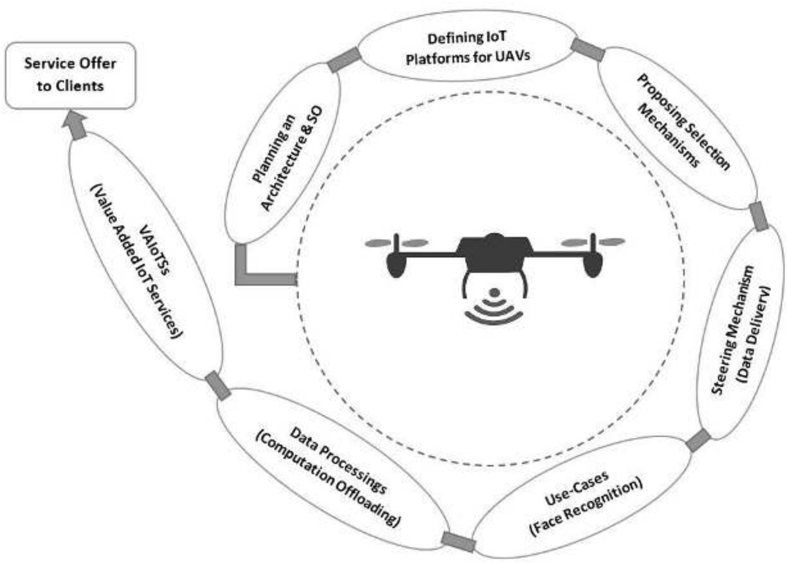


Figure 1.2. The overview of the contributions of this dissertation.

1. **An innovative UAV-based IoT platform is proposed.** This platform enables collecting data via remotely controllable IoT devices mounted on UAVs whenever needed.

2. **A communication architecture including an orchestration system is envisioned for a UAV network.** To provide VAIoTSs using a UAV IoT platform, there needs to be $i)$ a system orchestrator that is aware of diverse contextual information about UAVs, such as their flying routes, on board IoT equipment, and battery status and $ii)$ a communication architecture model that allows any UAV with any radio access technology to connect to the UAV's network in order to handle an IoT task.

3. **Value-added IoT services are defined and planned.** In addition to UAVs carrying out their original tasks (e.g. parcel delivery or environmental monitoring), they can be simultaneously used to offer numerous value-added services, in the area of the IoT. In fact, sharing the infrastructure of UAVs to provision diverse VAIoTSs would lower both capital and operational expenses and create a novel ecosystem with new stakeholders.

4. **An efficient UAV selection mechanism is proposed to optimally carry out diverse IoT services.** To get benefits from the VAIoTSs provided by the UAVs, there is a need for an optimal UAV selection mechanism that selects the most appropriate UAVs to perform diverse IoT tasks. For the UAV selection, the following three complementary solutions are considered: $i)$ energy-aware UAV selection (EAUS), which aims to reduce the energy consumption of UAVs, $ii)$ delay-aware UAV selection (DAUS), which aims to reduce the operational time of UAVs, and $iii)$ fair trade-off UAV selection (FTUS), which ensures a fair trade-off between a UAV's energy consumption and operation time.

5. **A connection-steering mechanism is proposed for reliable UAV communications.** This mechanism selects the network with the strongest radio signal among available networks to ensure network coverage and increase the transmission rates. The efficiency of this mechanism is examined using two different 4G networks of two different network providers. The results of the experiment prove the efficiency of the proposed mechanism in terms of an increase in data transmission rate and a reduction of UAVs' energy consumption.

6. **As an envisioned application of the UAV-based IoT platform, a crowd surveillance use case based on face recognition is proposed.** Using UAVs with appropriate IoT devices such as video cameras can offer an efficient crowd surveillance system by quickly detecting and recognizing suspicious persons in crowds of people. Thus, using a face recognition use case, crowd safety and security can be enhanced.

7. **The offloading of video data processing to an MEC node is proposed as opposed to the local processing of the video data on board a UAV.** The computation overhead required for some UAV use cases and, considering the limited power supply of UAVs, the processing of collected data by a UAV are challenging issues. Thus, through an experiment, the computation offloading approach to an MEC node is suggested as opposed to local processing for video data computations, which require significant energy resources.

2. UAV-based IoT communications

One of the significant issues with the deployment of value-added services is the way IoT devices are activated, sense the data, and deliver the data in an efficient way. The assumption is that the UAVs are equipped with 4G or higher technology to connect to deployed cellular networks. Further, they are also equipped with short-range communication like Wi-Fi so they can connect among themselves directly. In addition, UAVs need to communicate with each other in a continuous manner, employing a powerful networking method. Thus, to establish a way to provide forceful IoT data delivery and communication for UAVs, this chapter discusses i) the technologies applied in UAV communications, ii) UAV communication networks, and iii) UAV data processing.

2.1 Communication technologies for UAVs

To establish communication among UAVs and also with the network, an appropriate access network is needed. These access networks can be divided into two categories: short- and wide-range communications. Wide-range communications provide coverage over a large area such as cellular (e.g. LTE), broadband (e.g. Wimax), and satellite communications. The other possibility is short-range communication, which is used for short distances, e.g. Wi-Fi. In fact, advances in telecommunication technologies enable control of UAVs flying at high altitudes from considerable distances. LTE-4G systems and the upcoming 5G system have the ability to provide reliable mobile connectivity to UAVs for aerial data collection, processing, and analysis [11]. Communication among these IoT devices on board UAVs requires flexibility, speed, and reliability. These advanced communication technologies support reliability, connectivity, and high mobility of UAVs flying in the sky. For example, UAVs will gain benefits using 5G for real data and high-resolution video streaming such as 4k capacity. Moreover, UAVs can be equipped with communication relays and can operate at high altitudes, where these relay platforms can deliver mobile, persistent connectivity over different regions [24]. In a use case, Google has tested multiple prototypes of solar-powered Internet UAVs to ensure security for delivering Internet connectivity from the sky. On top of this, Google's project SkyBender uses UAVs to deliver next-generation 5G wireless Internet, which is up to 40 times faster than 4G systems [25].

2.2 UAV communication networks

An airborne network (AN) is a cyber-physical system (CPS) in which there is an intense interaction between physical and cyber components [26]. The synergy between the cyber and physical components significantly enhances the safety and security capabilities of next-generation air vehicle systems. In fact, there are four types of communication architectures that can be used for small UAS applications: direct link, satellite, cellular, and mesh networking [9]. Regarding the communication systems and networking among UAVs, the collaborative communication in UAV systems is an important part, as they can be equipped with different communication technologies [27]. This collaborative communication can be between UAVs themselves, or between UAVs and other nodes such as GCSs, wireless sensor networks, and other ground-moving vehicles. UAV systems apply different networking methods, some of which are presented below.

Node-to-node communication: A direct link or LoS communication between UAV(s) to GCS is the simplest architecture [9]. This type of communication occurs as UAV-to-UAV (U2U) and UAV-to-infrastructure (U2I). When a node-to-node communication is applied in a UAV network with multiple UAVs, one of the UAVs plays the role of a gateway, where it collects the data from other UAVs (through U2U communication) and then relays the collected data to the GCS.

Mesh networking: Mesh networking can be defined as an architecture where every node, i.e. a UAV or a GS, can act as a data relay. In addition, communication among multiple UAVs and GS can occur over several hops through intermediate nodes [9]. The advantages of mesh networking include the facts that *i*) the shorter communication range simplifies the link requirement, *ii*) U2U communication becomes direct, *iii*) any node connected to a cellular network enables communication with the other nodes, and *iv*) the communication range is extended.

Delay-tolerant network: Delay-tolerant network (DTN) architecture aims to provide interoperable communications between a wide range of networks that may have poor and disparate characteristics [28]. DTN architecture has been designed to address the needs of networks characterized by link-intermittent connectivity, lack of end-to-end connectivity between end users, and high latency [29]. In order to mitigate these problems, DTNs rely on store-and-forward message switching.

Mobile ad-hoc network: A mobile ad-hoc network (MANET) is a collection of independent mobile nodes connected with wireless links. Using these mobile nodes as hosts and relays, MANET configures an infrastructure-less network dynamically. These nodes have free movement and their network topology changes rapidly over time. To address the challenges of the UAV networks, the study in [30] compares the characteristics of MANETs with those of UAV networks. This study concludes that, for UAV networks, new protocols are required for adapting high mobility, fluid topology, intermittent links, power constraints, and link-quality changes.

Flying ad-hoc network: Flying ad-hoc networks (FANETs) basically represent a new class of ad-hoc networks composed of aerial vehicles [31–33]. It is a new networking paradigm based on the concept of MANETs, but the nodes in a FANET have a greater degree of mobility, and the distance between nodes is often higher than in a MANET [34]. A FANET resolves several design limitations with the infrastructure-based architecture approach. It solves the range restriction among UAVs and the GS and improves the reliability of the communication [35]. It also enables providing real-time communication without requiring any infrastructure. A FANET allows UAVs to communicate in an ad-hoc manner, while some UAVs communicate with GCS or via satellite.

2.3 UAV's Data processing

Data processing done by UAVs depends on the type of application and whether it is real-time or not. For non-real-time applications where there is no constraint on the delay, i.e. delay-tolerant applications, the collected data can be stored and processed in a remote cloud (i.e. cloud computing), and then a user can request the post-processed information from the cloud. Regarding real-time applications, the relevant data would be stored and processed in a local cloud that is in the vicinity of the monitored location. For crucial cases where a delay is critical, specific UAVs acting as a moving “cloudlet” can be used. Cloudlets are techniques that propose the augmentation of mobile computational resources with adjacent servers [11]. They increase processing capacity, conserve energy resources, and ease deployment.

In fact, this computation offloading (CO) solves the computing and storage resource restrictions of UAVs because intensive computation is not performed on board a UAV but offloaded to the cloud. In addition, CO reduces the power consumption of a UAV, since most of the tasks, especially computation-intense ones, are then performed in the cloud and not in the UAV [36]. The study in [37] titled “Communicating while computing” reviews a series of computational offloading mechanisms. This study affirms that mobile cloud computing (MCC) has three main advantages. First, it enhances battery lifetime by offloading energy-consuming tasks from a mobile device to the cloud. Second, it enables mobile devices to run complicated applications and provides higher data storage capabilities. Third, it improves reliability, as the data can be stored and backed up from a mobile device to a set of reliable fixed storage devices.

Considering the presented communication technologies and networks suitable for UAVs, the next two sub-chapters introduce the envisioned communication architecture for UAVs and an IoT platform applicable on board UAVs.

2.4 Vision of UAV-based architecture

In order to answer the first research question proposed in 1.3, this sub-chapter introduces an architecture envisioned for a UAS. The sub-chapter further discusses the related challenges and the requirements for developing such a UAV-based IoT communication architecture.

2.4.1 UAV-based IoT platform

The basic task of a UAV is to collect data from remote locations. This data collection requires having IoT devices, e.g. sensors and cameras, on board the UAV. In addition, a reliable data transmission system is needed to share the collected data with the other UAVs in the network or with the GCSs. These IoT devices are available in various types and they include the aforementioned communication technologies. Thus, to use these IoT

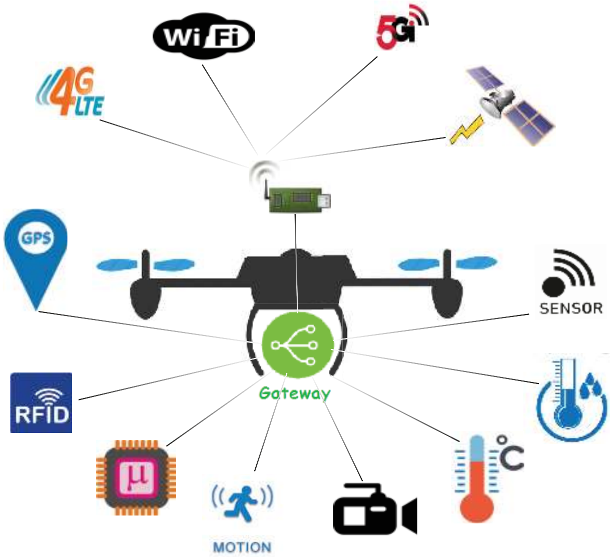


Figure 2.1. *U*AVs equipped with various IoT devices.

devices to collect and deliver IoT data from a given height, an integrative IoT platform needs to be mounted on board the UAV [38]. Hence, this IoT platform can be developed on an IoT gateway, e.g. a Raspberry Pi, that allows connecting, installing, and activating the diverse IoT devices, as shown in Fig.2.1.

Using the platform, IoT data can be collected remotely from the sky whenever IoT devices are activated in the intended positions. In addition, depending on the required energy, the collected data can be processed locally on board UAVs or it can be offloaded to an edge server on the ground. In fact, mounting such an IoT platform on board a UAV enables it to provide VAIoTSs from the sky. However, when UAVs are deployed massively, organizing them to provide VAIoTSs becomes a complex issue. In fact, each UAV may have a different type of IoT device, and each UAV may be equipped with a different kind of access technology (e.g. a cellular system or Wi-Fi). Therefore, there is a need for a central system orchestrator that organizes UAVs and their on board facilities. Furthermore, an efficient communication architecture that enables UAVs to connect to the system orchestrator or to other UAVs is necessary. The rest of this chapter is devoted to introducing an envisioned UAV communication architecture including the central system orchestrator.

2.4.2 Envisioned architecture for UAV communications

Fig. 2.2 illustrates the architecture envisioned in this dissertation. The figure presents a widespread network of flying UAVs, with some already in flight and some ready to fly when commanded. Each of these UAVs differs in size, capability (e.g. flight endurance), and equipment on board, and each is expected to perform a particular task. For example, one UAV is used for parcel delivery and another for environmental monitoring. By having the IoT platform on board the UAVs, as proposed in the previous sub-chapter, these UAVs are able to deliver VAIoTSs from a given height whenever their IoT devices are remotely actuated and controlled at the right time, in the right place, and in the right direction (e.g. for cameras). However, UAV flights are piloted based on predetermined waypoints to avoid any physical collisions with obstacles such as tall trees and towers. In fact, the UAVs need to be self-organized to avoid such as collisions.

In the envisioned architecture, UAVs can construct UAV clusters of various sizes. Furthermore, each UAV has a particular task to perform, and in each cluster UAVs communicate in a FANET while using different communication technologies, e.g. a cellular system, Wi-Fi, or satellite. In the envisioned architecture, due to the geographical proximity of the UAVs, their flying altitude, and their intended application and to overcome the communication technology limitations, the UAVs can construct clusters of various sizes. In fact, clustering enhances the capabilities of the UAVs by leveraging the range of their wireless communications, enabling them to share their IoT devices on board, thus increasing their computation resources and their data transmission capabilities. In addition, in each UAV cluster, a specific UAV can be elected to be a cluster head (CH) that works as a router (using a suitable routing protocol) and delivers the data collected from the other UAVs to the GCS (the system orchestrator [SO] in the envisioned architecture) through the core network that sustains connectivity to the different wireless technologies (as shown in Fig.2.2).

In the envisioned architecture, wireless communication happens as UAV-to-SO, U2U, UAV-to-cellular-infrastructure, and UAV-to-satellite communication in fly ad-hoc manner. The selection of communication technology is another concern in UAV communications. Actually, in some regions, there may exist connectivity and coverage limitations of cellular networks or satellite communications (SATCOM) to UAV systems. For instance, if a UAV is required to perform operations in a region where there is no base

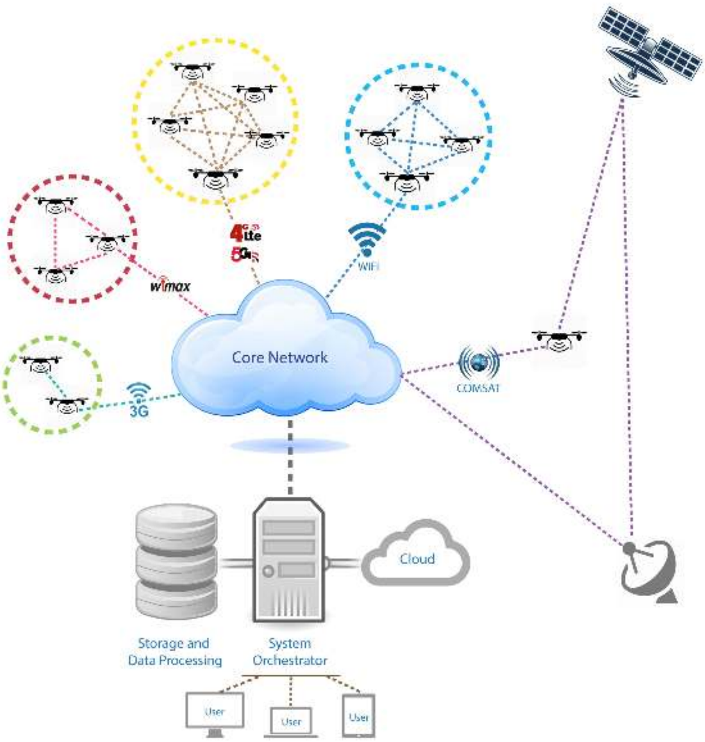


Figure 2.2. *T*he envisioned architecture for a UAV-based integrative IoT platform.

station (BS) to support cellular systems, then the UAV should be able to communicate via Wi-Fi or SATCOM in consonance with the region and the facilities on board. However, the limitations regarding UAV communications can be alleviated by using a FANET. Likewise, applying a DTN approach to these FANETs can ensure end-to-end connectivity, but at the price of increased delay. As shown in Fig.2.2, on one hand, the duty of the core network is to interconnect the UAVs, while the SO enables data exchange among diverse components of the network in a secure manner. On the other hand, the SO works as the brain of the whole system by employing a set of mechanisms and algorithms for collecting real-time data about the current status of the UAVs, their routes, their current level of energy, and their equipment, e.g. any avionics and IoT devices with their functionalities. In practice, the duty of the SO is to coordinate the UAV operations and their IoT devices. Fig. 2.3 presents the SO functionality and the parameters it takes into account when receiving a request from a user until the selection of UAVs to perform a task.

One of the main duties of the SO is to handle the requests for IoT services (events) from the clients (users) of the architecture. In fact, if a

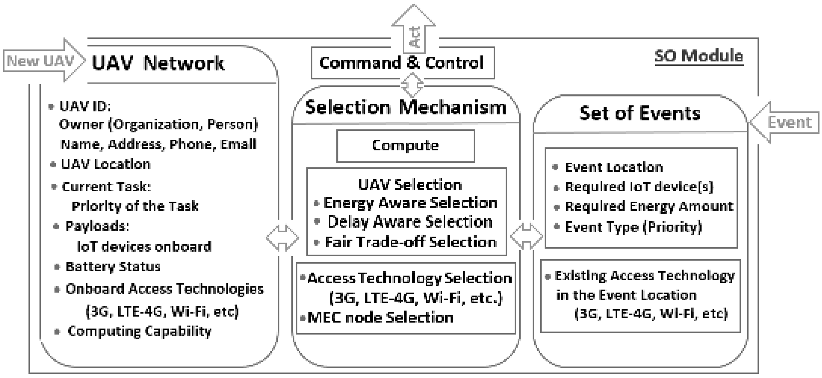


Figure 2.3. *T*he SO module and its functionality.

request from a user is received, SO sorts the most appropriate UAVs to handle that request. For instance, an environment-monitoring agency may request the level of air pollution in a particular region. The UAV sorting for this task may depend on different criteria such as 1) flying paths of the UAVs (i.e. current or planned ones), 2) the UAVs' geographical proximity to the region of interest, 3) their on board IoT devices (e.g. an air pollution meter), 4) their battery level (i.e. ensuring that if a UAV is commanded to perform an IoT service, its original task is not hampered by depleting its battery), and 5) the priority level of the UAVs' current mission (operation).

To handle an IoT event, the SO selects the UAVs, considering multiple parameters such as the size of the target area, the computation intensity of the IoT task, and the diversity of required IoT devices. When the UAVs have been selected, they are instructed to handle the IoT task by actuating their relevant IoT devices only when they are flying over the region of interest. In this fashion, the UAV energy budget is highly conserved. After the completion of the IoT task, the SO instructs the UAVs on how (e.g. in FANET fashion, DTN, UAV to cellular, etc.), where, and which access technology to use to deliver the sensed data. Additionally, the SO ensures interoperability among UAVs using different radio access technologies. For example, in case of a disaster recovery operation whereby heterogeneous UAVs are used, some UAVs may use cellular technology that operate in the coverage of a cellular system and other UAVs may use SATCOM to ensure reliable data exchange between the UAVs and the SO. Therefore, in this case, it is the duty of the SO to coordinate all of the UAVs to perform their tasks, i.e. disaster recovery.

The SO also coordinates the waypoints of UAVs to ensure collision-free flights. It also provides secured communications in the UAV networks by instructing the UAVs on which radio access technology to use and when and on where to deliver the collected data (to the SO, a mobile edge, or another UAV). The SO may connect to local storage or a remote cloud for the storage and processing of data received from diverse UAVs regarding various IoT service requests. In this way, the SO can regulate and support UAVs to have ubiquitous, convenient, and on-demand access to a shared pool of configurable computing and storage platform. Moreover, the SO is assumed to obtain all necessary intelligence to be self-* capable of autonomously self-operating, self-healing, self-configuring, and adequately resolving any possible conflicts from diverse policies.

3. UAV selection mechanism

3.1 Introduction

To benefit from the VAIoTSs offered by the UAVs, there is a need for an optimal UAV selection mechanism that selects the most appropriate UAVs to handle IoT tasks (events). This selection mechanism operates on the SO and is developed using efficient methods, functions, and constraints. In addition, to provide VAIoTSs, there needs to be complete information about the existing UAVs in the network. This information is about IoT devices on board the UAVs, the required equipment to handle an IoT event, the UAVs' residual energy amount, the amount of energy required to perform an IoT task, the distance to the location of an event, the time needed to travel between the current position of a UAV and the position of an event, the time required to complete an IoT task, and the priority level (urgency) of an event. In addition, it is necessary to assign priority levels for IoT events. In fact, some events may need urgent performance by UAVs, e.g. a photo shot at the scene of an accident. Moreover, the selection mechanism should perform in a way so that the UAVs consume energy as little as possible and shorten the operation time as much as possible.

Herein, the UAV selection mechanism, using the example shown in Fig. 3.1, is explained. Let u denote a UAV in a UAV network. The example shows that five UAVs are in flight and each one has different IoT devices on board, where the amount and types of these devices are shown with different shapes and colors. To handle an IoT event, the SO should select the UAVs that have the required devices on board. For instance, if a digital camera needs to be employed for taking photos, UAV1 and UAV3 are the candidates. In another example, the SO may receive a request for an IoT task that requires a UAV with a laser scanner, low-visibility

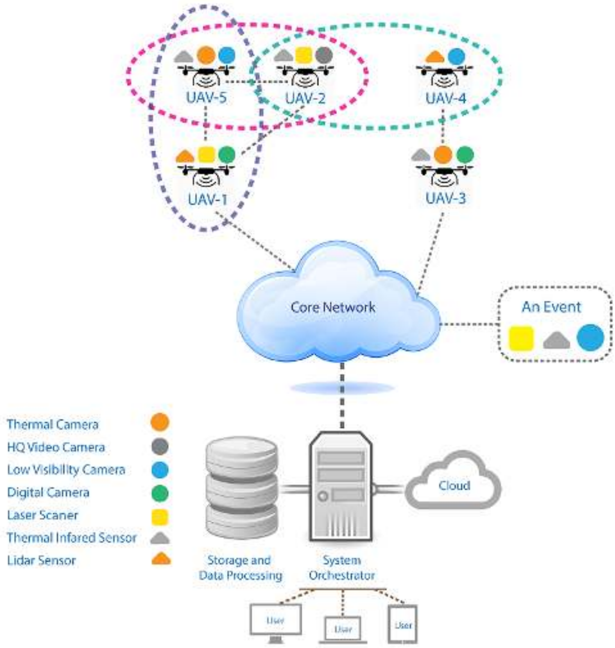


Figure 3.1. Network architecture and equipment-based UAV set selection.

camera, and thermal infrared sensor. In this example, the UAV selection process becomes more complex, since no single UAV has these devices on board. Thus, a group (cluster) of UAVs with the right IoT devices must be selected. Using this example and referring to Fig. 3.1, the possible groups of such UAVs are $\{u_1, u_5\}$, $\{u_2, u_5\}$, and $\{u_2, u_4\}$.

After constructing the possible sets of UAVs, the best set should be chosen. The requirement for the best set selection is based on the overall required time and energy to accomplish the task. The optimal set of UAVs is the cluster that needs the minimum amount of time and consumes the lowest amount of energy [39]. In fact, the overall required time for a UAV operation depends on the travel time of the UAV to the position of an event, the time it takes to perform the IoT task, i.e. sensing and data processing, and the data transmission times. Similarly, the overall required energy for task completion [20] is computed based on the amount of energy needed for the UAVs to travel, perform the IoT task, and transmit the data. Moreover, the energy consumption and the required time to perform an IoT event are best modeled by considering the environmental effects on the UAVs' flights and operations. Therefore, in the models for a UAV selection mechanism in this dissertation, the effect of wind on UAVs' travel, the effect of temperature on the IoT devices, and the effect of pathloss and shadowing on UAVs' data transmissions are considered.

3.2 Energy consumption and required time

In this sub-chapter, the energy consumption and the operation time are formulated for a UAV when selected to perform an IoT task. In the model, the following considerations are taken into account: $i)$ the UAV's travel, $ii)$ the UAV's sensing and processing, and $iii)$ the UAV's data transmission. In the model, u denotes a UAV, $\mathcal{N}$ denotes the set of UAVs in a UAV network, and $\mathcal{E}$ denotes an event [40].

3.2.1 The UAV's travel

In order to consider a realistic energy consumption model for UAVs, in this sub-section, the effect of wind on a UAV's speed is evaluated. In practice, the wind effects result from the interactions of the UAV's body b and the surrounding air a and depend on a vector that represents the wind w. Let us also denote the air-relative velocity vector of a UAV u by V_u^a, where the magnitude of V_u^a is called the airspeed V_a. The direction of the wind w is defined by the direction of V_u^a with respect to the body b and is also defined by two angles α and β, Where, the angle β is called the angle of attack that defines the longitudinal stability of the UAV. In addition, the angle between the velocity vector V_a of the UAV u and the longitudinal axis x_b of the UAV is called the side-slip angle and denoted by α, as shown in Fig.3.2. The transformation from the UAV's body frame b to the wind frame w is obtained through a sequence of UAV rotations, denoted by $(R_1(0), R_2(\alpha), R_3(-\beta))$ and given by [41]:

$$R_1(0) = 1, \quad R_2(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix}, and \ R_3(-\beta) = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

The rotation matrix is computed using the following relation:

$$R_b^w = R_1(0)R_2(\alpha)R_3(-\beta) = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \tag{3.2}$$

Let $V_u^w = \begin{bmatrix} V_a & 0 & 0 \end{bmatrix}^T$ be the inertial velocity vector measured in the direction of the wind axis. Having the components u, v, and w, the velocity vector of UAV relative to the surrounding air can be expressed as: $V_u^a = \begin{bmatrix} u & v & w \end{bmatrix}^T$, where $(V_u^a = R_w^b \ V_u^w)$. Using Equation 3.2 yields:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \begin{bmatrix} V_a \\ 0 \\ 0 \end{bmatrix} \tag{3.3}$$

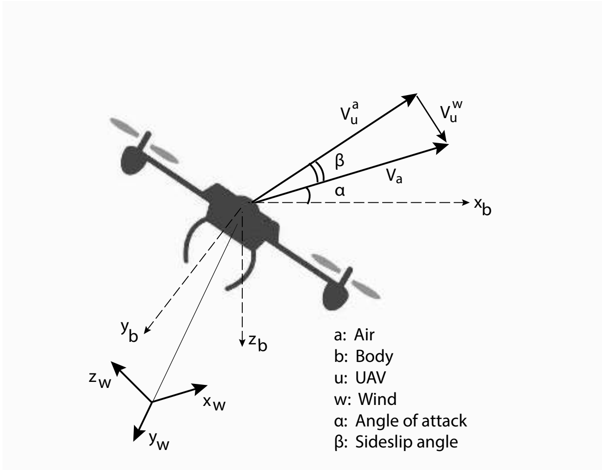


Figure 3.2. *A*ir-relative velocity and applied wind for a UAV.

Therefore, the airspeed components of the V_u^a are obtained through:

$$\begin{bmatrix} u & v & w \end{bmatrix}^T = V_a \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \end{bmatrix}^T \tag{3.4}$$

Thus, the airspeed V_a, and the angles α and β can be computed by:

$$V_a = \sqrt{u^2 + v^2 + w^2} \tag{3.5}$$

$$\alpha = \arctan(\frac{w}{u}) \tag{3.6}$$

$$\beta = \arcsin(\frac{v}{V_a}) \tag{3.7}$$

Let $u \in \mathcal{N}$ denote a UAV in $\mathcal{N}$. Let $\mathcal{D}_{u,\mathcal{E}}$ denote the distance from the UAV u to the event $\mathcal{E}$. Formally, to compute the distance $\mathcal{D}_{u,\mathcal{E}}$, the Euclidean three-space formulation is used as follows:

$$\mathcal{D}_{u,\mathcal{E}} = \sqrt{(x_{\mathcal{E}} - x_u)^2 + (y_{\mathcal{E}} - y_u)^2 + (z_{\mathcal{E}} - z_u)^2} \tag{3.8}$$

Then, the travel time Υ_u^{Travel} from the position of the UAV to the event position $\mathcal{E}$ is calculated by:

$$\Upsilon_u^{Travel} = \frac{\mathcal{D}_{u,\mathcal{E}}}{V_a} \tag{3.9}$$

To compute the energy consumption for traveling ξ_u^{Travel}, the following relations are defined:

$$\lambda = \frac{\xi_u^{Battery}}{\Upsilon_u^{Endurance}} \tag{3.10}$$

$$\xi_u^{Travel} = \lambda \cdot \Upsilon_u^{Travel} \tag{3.11}$$

where λ, $\xi_u^{Battery}$, and $\Upsilon_u^{Endurance}$ are the energy consumption per second, the full battery amount of the UAV u, and the maximum flight time that a UAV u can fly, respectively.

3.2.2 The UAV's sensing and processing

In UAV operations, keeping IoT devices operating continuously drains energy from the batteries. Thus, the IoT devices on board UAVs should be remotely turned on and off as needed; for the sake of energy efficiency, on board sensors and cameras must be actuated only when UAVs fly over the intended areas at particular times of interest. In addition, UAVs may be used in environments that have higher temperatures, e.g. in hot desert climates or above industrial steam pipes. Then, to calculate the energy consumption by IoT devices on board UAVs, considering the effect of temperature on the CPU of these devices, the total CPU power consumption P_{CPU} for each UAV per device is computed as follows [42]:

$$P_{CPU} = \sum_{i=0}^{n} (P_{leak} + P_{charge} + P_{short}) \tag{3.12}$$

where P_{leak} is the leakage power, P_{charge} is the power to charge the capacitors, and P_{short} is the short-circuit power. In addition, i to n refers to the number of IoT devices on board the UAV to be considered for CPU power consumption. Based on [43], P_{charge} is defined as follows:

$$P_{charge} = \mu \cdot C \cdot f \cdot V^2 \tag{3.13}$$

where μ is a constant that refers to the amount of systems active and switching modes. C is the capacitance, V is the voltage swing across C, and f is the switching frequency. Formally, P_{short} can be approximated as:

$$P_{short} = (\eta - 1) \cdot P_{charge} \tag{3.14}$$

where η is a scaling factor and represents the effects of short-circuit power. Normally P_{leak} originates from the leakage current I_{leak} and is computed as [44]:

$$P_{leak} = I_{leak} \cdot V_{dd} \tag{3.15}$$

where V_{dd} is the supply voltage of the circuit. Formally, a UAV's energy consumption by IoT devices in their active mode is computed as follows:

$$\xi_u^{SenseProcess} = P_{CPU} \cdot \Delta_t \tag{3.16}$$

where Δ_t is the duration of time that the device is active and $\xi_u^{SenseProcess}$ is the energy consumption of sensing and processing by the device.

3.2.3 The UAV's communication

Communication modeling: This sub-chapter models the communications between a UAV u and an eNodeB $\mathcal{B}$. In the model, an automatic repeat request (ARQ) scheme is used for data transmissions. This is used to enhance the communication reliability. The ARQ scheme works such that it sends a packet until it successfully arrives at the destination address. This means that a maximum number of retransmissions M is performed, where M varies randomly based on the channel conditions. In the communication, a combined path loss and shadowing model is used while the effect of interference is neglected. For the radio frequency propagation of the transmitter of the UAV u, typical urban and suburban environments and the optimal UAV altitude are considered. In the combined path loss and shadowing model, the ratio of received to transmitted power in (dB) is given by [45]:

$$\frac{P_r}{P_t}(db) = 10\log_{10} K - 10\varphi \log_{10} \frac{\mathcal{D}_{u,\mathcal{B}}}{\mathcal{D}_0} + n_{\mathcal{B}} \tag{3.17}$$

where P_r and P_t are the received and transmitted powers at the UAV u, respectively. φ is the path loss exponent, and K is a unitless constant that depends on the antenna characteristics. $\mathcal{D}_{u,\mathcal{B}}$ refers to the distance between the transmitter u and receiver $\mathcal{B}$, $\mathcal{D}_0$ is the reference distance, and $n_{\mathcal{B}}$ is a Gaussian-distributed random variable with zero mean and variance $\sigma^2_{\psi_{dB}}$.

Theorem. *A UAV $u \in \mathcal{N}$ fails to transmit its packet to an eNodeB $\mathcal{B}$ iff P_r falls below a given target minimum received power γ_{th}. However, with shadowing the received power P_r at any given distance $\mathcal{D}_{u,\mathcal{B}}$ from the transmitter is log-normally distributed with some probability of falling below γ_{th}. The outage probability $p_{out}(\gamma_{\text{th}}, \mathcal{D}_{u,\mathcal{B}})$ under path loss and shadowing is defined as the probability that the received power at a given distance $\mathcal{D}_{u,\mathcal{B}}$, $Pr(\mathcal{D}_{u,\mathcal{B}})$, falls below $\gamma_{\text{th}} : p_{out}(\gamma_{\text{th}}, \mathcal{D}_{u,\mathcal{B}}) = p(Pr(\mathcal{D}_{u,\mathcal{B}}) \leqslant \gamma_{\text{th}})$. This outage probability is expressed as [45]:*

$$\mathcal{P}_{u,\mathcal{B}} = p\big(P_r(\mathcal{D}_{u,\mathcal{B}}) \leqslant \gamma_{\text{th}}\big)$$

$$= 1 - Q\Big(\frac{\gamma_{\text{th}} - (P_t + 10\log_{10} K - 10\varphi \log_{10} \frac{\mathcal{D}_{u,\mathcal{B}}}{\mathcal{D}_0})}{\sigma_{\psi_{dB}}}\Big). \tag{3.18}$$

Proof. Here, the proof for the outage probability ($\mathcal{P}_{u,\mathcal{B}}$) between u and $\mathcal{B}$ in the UAV network is derived. By definition, the link $u - \mathcal{B}$ is in outage if P_r falls below a threshold level γ_{th} [46]. To determine an expression for $\mathcal{P}_{u,\mathcal{B}}$, let us consider that the shadow fading is modeled with log-normal distribution, i.e. $X \sim N(\mu, \sigma^2)$. For this distribution, the probability that the signal falls below the level x is obtained from $p(X \leqslant x) = 1 - Q(X) = 1 - Q(\frac{x-\mu}{\sigma})$. Let us recall that n_{dB} is a Gauss-distributed random variable with mean zero and variance $\sigma^2_{n_{dB}}$, noting that $\frac{P_r}{P_t}(dB) = P_{r(dB)} - P_{t(dB)}$. Considering that for the path loss and shadow fading, the outage probability can be derived by replacing $x = \gamma_{\text{th}}$, $\sigma = \sigma_{\psi_{dB}}$, and $\mu = P_t + 10\log_{10} K - 10\varphi\log_{10}\frac{\mathcal{D}_{u,\mathcal{B}}}{\mathcal{D}_0}\sigma_{\psi_{dB}}$, resulting in: $\mathcal{P}_{u,\mathcal{B}} = 1 - Q\big(\frac{\gamma_{\text{th}}-(P_t+10\log_{10}K-10\varphi\log_{10}\frac{\mathcal{D}_{u,\mathcal{B}}}{\mathcal{D}_0})}{\sigma_{\psi_{dB}}}\big)$. □

Communication time modeling: The IoT gateways on board UAVs are assumed to use a buffering system to handle the data packets $\mathcal{K}$ for the transmission. These packets hold the data about the sensed information from an event $\mathcal{E}$. Here, the aim is to analyze the average transmission time $\Upsilon_u^{Transmit}$ of sensed information from the UAV u to an eNodeB $\mathcal{B}$. Let $T_u^{Transmit}$ denote the sojourn time of a packet before its transmission to $\mathcal{B}$. Formally,

$$\Upsilon_u^{Transmit} = \mathcal{K} \cdot T_u^{Transmit}, \tag{3.19}$$

In fact, a successful reception of a packet at eNodeB $\mathcal{B}$ may require a random number of packet retransmissions. To evaluate the delay time in relation to the retransmission events, the average sojourn time $T_u^{Transmit}$ of a packet in the buffer of the transmitter u should be measured. The average sojourn time $T_u^{Transmit}$ of a packet is defined as the average time elapsed from the start of its transmission until the successful reception at the receiver. The packet's sojourn time in the buffer can be computed using the Pollaczek-Khinchin equation as in [47]:

$$T_u^{Transmit} = \mathbb{E}(N_{u,\mathcal{B}})T_F, \tag{3.20}$$

where T_F is the required time for a single transmission of a packet and $\mathbb{E}(N_{u,\mathcal{B}})$ is the average number of retransmissions of the packets transmitted from u. In an ARQ scheme, a maximum number of retransmissions M of a single packet is performed until a successful reception at the eNodeB $\mathcal{B}$. Note that a packet is discarded if it fails to be received after M retransmissions. Moreover, the number of retransmissions $N_{u,\mathcal{B}}$ changes randomly based on the UAV's position and the channel conditions between the transmitter u and the receiver $\mathcal{B}$. The average number of retransmissions $\mathbb{E}(N_{u,\mathcal{B}})$ can be computed by [48]:

$$\begin{aligned}\mathbb{E}(N_{u,\mathcal{B}}) &= 1 + \sum_{m=1}^{M-1} P(F^1, ..., F^m) = 1 + \sum_{m=1}^{M-1} (\mathcal{P}_{u,\mathcal{B}})^m \\ &= \sum_{m=0}^{M-1} (\mathcal{P}_{u,\mathcal{B}})^m = \frac{1-(\mathcal{P}_{u,\mathcal{B}})^M}{1-\mathcal{P}_{u,\mathcal{B}}},\end{aligned} \tag{3.21}$$

where $P(F^1, ..., F^m)$ refers to the probability of a reception failure at the $1^{st}, \ldots, m^{th}$ retransmissions. Since the channel realizations in each transmission are independent and identically distributed, the events of reception failures at each step are independent and have equal probabilities, thus $P(F^1, ..., F^m) = (\mathcal{P}_{u,\mathcal{B}})^m$. Using Equations 3.20 and 3.21 results in:

$$\Upsilon_u^{Transmit} = \mathcal{K} \cdot T_F \cdot \mathbb{E}(N_{u,\mathcal{B}}) = \mathcal{K} \cdot T_F \cdot \frac{1-(\mathcal{P}_{u,\mathcal{B}})^M}{1-\mathcal{P}_{u,\mathcal{B}}}. \tag{3.22}$$

Energy consumption model in communication: This sub-chapter aims to study the energy consumption ($\xi_u^{Transmit}$) needed for the packet transmissions at UAV u. It is worth noting that to compute this energy consumption, neither the energy required for sensing and processing nor the energy needed for the travel of the UAV u has been taken into account. Let us assume that a UAV has $\mathcal{K}$ packets to transmit, whereby these packets represent the sensed information about an event $\mathcal{E}$. Since the number of retransmissions changes based on the channel conditions, these changes make the consumed power a random variable. For this reason, in this sub-chapter, first average consumed power ($\bar{P}$) is evaluated, and then using the average power consumption, the average energy consumption is computed. In an ARQ scheme, the average power consumption $\bar{P}$ can be expressed as:

$$\begin{aligned}\bar{P} &= P_u \cdot P(S^1) + 2P_u \cdot P(F^1, S^2) + ... \\ &+ (M-1)P_u \cdot P(F^1, ..., S^{M-1}) + MP_x \cdot P(F^1, ..., F^{M-1}) \\ &= P_u \cdot \left(1 + \sum_{m=1}^{M-1} P(F^1, ..., F^m)\right) = P_u \cdot \left(1 + \sum_{m=1}^{M-1} (\mathcal{P}_{u,\mathcal{B}})^m\right) \\ &= P_u \cdot \mathbb{E}(T_{u,\mathcal{B}}) = P_u \cdot \frac{1-(\mathcal{P}_{u,\mathcal{B}})^M}{1-\mathcal{P}_{u,\mathcal{B}}},\end{aligned} \tag{3.23}$$

where P_u is the power consumption per retransmission at UAV u. Let $P(S^1)$ be the probability of successful reception at $\mathcal{B}$ of the first transmission, while $P(F^1, ..., S^{M-1})$ denotes the probability of a reception failure in the 1^{st}, 2^{nd}, ..., $(M-2)^{th}$ retransmissions and a successful reception at the $(M-1)^{th}$ retransmission. A probability of an event $P(S^1)$ means that a packet is successfully received after the first transmission and the power consumption is equal to P_u. Accordingly, a probability of an event $P(F^1, S^2)$ means that a packet is received correctly after two retransmissions and the power consumption for this event equals $2P_u$. Consequently, a probability of the event $P(F^1, ..., F^{M-1})$ explains that the 1^{st}, $\ldots, (M-1)^{th}$ retransmissions have failed and the power consumption is equal to MP_u. Then, the average power consumption is achieved by the sum of all possible values of power consumption and weighted by their respective probability of occurrence. The result in equation (3.23) shows that the average power consumption can be explained by the product of the power per retransmission P_u and the average number of retransmissions $\mathbb{E}(T_{u,\mathcal{B}})$. Hence, the average energy consumption $\Phi_{u,\mathcal{B}}$ of a single packet transmission is achieved by:

$$\begin{aligned}\Phi_{u,\mathcal{B}} &= P_u \cdot T_F \cdot P(S^1) + 2P_u \cdot T_F \cdot P(F^1, S^2) + ... + (M-1)P_u \\ &\cdot T_F \cdot P(F^1, ..., S^{M-1}) + MP_u \cdot T_F \cdot P(F^1, ..., F^{M-1}) \\ &= P_u \cdot T_F \cdot \left(1 + \sum_{m=1}^{M-1} (\mathcal{P}_{u,\mathcal{B}})^m\right) = P_u \cdot T_F \cdot \mathbb{E}(T_{u,\mathcal{B}}) = \bar{P} \cdot T_F.\end{aligned} \tag{3.24}$$

Therefore, the average energy consumption $\xi_u^{Transmit}$ to transmit the whole data packet ($\mathcal{K}$) can be obtained as:

$$\xi_u^{Transmit} = \mathcal{K} \cdot \Phi_{u,\mathcal{B}} = \mathcal{K} \cdot \bar{P} \cdot T_F. \tag{3.25}$$

3.3 Proposed solutions for UAV selection to carry out IoT tasks

In order to select the appropriate UAVs to handle various IoT events, as shown in Fig.3.3, in this sub-chapter, an SO-based UAV selection mechanism is proposed. Let $\mathcal{E}$ denote incoming IoT events and let $\mathcal{E}_{\Gamma}$ denote the events scheduled by the selection mechanism. Here, incoming events are events that have recently happened in the network, and scheduled events are events that have been already assigned to UAVs. Meanwhile, in the selection process, the SO should take into account the following constraints when selecting various UAVs: First, it has to consider the priority level of different events, the location of the events, and the type(s) of IoT devices required by those events. Then, it should select the most suitable UAVs for each event, considering the UAVs' locations, their residual energy amount, the types of IoT devices on board the UAVs, and the number of IoT devices mounted on the UAVs. Note that a UAV may be equipped with different types of devices, such as temperature and humidity sensors, at the same time.

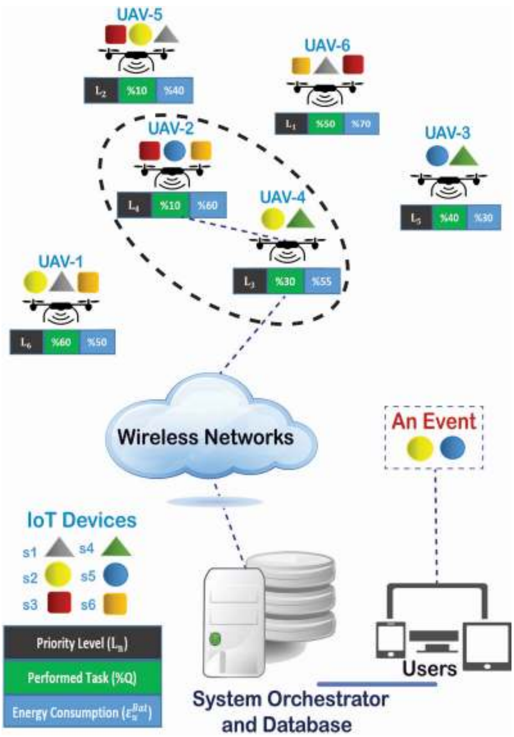


Figure 3.3. *U*AV selection mechanism based on the constraints.

Taking into account these considerations, the proposed selection mechanism is based on four steps, as shown in Fig. 3.4. In the first step, the incoming events are sorted according to their priority levels. In the second step, for each event level, constraints and rules are applied for selecting the eligible UAVs to perform the events of that level. In the third step, a subset of the eligible UAVs is selected to perform those events. In this step, three optimization solutions are proposed: i) the first solution aims to minimize as much as possible a UAV's energy consumption regardless of its operation time; ii) the second solution minimizes the UAV's operation time regardless of its energy consumption; and iii) the third solution aims to find a fair trade-off between both the UAV's energy consumption and its operation time using bargaining game. In the fourth step, the scheduled events $\mathcal{E}_\Gamma$ are refined. This sub-chapter explains the first, second, and fourth steps. To avoid a long sub-chapter, the third step, which includes the optimization problems, is discussed separately in sub-chapter 3.4.

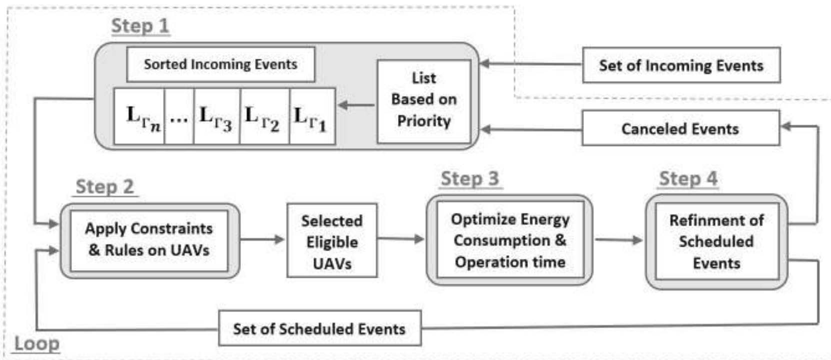


Figure 3.4. *T*he process of an eligible UAV selection mechanism for handling an incoming event.

3.3.1 First step: selection of eligible UAVs for incoming events

The IoT services (events) requested by users have different priorities, i.e. some events require more urgent handling than others. For example, streaming video of dangerous driving by a wicked or drunk driver in a street or an autobahn has a higher priority than reporting the temperature of a region. Thus, to prioritize service requests, a priority level is assigned to each one using the vector $L = \{L_1, L_2, L_3, \cdots, L_n\}$. The index of each priority refers to the priority level of the event. Note that the events that have higher priority levels should be scheduled first. Meanwhile, if multiple events have the same priority level, then they should be scheduled simultaneously. Therefore, the proposed mechanism groups and sorts events according to their priority levels, such that the events with the highest priorities are considered first. Then, the sorted events are passed on to the next step.

3.3.2 Second step: applying constraints and rules

To handle an event $\mathcal{E}$, let us denote by $\mathcal{S}_\mathcal{E}$ the set of IoT devices required by the event and by $\mathcal{S}_u$ the set of IoT devices on board a UAV u. Let $m\mathcal{H}_u$ denote the maximum altitude at which a UAV u is allowed to fly. Formally, a set of eligible UAVs ($\ddot{\mathcal{N}}$) should be selected to satisfy the following conditions:

- **UAV selection constraint:** A UAV should handle only one event at a given time, i.e. a UAV cannot be at multiple locations at the same time to handle different events. Formally, if denoting with $\ddot{\mathcal{N}}_i$ and $\ddot{\mathcal{N}}_j$ the set of UAVs that are selected for handling the events i and j, respectively, then $\ddot{\mathcal{N}}_i \cap \ddot{\mathcal{N}}_j = \emptyset$.
- **Device constraint:** The eligible UAVs must have the required IoT devices (s) to handle a scheduled event $\mathcal{E}$. If $\ddot{\mathcal{N}}_\mathcal{E}$ denotes the set of UAVs that will handle the event $\mathcal{E}$, then $\mathcal{S}_\mathcal{E} \subseteq \bigcup\limits_{u \in \ddot{\mathcal{N}}_\mathcal{E}} \mathcal{S}_u$, where $\mathcal{S}_\mathcal{E}$ denotes the set of devices required to handle the event $\mathcal{E}$ and $\mathcal{S}_u$ denotes the set of IoT devices on board a UAV u.
- **Energy constraint:** The selected UAVs should have enough energy resources to handle an event $\mathcal{E}$, i.e. $\xi_u^{Battery} > \xi_{u,e}^{ToT}$.
- **Time constraint:** The time latency of each selected UAV $(u \in \ddot{\mathcal{N}})$ should not exceed the threshold time Υ_{th} when handling a scheduled event $\mathcal{E} \in \mathcal{E}_\Gamma$.
- **Altitude constraint:** A UAV u has the ability to fly at the altitude of the event $\mathcal{E}$. Formally, $m\mathcal{H}_u \geq Z_\mathcal{E}$; $m\mathcal{H}_u$ is the maximum UAV flight altitude and $Z_\mathcal{E}$ is the altitude of the event $\mathcal{E}$.

In addition to the above-mentioned constraints, a set of rules $(\mathcal{R})$ is also defined for selecting the eligible UAVs for already scheduled events. $\mathcal{R}$ is variable and can be defined according to different use cases by the designers of the selection mechanism. It is worth noting that after applying the constraints and rules, if one or more UAVs that are currently assigned a scheduled event $\mathcal{E}_\Gamma$ are selected, then the event $\mathcal{E}_\Gamma$ would be stopped, canceled, or postponed for the next rounds.

Example. *An example of a rule defined for the UAV selection process.*

$< \mathcal{R} : \forall\, e_\Gamma \in \mathcal{E}_\Gamma, \forall u \in \ddot{\mathcal{N}}_{e_\Gamma} : L_\mathcal{E} > L_{\mathcal{E}_\Gamma}, \exists s \in \mathcal{S}_u \cap \mathcal{S}_\mathcal{E}, \mathcal{Q}_{e_\Gamma} < \mathcal{Q}_{th}, \xi_u^{Battery} > \xi_{u,e}^{ToT} \Rightarrow \textit{Select } u >$ *where* $\mathcal{Q}_{e_\Gamma}$ *denotes the number of already accomplished tasks for a scheduled event* e_Γ*, and* $\mathcal{Q}_{th}$ *denotes that the predefined threshold should not be exceeded before canceling a scheduled event* e_Γ*. In this example, to handle an incoming event (*$\mathcal{E}$*), the rule* $\mathcal{R}$ *is defined to enable the selection of the UAVs that are already busy with the ongoing tasks (*$\forall u \in \ddot{\mathcal{N}}_{e_\Gamma}$*). Note that in this rule, the UAVs that are already handling the ongoing scheduled events would be selected as eligible for performing incoming events. As mentioned above, in step three, only a subset of eligible UAVs would be selected for performing the incoming events. Thus, if the UAVs for a scheduled event* e_Γ *do not belong to that subset, then* e_Γ *should be kept and should not be canceled. From this rule, the UAVs assigned the scheduled event* e_Γ *would be considered eligible for an incoming event* $\mathcal{E}$ *iff:* $i)$ *the event* $\mathcal{E}$ *has a higher priority than the ongoing event* e_Γ*,* $ii)$ *the UAV u has some devices required by* $\mathcal{E}$*, and* $iii)$ *the number of already accomplished tasks from the event* e_Γ *should not exceed* $\mathcal{Q}_{th}$ *(this rule is suggested to save the energy of various UAVs, as a canceled event would be postponed and can be handled from scratch), and* $iv)$ *the remaining energy at UAV* u *(* $\xi_u^{Battery}$*) should exceed the required energy for performing the event* $\mathcal{E}$ *(*$\xi_{u,e}^{ToT}$*). This means that a UAV's energy should be sufficient to handle the event* $\mathcal{E}$*.*

3.3.3 Fourth step: refinement of the scheduled event(s)

In the fourth step, after the designation of the subset of eligible UAVs, refinement should be performed. This step is executed to inform the various UAVs about their new tasks and cancel their current scheduled events. Let $\mathcal{U}$ denote the selected subset of eligible UAVs. Formally, $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$, whereby $\mathcal{U}_1$ is the set of free UAVs that do not have previous tasks, and $\mathcal{U}_2$ is the set of UAVs that are already assigned other scheduled events. In this step, the SO would send the notification to the selected UAVs $\mathcal{U}$ to perform their tasks for various events. Moreover, the SO would cancel the events that are the responsibility of various UAVs $\mathcal{U}_2$. The canceled event should be listed again in the queue of incoming events to be executed later.

3.4 Optimal solutions for UAV selection (third step)

In this sub-chapter, three solutions are proposed to select a subset of eligible UAVs to handle various incoming tasks. Through this selection, UAVs handle the incoming events $\mathcal{E}_L$ of a specific level L. Linear integer problem optimizations are used in these solutions to obtain the optimal allocation of UAVs to carry out diverse IoT tasks. The first solution aims to minimize as much as possible the energy consumption needed to handle the events. The second solution aims to minimize the time needed to accomplish the UAV's tasks regardless of energy consumption. The third optimization aims to find a fair trade-off between the two conflicting objectives, i.e. the UAV's energy consumption and its operation time. Let $\xi_{u,\mathcal{E}}^{ToT}$ and $\Upsilon_{u,\mathcal{E}}^{ToT}$ denote the expected energy consumption and operation time, respectively, for a UAV u to perform an event $\mathcal{E}$. Formally,

$$\xi_{u,\mathcal{E}}^{ToT} = \xi_{u,\mathcal{E}}^{Travel} + \xi_{u,\mathcal{E}}^{SenseProcess} + \xi_{u,\mathcal{E}}^{Transmit}, \tag{3.26}$$

where $\xi_{u,\mathcal{E}}^{Travel}$, $\xi_{u,\mathcal{E}}^{SenseProcess}$ and $\xi_{u,\mathcal{E}}^{Transmit}$ are energy consumption for the travel, sensing and processing, and data transmission, respectively. Similarly,

$$\Upsilon_{u,\mathcal{E}}^{ToT} = \Upsilon_{u,\mathcal{E}}^{Travel} + \Upsilon_{u,\mathcal{E}}^{SenseProcess} + \Upsilon_{u,\mathcal{E}}^{Transmit}, \tag{3.27}$$

where $\Upsilon_{u,\mathcal{E}}^{Travel}$, $\Upsilon_{u,\mathcal{E}}^{SenseProcess}$ and $\Upsilon_{u,\mathcal{E}}^{Transmit}$ are the time duration required for the travel, sensing and processing, and transmission, respectively.

3.4.1 Optimization of energy consumption

This sub-chapter proposes a solution, dubbed energy-aware UAV selection (EAUS), that aims to minimize as much as possible the energy consumption needed to handle the various IoT events. Let $\mathcal{X}_{u,\mathcal{E}}$ be a boolean decision variable that equals 1 if a UAV $u \in \ddot{\mathcal{N}}$ handles an event$\mathcal{E} \in \mathcal{E}_L$; otherwise it equals 0.

$$\mathcal{X}_{u,\mathcal{E}} = \begin{cases} 1 & \text{If } u \text{ is selected to handle an event } \mathcal{E} \\ 0 & \text{Otherwise} \end{cases} \tag{3.28}$$

The EAUS solution aims to select the minimum number of UAVs while ensuring that the time latency does not exceed a predefined threshold Υ_{th}. The EAUS solution is formulated through the following linear integer optimization problem (**OP1**):

$$\begin{cases} \textbf{min} \sum\limits_{u \in \ddot{\mathcal{N}}} \sum\limits_{\mathcal{E} \in \mathcal{E}_L} \xi^{ToT}_{u,\mathcal{E}} \cdot \mathcal{X}_{u,\mathcal{E}} \\ \quad \textbf{s. t.} \\ \quad \forall \mathcal{E} \in \mathcal{E}_L, \forall s \in \mathcal{S}_{\mathcal{E}} : \sum\limits_{u \in \ddot{\mathcal{N}} \wedge s \in \mathcal{S}_u} \mathcal{X}_{u,\mathcal{E}} \geq 1 \\ \quad \forall u \in \ddot{\mathcal{N}} : \Upsilon^{ToT}_{u,\mathcal{E}} \cdot \mathcal{X}_{u,\mathcal{E}} \leq \Upsilon_{th} \\ \quad \forall \mathcal{E} \in \mathcal{E}_L, \forall u \in \ddot{\mathcal{N}}, \forall \mathcal{E}' \in \mathcal{E}_L, \mathcal{E} \neq \mathcal{E}' : \mathcal{X}_{u,\mathcal{E}} + \mathcal{X}_{u,\mathcal{E}'} \leq 1 \\ \quad \forall \mathcal{E} \in \mathcal{E}_L, \forall u \in \ddot{\mathcal{N}} : \mathcal{X}_{u,\mathcal{E}} \in \{0, 1\} \end{cases} \tag{3.29}$$

The objective of this optimization is to minimize the energy consumption of the selected UAVs. Meanwhile, the constraints of the optimization ensure the following statements. The first constraint ensures that the selected UAVs have the required IoT devices to deal with the events $\mathcal{E}_L$. The second constraint ensures that the time latency of each selected UAV $u \in \ddot{\mathcal{N}}$ does not exceed the threshold Υ_{th} when handling each event $\mathcal{E} \in \mathcal{E}_L$. The third constraint ensures that a UAV $u \in \ddot{\mathcal{N}}$ should not handle two disparate events at the same time. This means that one UAV cannot be at two variant locations at the same time. The last constraint ensures that $\mathcal{X}_{u,\mathcal{E}}$ is a boolean decision variable.

3.4.2 Optimization of operation time

For the second solution, DAUS is proposed, which aims to minimize as much as possible a UAV's operation time, while ensuring that the expected energy consumption during the missions does not exceed a predefined threshold Υ_{th}. Let $\mathcal{X}_{u,\mathcal{E}}$ be a boolean decision variable that equals 1 if a UAV $u \in \ddot{\mathcal{N}}$ is selected to handle an event $\mathcal{E} \in \mathcal{E}_L$. Otherwise, $\mathcal{X}_{u,\mathcal{E}}$ equals 0.

$$\mathcal{X}_{u,\mathcal{E}} = \begin{cases} 1 & \text{If } u \text{ is selected to handle an event } \mathcal{E} \\ 0 & \text{Otherwise} \end{cases} \tag{3.30}$$

The DAUS solution aims to minimize the operation time of the UAVs while maintaining the total energy consumption of UAVs below a predefined threshold ξ_{th}. This solution is formulated through the following linear integer optimization problem (**OP2**):

$$\begin{cases} \textbf{min} \ \ \mathcal{Z} \\ \quad \textbf{s. t.} \\ \quad \forall \mathcal{E} \in \mathcal{E}_L, \forall s \in \mathcal{S}_{\mathcal{E}} : \sum\limits_{u \in \ddot{\mathcal{N}} \wedge s \in \mathcal{S}_u} \mathcal{X}_{u,\mathcal{E}} \geq 1 \\ \quad \sum\limits_{u \in \ddot{\mathcal{N}}} \sum\limits_{\mathcal{E} \in \mathcal{E}_L} \xi^{ToT}_{u,\mathcal{E}} \cdot \mathcal{X}_{u,\mathcal{E}} \leq \xi_{th} \\ \quad \forall \mathcal{E} \in \mathcal{E}_L, \forall u \in \ddot{\mathcal{N}} : \Upsilon^{ToT}_{u,\mathcal{E}} \cdot \mathcal{X}_{u,\mathcal{E}} \leq \mathcal{Z} \\ \quad \forall \mathcal{E} \in \mathcal{E}_L, \forall u \in \ddot{\mathcal{N}}, \forall \mathcal{E}' \in \mathcal{E}_L, \mathcal{E} \neq \mathcal{E}' : \mathcal{X}_{u,\mathcal{E}} + \mathcal{X}_{u,\mathcal{E}'} \leq 1 \\ \quad \forall \mathcal{E} \in \mathcal{E}_L, \forall u \in \ddot{\mathcal{N}} : \mathcal{X}_{u,\mathcal{E}} \in \{0, 1\} \end{cases} \tag{3.31}$$

The objective of this solution aims to minimize UAVs' operation time.

Meanwhile, the constraints ensure the following statements. The first constraint ensures that the selected UAVs have the required IoT devices to deal with each event $\mathcal{E} \in \mathcal{E}_L$. The second constraint ensures that the energy consumption of selected UAVs should not exceed the threshold ξ_{th}. The third constraint aims to find the maximum operation time $\mathcal{Z}$. The fourth constraint ensures that each UAV $u \in \ddot{\mathcal{N}}$ should not handle two disparate events at the same time. The last constraint ensures that $\mathcal{X}_{u,\mathcal{E}}$ is a boolean decision variable.

3.4.3 Fair trade-off between energy consumption and operation time using a bargaining game

For the third solution, FTUS is proposed, which aims to find a fair trade-off between the two conflicting objectives, which are energy consumption and operation time. It is worth noting that a bargaining game is used to find a fair trade-off between the players (the conflicting objectives). Then, in FTUS, the UAV energy consumption and operation time are considered as two players that would like to barter goods.

Definition (Cooperative Games). *In cooperative games, the players are assumed to attain either a most desirable point when negotiation succeeds or a disagreement point when negotiation fails. For the game, let us consider two people who would like to barter goods, and each one wants to increase his benefits. Let Φ be the vector payoff of these players. Formally, $\Phi = \{(\mathcal{O}_1(x), \mathcal{O}_2(x)), x = (x_1, x_2) \in \mathcal{X}\}$, where $\mathcal{X}$ is the set of the two players' strategies and $\mathcal{O}_1(x)$ and $\mathcal{O}_2(x)$ represent the utility functions of the two players, respectively. In addition, the nash bargaining model (NBM) [49] presents a cooperative game with non-transferable utility. This means that the utility scales of the players are measured in non-comparable units. A Nash bargaining game is based on two elements assumed to be given and known to the players. The first element is the set of vector payoffs Φ achieved by the players if they agree to cooperate. Φ should be a convex and compact set. The second element is the threat point, $\omega = (\mathcal{O}_1^{\omega}, \mathcal{O}_2^{\omega}) \in \Phi$, which represents the pair of utility, where the two players fail to achieve an agreement. In NBM, the aim is to find a fair and reasonable point, $(\mathcal{O}_1^*, \mathcal{O}_2^*) = \mathcal{F}(\Phi, \mathcal{O}_1^{\omega}, \mathcal{O}_2^{\omega}) \in \Phi$. Based on Nash theory, a set of axioms [49] are defined that lead to finding the unique Pareto-optimal solution $(\mathcal{O}_1^*, \mathcal{O}_2^*)$. Moreover, the unique solution $(\overline{\mathcal{O}}, \overline{\mathcal{V}})$ satisfying these axioms is proven to be the solution of the following optimization problem:*

$$\begin{cases} \max \quad (\mathcal{O}_1(x) - \mathcal{O}_1^{\omega})(\mathcal{O}_2(x) - \mathcal{O}_2^{\omega}) \\ \quad \textbf{\textit{s. t.}} \\ \quad (\mathcal{O}_1(x), \mathcal{O}_2(x)) \in \Phi \\ \quad (\mathcal{O}_1(x), \mathcal{O}_2(x)) \geq (\mathcal{O}_1^{\omega}, \mathcal{O}_2^{\omega}) \end{cases} \tag{3.32}$$

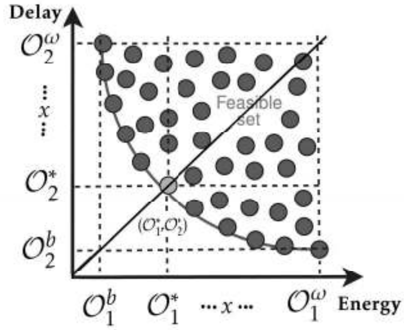


Figure 3.5. *F*TUS bargaining.

An enhanced solution for a Nash bargaining game, named Kalai Smorodinsky bargaining solution (KSBS), is proposed by Kalai and Smorodinsky [50]. The aim of KSBS is to enhance the fairness between the players by sharing the same utility fraction r among them. KSBS preserves the same Nash bargaining axioms except the independence of irrelevant alternatives. KSBS also has a new axiom called monotonical. In contrast to the Nash bargaining game and in addition to the disagreement point $\omega = (\mathcal{O}_1^\omega, \mathcal{O}_2^\omega) \in \Phi$, KSBS needs the ideal point $x^b = (\mathcal{O}_1^b, \mathcal{O}_2^b)/x^b \in \Phi$, the best utility that both players can achieve separately without bargaining. Kalai and Smorodinsky prove that the unique solution to satisfy KSBS's axioms is the solution of the following optimization problem:

$$\begin{cases} \textbf{max } r \\ \quad \textbf{\textit{s. t.}} \\ \quad (\mathcal{O}_1(x), \mathcal{O}_2(x)) \in \Phi \\ \quad r = \dfrac{\mathcal{O}_1(x) - \mathcal{O}_1^\omega}{\mathcal{O}_1^b - \mathcal{O}_1^\omega} \\ \quad r = \dfrac{\mathcal{O}_2(x) - \mathcal{O}_2^\omega}{\mathcal{O}_2^b - \mathcal{O}_2^\omega} \end{cases} \tag{3.33}$$

Therefore, using the definition of cooperative games, the followings describe the third solution, FTUS. Let $\omega = (\mathcal{O}_D^\omega, \mathcal{O}_E^\omega)$ and $b = (\mathcal{O}_D^b, \mathcal{O}_E^b)$ denote the threat and best points of the KSBS game for energy consumption and response time, respectively. As mentioned earlier, Υ_{th} and ξ_{th} denote the threshold values of operation time and energy consumption, respectively. In a KSBS game, both players, i.e. energy and delay, should bargain to increase their benefits, which is in conflict with and opposite to their utility function defined by the optimization problems (**OP1**) and (**OP2**). In order to use the KSBS game to ensure a fair trade-off between operation time and energy consumption, as depicted in Fig. 3.5, the utility function is inversed to be the smallest value better for both players. The fair Pareto-optimal solution FTUS is formulated as follows:

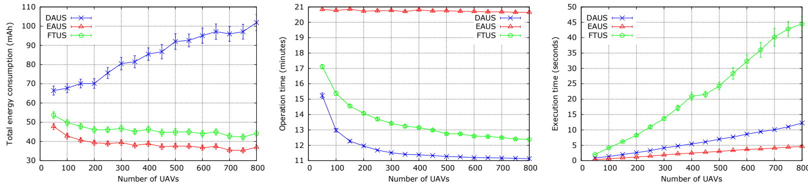
$$
\begin{cases}
\textbf{max } r \\
\quad \textbf{s. t.} \\
\quad \forall \mathcal{E} \in \mathcal{E}_L, \forall s \in \mathcal{S}_{\mathcal{E}} : \sum\limits_{u \in \ddot{\mathcal{N}} \wedge s \in \mathcal{S}_u} \mathcal{X}_{u,\mathcal{E}} \geq 1 \\
\quad \sum\limits_{u \in \ddot{\mathcal{N}}} \sum\limits_{\mathcal{E} \in \mathcal{E}_L} \xi^{ToT}_{u,\mathcal{E}} \cdot \mathcal{X}_{u,\mathcal{E}} \leq \mathcal{O}^{\omega}_{E} \\
\quad \forall \mathcal{E} \in \mathcal{E}_L, \forall u \in \ddot{\mathcal{N}} : \Upsilon^{ToT}_{u,\mathcal{E}} \cdot \mathcal{X}_{u,\mathcal{E}} \leq \mathcal{O}^{\omega}_{D} \\
\quad \forall \mathcal{E} \in \mathcal{E}_L, \forall u \in \ddot{\mathcal{N}}, \forall \mathcal{E}' \in \mathcal{E}_L, \mathcal{E} \neq \mathcal{E}' : \mathcal{X}_{u,\mathcal{E}} + \mathcal{X}_{u,\mathcal{E}'} \leq 1 \\
\quad \forall \mathcal{E} \in \mathcal{E}_L, \forall u \in \ddot{\mathcal{N}} : \Upsilon^{ToT}_{u,\mathcal{E}} \cdot \mathcal{X}_{u,\mathcal{E}} \leq \mathcal{O}_D(x) \\
\quad \mathcal{O}_E(x) = \sum\limits_{u \in \ddot{\mathcal{N}}} \sum\limits_{\mathcal{E} \in \mathcal{E}_L} \xi^{ToT}_{u,\mathcal{E}} \cdot \mathcal{X}_{u,\mathcal{E}} \\
\quad r = \frac{\mathcal{O}^{\omega}_{D} - \mathcal{O}_D(x)}{\mathcal{O}^{\omega}_{D} - \mathcal{O}^{b}_{D}} \\
\quad r = \frac{\mathcal{O}^{\omega}_{E} - \mathcal{O}_E(x)}{\mathcal{O}^{\omega}_{E} - \mathcal{O}^{b}_{E}} \\
\quad \forall u \in \ddot{\mathcal{N}} : \mathcal{X}_{u,\mathcal{E}} \in \{0,1\}
\end{cases}
\tag{3.34}
$$

3.5 Performance evaluation

In this sub-chapter, the three proposed solutions, EAUS, DAUS and FTUS, using Python and Gurobi optimization tools are evaluated. In the simulations, each plotted point represents the average of 100 executions. The plots from these simulations are presented with a 95% confidence interval.

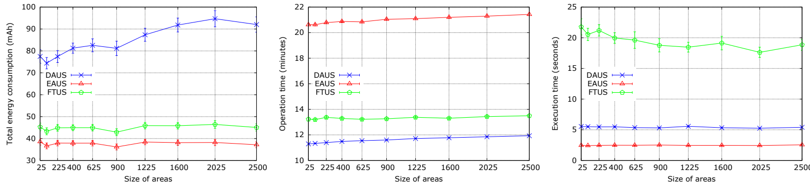
3.5.1 Network parameters

In the simulations, the solutions are evaluated in terms of the following criteria: energy consumption, operation time, and execution time. While the first and second evaluations are based on the energy and operation time of UAVs for accomplishing IoT tasks, the third criteria evaluates the required time for executing each solution. The different solutions are evaluated by varying $i)$ the number of UAVs, $ii)$ the size of the UAVs' flying area, $iii)$ the number of events, and iv) the types of IoT devices in the network (i.e. the IoT devices required by various events). In fact, four levels of priorities of events are considered in the simulations. The maximum altitude of UAVs is set to 300 meters. In Fig. 3.6, the proposed solutions are evaluated by varying the number of UAVs from 50 to 800 while setting $i)$ the number of events to 50, $ii)$ the size of area to 900 km^2, and $iii)$ the number of IoT devices to 10. Fig. 3.7 shows the performance of the three solutions by varying the size of the deployed area from 20 km^2 to 2500 km^2 while setting $i)$ the number of UAVs to 400, $ii)$ the number of events to 50, and $iii)$ the number of IoT devices to 10. Fig. 3.8 illustrates the performance of the proposed solutions by varying the number of IoT



(a) Energy consumption (b) Operation time (c) Execution time

Figure 3.6. Performance of the proposed solutions as a function of number of UAVs.



(a) Energy consumption (b) Operation time (c) Execution time

Figure 3.7. Performance of the proposed solutions as a function of size of areas.

devices while setting $i)$ the number of UAVs to 400, $ii)$ the number of events to 50, and $iii)$ the size of area to $900km^2$. The number of IoT devices is randomly selected from within an interval $[5, 100]$. Fig. 3.9 shows the performance of the three solutions by varying the number of events from 10 to 150 events while setting $i)$ the number of UAVs to 400, $ii)$ the size of the area to $900km^2$, and $iii)$ the number of IoT devices to 10.

3.5.2 Energy consumption

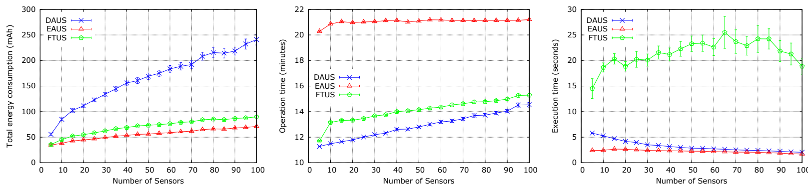
Figs. 3.6(a), 3.7(a) 3.8(a), and 3.9(a) show the performance evaluation of the three solutions, namely EAUS, DAUS, and FTUS, in terms of energy consumption. Fig. 3.6(a) shows that the number of UAVs has a positive impact on the energy consumption in the EAUS solution. Obviously, when the number of UAVs to handle a fixed number of IoT events is high in a fixed area, this high availability of UAVs would eventually enhance the handling of IoT events, reducing the travel distance of UAVs, which is the major source of energy consumption. Based on this figure, one can observe that the number of UAVs has a negative impact on the energy consumption in the DAUS solution. This is attributable to the fact that the DAUS solution aims to reduce operation time without taking into account energy consumption. Thus, increasing the number of UAVs consequently increases the probability for selecting new UAVs to handle new IoT events, which ultimately increases overall energy consumption. As mentioned in the previous sub-chapter, FTUS aims to find a fair trade-off between energy consumption and operation time. Therefore, in contrast to DAUS, FTUS exhibits a performance similar to that of EAUS in terms of energy consumption when the number of UAVs increases.

Fig. 3.7(a) shows the impact of the size of area on overall energy consumption. Based on this figure, the observation is that the area size does not have a high impact on the energy consumption in the case of EAUS. This is due to the fact that fixing the number of UAVs to 400 gives a high number of possibilities for EAUS to select suitable UAVs to carry out IoT events, and then the impact of the area size on energy consumption becomes minimal. Meanwhile, it is clear that the increase in the number of UAVs has a negative impact on the energy consumption in the DAUS solution. The solution aims to minimize operation time without caring about energy consumption. Increasing the number of UAVs would provide DAUS more chances to select UAVs with high energy consumption. In addition, FTUS shows a behavior similar to that of EAUS in terms of energy consumption.

Fig. 3.8(a) shows the impact of the number of IoT devices on energy consumption. This figure shows that energy consumption in EAUS increases slightly from 40 to 60 mAh. DAUS shows a rapid increase from 50 to 250 mAh, which is against savings in energy consumption. FTUS exhibits a performance similar to that of EAUS. Meanwhile, Fig. 3.9(a) illustrates the performance of the three solutions in terms of the number of IoT events. The first observation from this figure is that the increase in the number of events has a negative impact on energy consumption. This figure demonstrates that both EAUS and FTUS have better performance than DAUS in terms of energy consumption. All in all, the obtained results demonstrate the superiority of the EAUS and FTUS solutions in terms of energy efficiency compared to the DAUS solution.

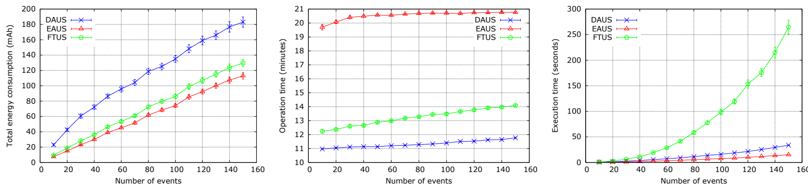
3.5.3 Operation time

Figs. 3.6(b), 3.7(b), 3.8(b), and 3.9(b) show the performance of the three solutions, i.e. EAUS, DAUS, and FTUS, in terms of the UAVs' operation time. Fig. 3.6(b) shows the impact of the number of UAVs on operation time. The first observation one can draw from this figure is that an increase in the number of UAVs has a positive impact on the operation time in both the DAUS and FTUS solutions. Increasing the number of UAVs gives both solutions more choices in selecting the UAVs that are closest to the target areas of various IoT events and then reducing the operation



(a) Energy consumption (b) Operation time (c) Execution time

Figure 3.8. Performance of the proposed solutions as a function of number of sensors.



(a) Energy consumption (b) Operation time (c) Execution time

Figure 3.9. Performance of the proposed solutions as a function of number of events.

time. In contrast to the DAUS and FTUS solutions, the aim of EAUS is to save energy consumption regardless of operation time; for this reason, changes in the number of UAVs do not have any impact on operation time in EAUS, as depicted in Fig. 3.6(b). Fig. 3.7(b) illustrates the impact of the size of the flying area on EAUS, DUAS, and FTUS in terms of operation time. From this figure, it is clear that both DAUS and FTUS outperform EAUS in terms of operation time.

Fig.3.8(b) shows the impact of the number of IoT devices on the operation time required by each solution. It is obvious that the number of IoT devices in the network has a negative impact on the operation time in both DAUS and FTUS. In fact, each IoT device needs some time to get the measurements. Therefore, increasing the number of IoT devices per event substantially affects the operation time needed. Fig. 3.9(b) depicts the performance of the three solutions in terms of operation time when varying the number of IoT events in the network. This figure explains that both DAUS and FTUS have better performance than EAUS. This figure shows that increasing the number of IoT events has a negative impact on the three solutions. The results obtained from Figs. 3.6(b), 3.7(b), 3.8(b), and 3.9(b) clearly show the superiority of both the DAUS and FTUS solutions over the EAUS solution. The obtained results from this sub-chapter and the previous one demonstrate the efficiency of each solution in achieving its main design objectives. These results also show the efficiency of FTUS in achieving a fair trade-off between energy consumption and operation time.

3.5.4 Execution time

Figs. 3.6(c), 3.7(c), 3.8(c), and 3.9(c) show the execution times of the three algorithms for different system input parameters. These simulations show that EAUS has the best performance in terms of execution time. The execution time of DAUS is twice that of EAUS, while FTUS requires more time. In fact, the execution time of EAUS is less than that of DAUS, as the optimization problem of EAUS has fewer constraints. While DAUS requires four sets of constraints, EAUS uses only three sets of constraints. In addition, FTUS needs more execution time than DAUS and EAUS due to the fact that *i)* it has more constraints and *ii)* it needs the execution of both solutions to get the threat and best points. From Figs. 3.6(c), 3.8(c), and 3.9(c), it is clear that the increase in the number of UAVs, events, and IoT devices has a negative impact on execution times. Actually, increasing the number of UAVs, events, and IoT devices leads to an increase in the number of variables and constraints on the optimization problems, which, in turn, has a negative impact on execution time. Based on Fig. 3.7(c), an increase in the size of flying areas does not have any impact on execution time. This is because, however the size of the flying areas increase, the number of variables and constraints in the system remains the same.

4. Reliable communications for UAVs

In order to overcome problems with network coverage limitations, data throughput, and high-speed Internet connectivity till the deployment of next-generation mobile networks, LTE-4G networks are the potential candidates to be used on board UAVs. Particularly, employing reliable networks such as LTE-4G becomes vital in real-time applications, e.g. a UAV crowd surveillance and video streaming use case. Although the use of LTE-4G networks on board UAVs provides the necessary requirements for them to perform a successful operation, the use of only one network connection is not sufficient to establish steady and reliable connectivity [51, 52]. Most network operators do not support total coverage for urban and rural areas, and the signal strength offered by these operators may vary in different places. There may even be some areas that are not covered at all by a specific operator. Thus, weak signal strengths may prevent UAVs from appropriately performing their tasks. For this reason, the use of only one network connection is not sufficient to establish steady and reliable connectivity. This chapter is developed to seek energy-efficient solutions to overcome UAV network coverage problems and lead to reliable UAV communications. To achieve this, a mechanism for steering connections to multiple mobile networks [53] is proposed. The benefit of employing such a mechanism is to select a mobile network that provides the best signal quality. This means that UAVs will be able to select the network operator with the highest received signal strength indicator (RSSI). The idea of the proposed mechanism is illustrated in Fig. 4.1.

4.1 Test-bed for connection steering mechanism

To study the feasibility and efficiency of the proposed connection steering mechanism between mobile networks, a test-bed in a laboratory environ-

Figure 4.1. Connection to two mobile network operators (MNO)s.

ment was developed. Fig. 4.2 shows this test-bed, which consists of the following equipment: one Raspberry Pi2 that works as the IoT gateway on board a UAV; a USB hub, which increases the number of USB ports; a web camera for video streaming; and two LTE-4G/3G USB modems, each loaded with the SIM cards of Finnish mobile network operators, namely Elisa and Sonera. For network connection, a wvdial point-to-point dialer [54] was used to enable an Internet connection through both modems. In the test-bed, a radio frequency (RF) chamber was used to attenuate the measured signal at the transmitters, i.e. 4G modems. In the experiment, the signal strength, i.e. RSSI, was measured from the USB modems using the Python programming language. In addition, transport control protocol (TCP) was used to perform a reliable transmission of video frames. In the experiment, the performance of the steering mechanism between two LTE-4G networks was studied in terms of energy consumption ($\mathcal{E}$) and packet transmission rates ($\mathcal{N}$) at the IoT gateway through changing RSSI values at the USB modems. To compute the $\mathcal{E}$, a TOE8842 dual power supply was used as input DC power of RPi and a $6\frac{1}{2}$ digit resolution Multimeter was used to measure the Current $\mathcal{I}$. The current consumptions were stored in Excel files through the KI-Tool, which was installed on a separate laptop. To calculate the $\mathcal{N}$ from the IoT gateway, the Wireshark tool was installed on a laptop (the client), and the TCP packets sent from the source IP addresses were filtered.

To test the proposed steering mechanism, a connection via the first 4G USB modem on the IoT gateway was established and a request for the video streams from the client was sent. To compute $\mathcal{E}$ and $\mathcal{N}$, the experiment was conducted in the following three steps. First, using the RF

Figure 4.2. Laboratory experiment (test-bed).

chamber and the attenuator, the RSSI values were adjusted 12 times in various ranges for the first 4G USB modem. These ranges held particular RSSI values for the first network operator. The video streaming was performed over the connected 4G link for a duration of 60 seconds. Then, $\mathcal{E}$ and $\mathcal{N}$ were computed for the first operator's network. In the next step, for the second mobile operator's network, the same approach was applied, i.e. $\mathcal{E}$ and $\mathcal{N}$ were computed for the same duration (60) seconds. Based on the experiments, the measured RSSI values were in the ranges of (RSSI-0.99 ; RSSI+0.99) each time in various RF chamber attenuations. In the last step, both 4G networks (Elisa and Sonera) were connected to the IoT gateway, and the same procedure as in steps one and two was applied. The connection via two networks was made by developing a code using Python that selects the best LTE network, i.e. the network with the stronger signal quality. Eventually, video streaming was established to record $\mathcal{E}$ and $\mathcal{N}$.

The Python code developed for the connection steering mechanism is shown in Fig. 4.3. The code was developed so that default values were assigned for the two networks. The camera was initialized and the window size defined for the camera. The RSSI value using the "comgt" command was received from the USB port of the RPi as a string and converted to a float number. The code uses a condition that if the received RSSI value is bigger than 22 it stays connected to the same USB port. But if the RSSI value is smaller than 22, using the system commands, the default IP $(ppp0)$ is deleted and the second IP address $(ppp1)$ is used as the communicator network. In addition, while there is no breakdown in the connection, the system stays self-connected and the images are transmitted from the camera.

```python
def streamVideo(self):
    ElisaTh = 10.0
    SoneraTh = 30.0
    pygame.camera.init()
    webcam = pygame.camera.Camera("/dev/video0",(320,240))
    webcam.start()
    typeDone = False

    while True:
        f = os.popen("comgt -d /dev/ttyUSB6 sig")
        value = f.readlines()
        value = value[0].split(":")[1].strip()
        try:
            RSSIvalue = float(value.replace(',','.'))
        except:
            print "Error when converting the string!"
            continue

        print(RSSIvalue)

        if float(RSSIvalue) >= 22.0:
            if typeDone == False:
                os.system("route del default")
                os.system("route add default dev ppp0")
                typeDone = True
        else:
            if typeDone == True:
                os.system("route del default")
                os.system("route add default dev ppp1")
                typeDone = False

        self.connect()
        image = webcam.get_image()
        data = pygame.image.tostring(image,"RGB")
        try:
            self.__connection.send(data)
        except:
            print "Error happens"

            self.disconnect()
```

Figure 4.3. The Python code developed for selecting the best LTE network.

4.2 Test-bed result analysis

The results from the measurements illustrated in Fig. 4.4 show the impact of current consumption $\mathcal{I}$ (over time in seconds) by the IoT gateway when one and two 4G USB modems with their networks are connected to RPi in silent and active modes, i.e. video streaming. The figure shows that the current consumption is approximately $15\ mAp/s$ when two modems are connected to an IoT gateway. It is obvious that this amount is very small compared to when the modems are in streaming modes. Thus, it is easy to conclude that having two 4G USB modems on board UAVs does not have a high impact on the UAVs' total energy consumption.

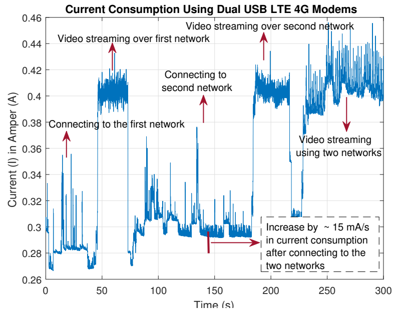
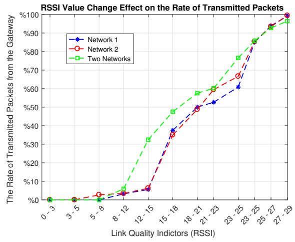


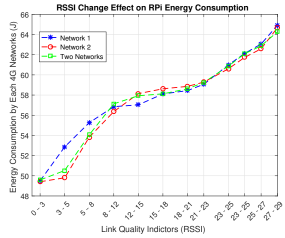
Figure 4.4. Current consumption using dual LTE modems.

The results from having a connection steering mechanism between two mobile networks for UAVs are shown in Fig. 4.5 in three subfigures, i.e. 4.5(a), 4.5(b), and 4.5(c). These results show that using each of the networks (Elisa and Sonera) results in almost similar performances in terms of $\mathcal{E}$ and $\mathcal{N}$ with changing RSSI qualities. Figures 4.5(a) and 4.5(b) show that the packet transmission rates $\mathcal{N}$ and energy consumptions $\mathcal{E}$ increase proportionally with particular RSSI value ranges (intervals). In the experiment, video streaming was done for 60 seconds per RSSI range, and the aim was to send a possible data packet rate $\mathcal{N}$ within this time per RSSI range (transmitting all the data packets was not the target). Thus, the rates of the transmitted packets within this time vary in the particular RSSI intervals.

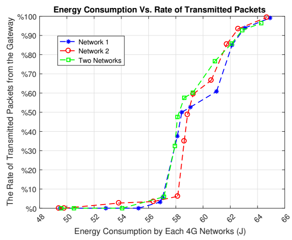
Fig. 4.5(a) shows the performance of $\mathcal{N}$ through RSSI value changes. The figure shows that $\mathcal{N}$ increases proportionally as RSSI strengthens. The reason is that, when there is a weak RSSI quality, the transmission bandwidth is low, so more packets are dropped behind the transmitter until the simulation time expires. In the same manner, when the signal (RSSI) strengthens, the bandwidth of the link increases, which results in the enhancement of $\mathcal{N}$. In addition, Fig. 4.5(b) shows that if RSSI quality at the IoT gateway is weak, the amount of $\mathcal{E}$ becomes low; correspondingly, if the RSSI quality strengthens, $\mathcal{E}$ increases. This is because of the TCP speed limits. In fact, when the RSSI quality is weak, the transmitter does not transmit and $\mathcal{E}$ remains low. In TCP, if a transmitter does not receive acknowledgement for its sent packet, it stops and waits a certain time until the next retransmission. Then, when the RSSI value is low, the receiver does not receive the acknowledgements (sent from the server) and stops sending packets until the waiting time is exceeded. Meanwhile, this continues until the experiment time (60 seconds) expires. Therefore,



(a) RSSI value change effect on the rate of transmitted packets



(b) RSSI change effect on RPi's energy consumption



(c) Energy consumption versus the rate of transmitted packets

Figure 4.5. Performances of the test-bed experiment.

in lower RSSI ranges less energy is consumed, since the transmitter is not sending and is in waiting mode. Fig. 4.5(c) demonstrates the relationship between the values of $\mathcal{E}$ and $\mathcal{N}$ that approves the results shown in Figs. 4.5(a) and 4.5(b).

However, the results presented in these figures show that at a specific energy consumption (e.g. more than $62J$), the received signal strength of both networks is high; thus, both networks demonstrate a high performance that assists the efficiency of the proposed mechanism. But considering lower energy consumption amounts (e.g. less than $62J$), the proposed mechanism using two networks shows better performance. This is clearly shown in Fig. 4.5(a) with the rate of transmitted packets from the UAV's IoT gateway. In conclusion, it must be stated that the main advantage of the connection steering mechanism is to increase the network reliability for UAV communications. This enhancement in network reliability is guaranteed by, first, obtaining a steady network connectivity for the variant UAVs, and second, increasing the UAVs' data transmission rates. These are achieved by selecting a stronger signal for the UAV's communication port. Thus, employing the proposed mechanism for the UAVs, especially in real-time applications, e.g. video streaming, enhances the reliability and efficiency of their services, i.e. QoS.

4.3 Performance evaluation

To evaluate the performance of a connection steering mechanism between multiple mobile networks, a real field experiment is necessary. In fact, for this dissertation, laboratory tests were done that did not allow any real UAV flight experiments to be done. Thus, due to the limitations caused by the test-bed, to validate the proposed connection steering mechanism, a discrete time Markov chain (DTMC) model was used to mimic the UAVs' mobility. In addition, to model the DTMC, the results from the experiments on the impact of RSSI changes on $\mathcal{E}$ and $\mathcal{N}$ for the two 4G networks (Elisa and Sonera) were used. In fact, based on [55], it was possible to group the quality of RSSI values into four quality ranges: poor, good, very good, and excellent. Then, using the Markov model, simulations were performed in a long-term run with distinctive transition probabilities, and the steady state of the DTMC model was calculated. The concept of DTMC is reviewed in sub-chapter 4.3.1; a Markov model along with its transition matrix and diagram is defined in sub-chapter 4.3.2; and the results of the DTMC model are analyzed in sub-chapter 4.3.3.

4.3.1 Discrete time Markov chain

A Markov chain is a discrete-time stochastic process. It is a sequence of stochastic events at different times, where the current state of a system is independent of all past states. Let us assume that X_n takes values from the state space $S = \{1, 2, ..., m\}$ [56].

Definition. *A Markov chain is a sequence of random variables such that the next state X_{n+1} depends only on the current state X_n. This means a sequence of random variables X_0, X_1, X_2,... take different values from the state space $\{1, 2, ..., m\}$. It is called a Markov chain if there is an m-by-m matrix $P = [p_{ij}]$, where for any $n \geqslant 0$,*

$$\begin{aligned} P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0) \\ = P(X_{n+1} = j | X_n = i) = p_{ij} \end{aligned} \tag{4.1}$$

The matrix P is called the transition matrix of the chain, and the p_{ij} is the transition probability from i to j. Generally, P_{ij} with state space of i and $j = 0, 1, \cdots, m$, is a one-step transition probability matrix and is constructed as:

$$P = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1m} \\ P_{21} & P_{22} & \dots & P_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1} & P_{m2} & \dots & P_{mm} \end{bmatrix} \quad (4.2)$$

where p_{ij} has the following properties, considering that p_{ij}^n has an n-step transition probability:

$$\begin{cases} \sum_j p_{ij} = 1 \\ 0 \leqslant p_{ij} \leqslant 1 \\ p_{ij}^n = P(X_n = j | X_0 = i) \end{cases} \quad (4.3)$$

Definition. *A row vector* $\pi = [\pi_1, \pi_2, ..., \pi_m]$*, such that* $\pi_i \geqslant 0$ *and* $\sum_i \pi_i = 1$*, is a steady-state distribution for a Markov chain with transition matrix* P *if*

$$\sum_i \pi_i p_{ij} = \pi_j \;\; equivalently \;\; \pi_i P = \pi_j \quad (4.4)$$

Note that an initial probability vector (a column vector) is constructed from the steady-state distribution matrix.

4.3.2 DTMC modeling for the mechanism

By defining a DTMC model, the aim is to validate the results obtained from the test-bed described in sub-chapter 4.1. The range of RSSI values measured from the test-bed vary within the interval $[0.99, 29.99]$, where each value from this range indicates the signal quality of the LTE-4G network. Based on [55], these values can be grouped into four quality ranges, presented in table 4.1. Thus, using this table, the state space that includes four states can be defined as $S = \{P, G, V, E\}$. According to the results shown in Fig. 4.5, the mechanism always selects the best signal quality or remains with the same network if it does not detect a stronger signal. This means, in the DTMC model, the states of the space intend to move or remain in the best state with a higher probability.

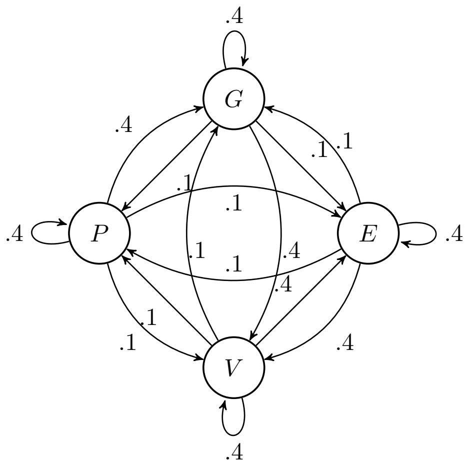
Table 4.1. RSSI Quality Ranges.

Quality	Ranges (α)
Poor (P)	$0.99 \leqslant \alpha \leqslant 8.99$
Good (G)	$9.99 \leqslant \alpha \leqslant 13.99$
Very Good (V)	$14.99 \leqslant \alpha \leqslant 18.99$
Excellent (E)	$19.99 \leqslant \alpha \leqslant 29.99$

Thus, employing the results of the experiments performed in this dissertation, the transition probability matrix P can be defined as in Fig. 4.5. This matrix explains: The state remains with the probability 0.4 in the same state and moves to a better state with the probability 0.4. The movement to another state happens when the model detects a stronger signal. In matrix P, the probabilities for moving to a better state or remaining in the same state are set to higher values compared to other transitions. Because the probability for a state to move to one with a better signal is higher than moving to one with a poor signal, the model always selects the higher signal strength if there is one.

Furthermore, the state values that are equal to 0.4 mean that those states persist in their current state (current received signal quality) and not changing to another one. Based on the results of the test experiments, the probabilities of other states are fixed to 0.1. For example, the movement from a poor to an excellent state rarely happens, and the probability that a state moves from "P" to "E" is taken to be 0.1. This is because the experiment showed that to move from "P" to "E" the state first moves to "G", "V", and "E" sequentially. The connection steering mechanism uses this approach to avoid dropping to a poor signal, as it should select the signals with higher qualities. It is worth noting that in matrix P, each row corresponds to the current signal quality, and each column corresponds to the next appearing signal qualities.

Therefore, the connection steering mechanism can be modeled as a four-state DTMC with the transition diagram shown in Fig. 4.6. This diagram graphically presents the same information provided by the transition matrix in equation 4.5. The four circles of the diagram represent the four states for the RSSI quality ranges defined in table 4.1. The arrows show the transition probability from one state to another. In a DTMC, a move to a better state happens if one of the modems detects a stronger signal. For instance, the state stays in "P" if both modems detect poor signals, and it moves to state "E" if at least one of the networks provides a strong signal. Moreover, the DTMC is an ergodic Markov chain, as it is irreducible, aperiodic, and positively recurrent. Then, DTMC has a unique steady-state that would be used for getting the simulation results, as discussed in 4.3.3.



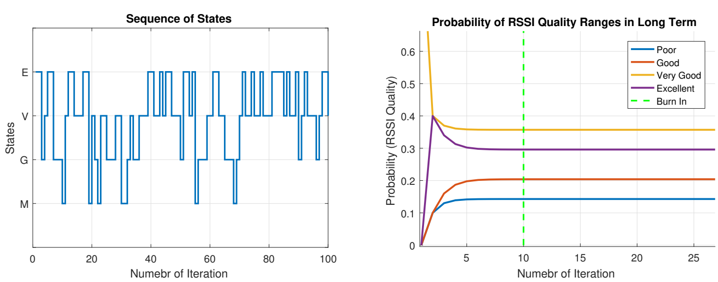
$$P = \begin{bmatrix} .4 & .4 & .1 & .1 \\ .1 & .4 & .4 & .1 \\ .1 & .1 & .4 & .4 \\ .1 & .1 & .4 & .4 \end{bmatrix} \tag{4.5}$$

Figure 4.6. DTMC transition diagram.

4.3.3 DTMC result analysis

The simulations in Fig. 4.7 explain the probabilities of being in various DTMC states in a long-term run. In addition, the plot in Fig. 4.7(a) illustrates the sequence of the states in diverse qualities. This figure shows what happens when the states move between the four qualities, i.e. from poor to excellent. It depicts that the model occupies the higher-quality states most of the time. This confirms that the overall performance of the DTMC model is high. Accordingly, Fig. 4.7(b) shows the probabilities of the four states in the long-term run. The observation is that in the second run, the probabilities of "V" and "E" start at 0.4. They continue so that the probability of "V" follows 0.38 and the probability of "E" continues at 0.3, both after reaching a stable point (almost 10 iterations).

The interpretation based on these results is that the mechanism's higher signal qualities (V and E) have higher probabilities than the lower signal qualities (G and P). Furthermore, these simulations confirm the results



(a) Sequence of states in a long-term run. (b) Probabilities in a long-term run.

Figure 4.7. Simulation results using a Markov chain.

obtained from the test-bed experiments, since an increase in RSSI increases the $\mathcal{E}$. Therefore, the mechanism consumes higher energy, and accordingly the rate of $\mathcal{N}$ is enhanced. In addition, the steady state of an ergodic Markov chain is a unique non-negative solution. In the Markov model, the steady state is:

$$[0.1250 \quad 0.1964 \quad 0.3750 \quad 0.3036]$$

This steady-state vector proves that the DTMC model is in state "P" with probability equal to 0.1250. The results show that the probabilities that the model is in states "G," "V," and "E" are 0.1964, 0.3750, and 0.3036, respectively. This output also approves the results of Fig. 4.5(a), meaning that the mechanism aims to select the higher qualities by moving to better states such as "V" and "E." Hence, the model consumes more energy and the packet transmission rate increases relatively, meaning that the performance of the proposed mechanism enhances and results in providing better services. Moreover, by having the state space S = {P, G, V, E}, and using the test-bed results of the connection steering mechanism, the $\mathcal{E}_{Average}$ and $\mathcal{N}_{Average}$ can be calculated as follows:

$$\mathcal{E}_{Average} = \sum_{s \in S} p_s \cdot \mathcal{E}_s \tag{4.6}$$

$$\mathcal{N}_{Average} = \sum_{s \in S} p_s \cdot \mathcal{N}_s \tag{4.7}$$

where $\mathcal{E}_{Average}$ is the average energy consumption and $\mathcal{N}_{Average}$ is the average rate of the transmitted packets. Let p_s denote the probability of state s in the steady-state vector. Using these equations leads to the following results: $\mathcal{E}_{Average} = 59.2559$ Joule, and $\mathcal{N}_{Average} = 57.7384 \times 10^3$. These results prove an efficient overall average performance of the DTMC model in all of the states. The results affirm a very good packet transmission rate $\mathcal{N}$ and guarantee a high QoS, which is mandatory in real-time applications, e.g. video streaming. The results also confirm an acceptable amount of energy consumption $\mathcal{E}$, which is a vital factor in a UAV's energy budget. Therefore, the connection steering mechanism offers an energy-efficient solution for UAV data communications. The output of the steady-state vector proves that the mechanism connects to the LTE networks with "very good" links with the highest probability, i.e. the overall performance of the mechanism is high. This means that the mechanism avoids unavailing efforts to transmit the data using lower-quality signals, i.e. "poor" and "good." As such, the mechanism avoids spending energy for unsuccessful transmissions. Therefore, UAVs do not need to perform data retransmissions, which avoids wasting their energy resources. Moreover, Fig. 4.4 showed that the amount of the current consumed using dual LTE modems (two 4G USBs) on board UAVs is negligible. Hence, the steering mechanism works as an energy-efficient solution for UAVs.

5. UAV data processing

In public places such as stadiums, protecting civilians from threats is an important concern. This protection can be achieved by anticipating crimes through the detection and recognition of criminals in crowds of people. Traditional patrolling systems need many security guards and significant human effort to provide the necessary safety. Fortunately, with recent advancements in aerial systems, UAVs can be employed to help security guards remotely surveil people at the desired places. Moreover, UAVs can also be used not just to control but to track, detect, and recognize criminals using face recognition methods. In such a use case, high-resolution cameras can be mounted on UAVs, and by applying face recognition techniques on the streamed videos, suspected people can be recognized in an efficient manner. But a major challenge in such a use case relates to the amount of energy consumption needed for the computations. Because of the computational overhead required by a face-recognition use case and given the limited power supply of UAVs, the processing of data collected by a UAV is a challenging issue. This is because a UAV's energy budget is limited.

Today, depending on the type of UAV, batteries available on the market do not allow UAV flights longer than 90 minutes, and that is without any processing being done on board [57]. Thus, to ensure longer flight times, the computational overhead on board UAVs should be as lightweight as possible. In the meantime, one effective solution can be offloading the heavy computations, e.g. video data processing, to an MEC node. However, depending on the underlying radio access technology (RAT), i.e. WiFi or LTE-4G, streaming videos from UAVs to an MEC node still requires a significant amount of energy. Thus, it is important to determine which data processes of which use cases should be performed on board UAVs and which can be offloaded to an MEC node. To study the effect of computation offloading vs. on board (local) processing of data in a UAV use case, this chapter investigates the benefits (or drawbacks) of the offloading process in terms of energy consumption and processing time. Thus, this chapter introduces an envisioned use case scenario. Then, the chapter analyzes the results of a UAV-based face recognition experiment developed in the laboratory environment. In the development, the local binary pattern histogram (LBPH) method from the open source computer vision (OpenCV) is employed to recognize suspicious faces [58].

5.1 An envisioned scenario for face recognition

UAVs equipped with high-resolution video cameras can be used for a face recognition use case. They can offer an efficient crowd surveillance system that detects any suspicious action and recognizes criminal faces in a crowd. Conclusively, the use of UAVs provides a bird's-eye view (face recognition) in a crowd surveillance use case. Therefore, crowd safety and security can be enhanced, while at the same time, the number of security guards deployed on the ground can be reduced. In this chapter, to implement a face recognition use case, the scenario schematically presented in Fig. 5.1 is envisioned. In this scenario, security guards access the control station and surveil a crowd of people. Upon noticing an uncommon behavior from a particular person (or group of persons), they command a UAV to take a video of the person(s) and apply face recognition on the cap-

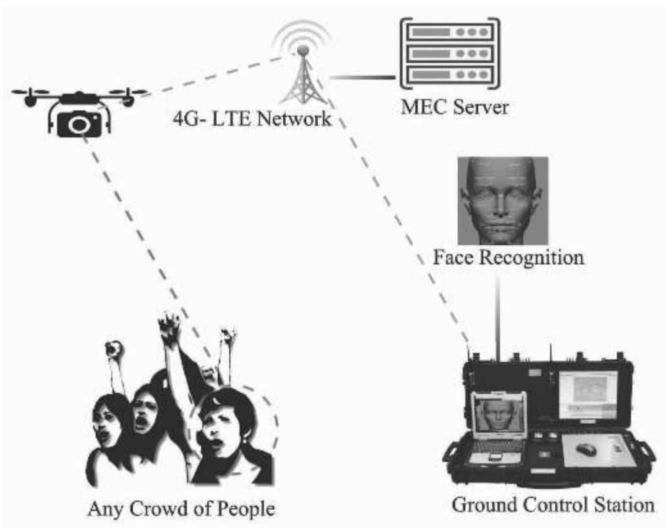


Figure 5.1. High-level diagram of the envisioned experiment scenario.

tured video to identify the suspicious person(s) and verify if he/they have any criminal records. In the scenario, a UAV equipped with a video camera and connected to the GCS through an LTE-4G network provides data communications. It is considered that the processing of recorded video for face recognition can be performed locally (on board the UAV) and at remote servers, i.e. enabling offloading of the face recognition operation to an MEC node. In fact, along with advances in communication technologies, MEC facilitates the offloading process from UAVs. Actually, the expectation is that MEC nodes are widely deployed in the network and UAVs do not need to travel to carry out the data offloads.

The envisioned scenario with its infrastructure is defined to implement a face recognition use case. The process of face recognition consists of well-defined steps, namely facial feature extraction, database creation of known faces, and face detection that matches the video-taped faces with profiled ones. In addition, many available video analytic tools can cope with the high mobility feature of UAVs and perform face recognition with high accuracy. For example, OpenCV is an outstanding software library that presents noticeable algorithms for face recognition. Using OpenCV, the face recognition of not only one but multiple faces at the same time becomes possible. OpenCV employs machine-learning techniques to search for profiled faces in a video frame. To perform precise face recognition, OpenCV uses LBPH with its associated libraries and databases. The approach of LBPH is to summarize the local structure in an image by comparing the pixels with its adjacent ones.

5.2 A face recognition test-bed

To find a solution for the challenge of UAV energy consumption for face recognition, this sub-chapter studies the impact of offloading the computation process versus local processing through a small scale test-bed, as shown in Fig. 5.2. The test-bed environment consists of a Raspberry Pi (RPi) that works as the local processing unit and a laptop that serves as the MEC node. The laptop works as the command and control station of the UAV gateway and RPi is used to turn the camera on and off or commanding it, whether to locally process the face recognition or to offload the processing to the MEC node. To perform face recognition, OpenCV's LBPH algorithm developed in Python was used to recognize particular faces from a database of 40 faces, each stored in a separate directory. It is

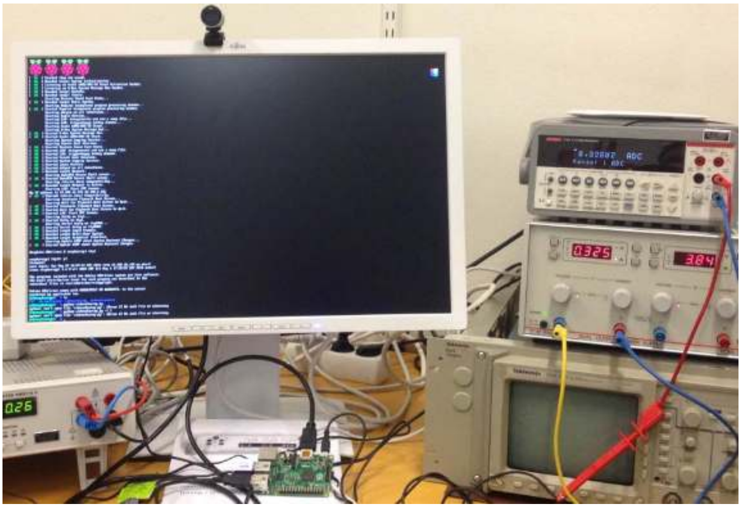


Figure 5.2. Test-bed for energy consumption measurement.

worth noting that each person has 10 different facial details and expressions (e.g. open or closed eyes, smiling, not smiling, face with or without glasses). In the experiments, the following hardware devices were used: a laptop (mobile edge) that includes an Intel Core i5 − 3570 CPU at a frequency of 3.40 GHz and a memory of 16 GB (RAM) and runs via Ubuntu version 16.04; and a RPi (locally embedded computing node) that has 1 GB of memory capacity, a CPU of 900 MHz, input power of micro-USB socket $5V/2A$, and a quad-core ARM Cortex-A53 architecture model. A TOE8842 power supply was used as the DC power generator for RPi, with input power set to 5 volts. For the energy consumption measurement, a 6-Digit Resolution Digital Multi-meter was used to measure the Current $\mathcal{I}$. In the experiment, using a camera, 10 videos of various lengths were taken from real life, each of which contained a group of people. The duration of the i^{th} video was i seconds (i.e. the duration of the fifth video was 5 seconds). Therefore, the performance in terms of energy consumption and processing time evaluated when the face recognition operation was carried out locally (on board the UAV) and when it was offloaded to the MEC node. For this evaluation, two experiments were conducted.

Fig. 5.3 depicts the results of the first experiment, whose aim was to recognize the faces of five suspected people while varying the video lengths. The figure illustrates that it is highly efficient to offload the face recognition operation to an MEC node rather than processing it locally on board an energy-limited UAV. In fact, local processing of the video data uses a considerable amount of energy and drains the UAV's battery. Moreover,

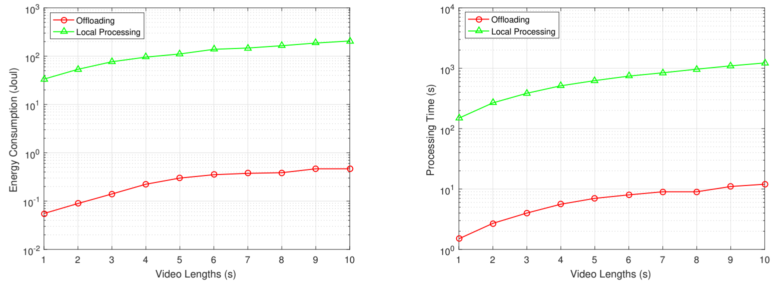


Figure 5.3. Performance evaluation when multiple videos with various lengths are processed locally on board a UAV and when their computation is offloaded to an MEC.

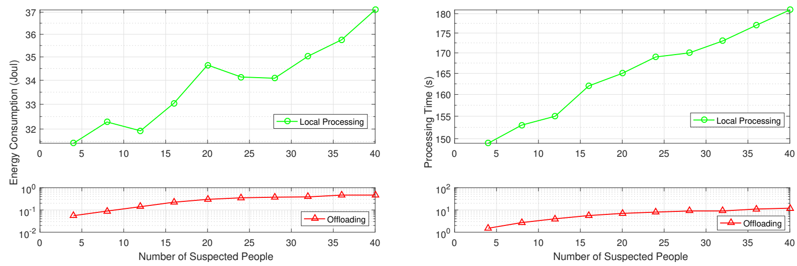


Figure 5.4. Performance evaluation when a 1s-long video is processed locally on board a UAV and when its computation is offloaded to an MEC to recognize various numbers of profiled persons.

the offloading process drastically reduces the processing time compared to performing the face recognition locally on board the UAV. The figure shows that the offloading process requires 100 times less energy consumption and processing time than local processing of video on board the UAV. Fig. 5.4 shows the results of the second experiment, when the video duration is set to one second $(1s)$ and the number of suspected people varied. The results demonstrate that when video processing is offloaded, the energy and the processing time required for a UAV remains the same regardless of the number of profiled persons. However, when the video is processed locally, the required energy and processing time increase somewhat linearly with the number of profiled people.

6. Discussion, conclusion, and future research

This chapter discusses some of the challenges regarding UAV-based IoT communication and the findings of this dissertation. The chapter also offers some suggestions for future research.

6.1 Discussion

The deployment of UAV-based IoT platforms and a central system orchestrator that enables the delivery of VAIoTSs from a given height were the key concepts of this dissertation. This deployment and offering VAIoTSs through a widespread network of flying UAVs requires overcoming many challenges. The initial challenge may concern finding and selecting a suitable set of UAVs to perform an IoT task. The selection of UAVs requires orchestrating them, collecting real-time data about their status, and computing the collected data through a set of mechanisms and algorithms. In addition, orchestration of UAVs becomes a significant concern, since they differ in size, geographic location, and capabilities such as flight endurance, energy resources, and the payloads on board.

In addition, for some IoT tasks or due to the geographical proximity of the UAVs and their flying altitude, a cluster of UAVs may be needed to do a task. Therefore, UAV clustering may entail many challenges such as *i*) how to construct UAV clusters for different use cases with diverse geographical proximities, *ii*) how to establish communication between flying UAVs through a FANET within a cluster (when the UAVs may use different communication technologies, such as a cellular system, Wi-Fi, or satellite), and *iii*) how to select a specific UAV in each cluster as a cluster head to manage the communication between the UAVs and perform the delivery of the collected data. Meanwhile, due to the dynamic of UAVs and their mobility feature in moving in 3D space with higher speeds, another challenge is enabling reliable and seamless wireless communication in UAV-to-GCS, UAV-to-UAV, UAV-to-cellular, and UAV-to-satellite in FANET scenarios. Moreover, data processing done by UAVs is another challenge because the energy budgets of UAVs are limited; it is a challenging issue to decide whether to process the collected or sensed data locally on board a UAV or deliver the data to the ground station.

This dissertation was developed to seek solutions to some of these challenges in order to offer VAIoTSs using UAVs. Different chapters of the dissertation have contributed to the current state of knowledge about UAVs with the following achievements. Chapter 2 introduced state-of-the-art works on UAVs and the IoT platform. This chapter discussed the possible communication technologies that can be used on board UAVs, the UAV communication networks, and data processing and computation. This chapter also highlighted UAVs' potential for the delivery of IoT services from the sky. It envisioned an efficient architecture including a central system orchestrator for the delivery of UAV-based IoT services and a novel UAV-based IoT platform, in which having the platform on board and being part of the envisioned communication architecture enables a UAV to become a potential candidate for offering VAIoTSs to clients. In order to select a UAV or a set of UAVs to perform IoT tasks, Chapter 3 proposed a UAV selection mechanism. The mechanism uses information about the geographical locations of UAVs, their on board IoT devices, their energy budget, and the priority levels of the events. The proposed mechanism in this chapter was designed based on three optimization solutions that select UAVs considering their total energy consumption, their operation time, and a trade-off solution between energy and time.

After selecting the appropriate set of UAVs to handle an IoT task, there needs to be a reliable communication link with the GCS or BS. Therefore, Chapter 4 proposed a mechanism for steering connections between multiple mobile networks. This mechanism was developed to overcome the problem of network coverage limitations and signal strength fluctuations. The mechanism guarantees steady connections to the BSs or GCSs. It also guarantees higher data transmission rates by selecting the network with the stronger signal (RSSI). These two features enable obtaining reliable links in UAV communications. The next step for the delivery of VAIoTSs is processing the data collected by the UAVs. For this reason, Chapter 5 introduced a crowd surveillance use case to study UAVs' data processing. Since the processing of video data requires a noticeable amount of energy, this chapter studied the offloading of video data processing to an MEC node and compared it to local processing of video data on board a UAV. The study proves that an MEC-based offloading approach is highly efficient in saving the scarce energy of UAVs. In conclusion, this dissertation, through the contributions in each of its chapters, contributes to the field of UAV-based IoT communications.

6.2 Conclusion

The research presented in this dissertation aimed to advance the current state of knowledge about UAV-based IoT communications. Thus, the dissertation developed acceptable answers to the following research questions, concerning 1) The UAVs' system orchestrator–based architecture model for the delivery of UAV-based VAIoTSs, 2) UAV selection mechanism, 3) UAVs' reliable communications, and 4) UAVs' MEC-based computation offloading.

To answer the first research question, Chapter 2, introduced a novel architecture for UAV-based IoT communication and an innovative UAV-based IoT platform that operates from the sky. This architecture consists of a system orchestrator that contains the information about the UAVs and the clients to perform an IoT task. To define this architecture, using Publication I, an extensive survey of UAVs and the UAV-based IoT communications was performed. Chapter 3 was developed to answer the second research question that was the UAV selection mechanism. This chapter proposed three optimization solutions of EAUS, DAUS, and FTUS. The EAUS aims to reduce UAVs' total energy consumption, DAUS aims to decrease UAVs' operational time, and FTUS aims to find a fair solution between energy and time. The results obtained from these solutions proved each solution's efficiency in achieving its key design objectives.

Chapter 4 was developed to answer the third research question. This chapter proposed a connection steering mechanism between mobile networks that enables reliable communications for UAVs. Employing this mechanism led to the following substantial achievements: *i*) Since the UAVs do not have to perform data re-transmissions, there was a considerable decrease in energy consumption; and *ii*) as UAVs select the network with the stronger signal quality, a higher data transmission is guaranteed. Thus, the proposed mechanism proved its appropriateness for providing reliable communications for UAV-based IoT communications. To answer the fourth research question, Chapter 5 investigated the UAVs' energy consumption by computation offloading. For this evaluation, a UAV-based crowd surveillance use case was proposed, developed, and tested. The results from the laboratory tests proved the benefits of computation offloading in saving UAVs' scarce energy, which is highly effective for prolonging a UAV's operation time.

6.3 Future research

UAVs are gaining much momentum due to the vast number of their applications. It is expected that the UAVs enter nearly every sector of our lives, including search-and-rescue operations, public surveillance, monitoring (e.g. oil and gas pipelines, road traffic, and agricultural activities), and disaster management. However, the efficient and safe deployment of the UAVs faces many challenging issues, such as physical collision, path planning, data collection, communication and networking, and data delivery. This dissertation has highlighted the following challenges for future research. An important challenge relates to the management of a massive number of UAVs because each UAV may host more than one IoT device on board, such as different types of sensors and cameras. Managing and controlling many devices on board and sometimes with conflicts of interest (e.g. taking a video and photos from two different angles from one fixed location) becomes a problem for the efficient introduction of UAV-based VAIoTSs. Therefore, efficient methods and algorithms to solve the concurrence of IoT devices on board should be proposed in order to realize and validate the vision of VAIoTSs.

Another significant challenge pertains to UAVs' power consumption, which can come from three main sources: *i*) the power consumed by the UAV itself, *ii*) the power consumed by the IoT devices on board, such as sensors and cameras, and *iii*) the power consumed for data communication to deliver collected data and for the command and control unit. Since, UAVs have a certain budget of energy, the power amount required for IoT devices and data communication should be highly controlled in order to minimize power consumption. Thus, there is a need for efficient algorithms and methods to control and manage IoT devices and communication to the network by turning on and off at the desired places or by specific application. The other challenge relates to UAVs' communication network security because a UAV may hold important and protected data. For example, jammers may intend to jam a GPS signal that navigates the path of the UAV. In the case of jamming, a UAV's communication is disrupted, and thus it loses its ability to determine its location, altitude, and travel direction. Therefore, methods to evade aerial jammers for communication are needed.

An important challenge in UAV applications is constructing and employing a cluster of UAVs in order to perform a specific task. For example, in the scenario of surveillance and face recognition, a cluster of UAVs may cooperate to recognize and detect criminals in a crowd of people. This could include taking photos of the faces of the people in the crowd and analyzing their facial features from different directions. In addition, a significant issue regarding clusters of UAVs relates to their data computations. In fact, in some scenarios, e.g. disaster management, that may happen in distant locations with no possibilities of ground computing facilities, a cluster of UAVs is needed to perform tasks by sharing their computation capabilities. Thus, efficient algorithms need to be developed and tested for such scenarios. Ultimately, in addition to the open research areas mentioned in this chapter, this dissertation suggests applying the proposed IoT platform for unmanned systems other than UAVs, such as automated cars and boats. Hence, this may open a new research platform with numerous research questions in the area of unmanned systems.

References

[1] S. W. Loke, "The internet of flying-things: Opportunities and challenges with airborne fog computing and mobile cloud in the clouds," *CoRR*, vol. abs/1507.04492, 2015. [Online]. Available: http://arxiv.org/abs/1507.04492

[2] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives," *IEEE Internet of Things Journal.*, vol. 3, no. 6, pp. 899–922, December 2016.

[3] S. Qazi, A. S. Siddiqui, and A. I. Wagan, "UAV Based Real Time Video Surveillance Over 4G LTE," in *International Conference on Open Source Systems Technologies (ICOSST)*, Lahore, Pakistan, December 2015, pp. 141–145.

[4] O. Vermesan and P. Friess, *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. Aalborg, Denmark: River Publishers Series in Communications, 2013.

[5] S. Colin, "Why Drones Are the Future of The Internet of Things?" 2016, [Online; accessed 06-July-2016]. [Online]. Available: http://droneanalyst.com/2014/12/01/drones-are-the-future-of-iot/

[6] Development Team, "Powerfull Cloud-based software," [Online; accessed 06-July-2017]. [Online]. Available: https://www.dronedeploy.com/

[7] M. Bauer, M. Boussard, N. Bui, F. Carrez, P. Giacomin, E. Ho, C. Jardak, J. D. Loof, C. Magerkurth, S. Meissner, A. Nettstrater, A. Olivereau, A. Serbanati, M. Thoma, J. W. Walewski, "Internet of things architecture: Iot-a," European Commission within the Seventh Framework Programme, Deliverable D1.4, 2012.

[8] S. Ouahouah, T. Taleb, J. Song, and C. Benzaid, "Efficient Offloading Mechanism for UAVs-based Value Added Services," in *IEEE International Conference on Communications (ICC)*, Paris, France, May 2017, pp. 1–6.

[9] E. W. Frew and T. X. Brown, "Airborne Communication Networks for Small Unmanned Aircraft Systems," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2008–2027, December 2008.

[10] B. Lee, J. Lee, Y. J. Lee, and S. Sung, "Development of Cellular Data Network Enabled Autonomous Rotary UAV," in *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, Gwangju, Korea, October 2013, pp. 719–722.

[11] H. Shariatmadari, R. Ratasuk, S. Iraji, A. Laya, T. Taleb, R. Jäntti, and A. Ghosh, "Machine-Type Communications: Current Ctatus and Future Perspectives Toward 5G Systems," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 10–17, September 2015.

[12] M. Batistatos, D. Zarbouti, G. Tsoulos, and G. Athanasiadou, "Flying Relays for 4G Service-on-Demand Applications," in *10th European Conference on Antennas and Propagation (EuCAP)*, Davos, Switzerland, April 2016.

[13] J. C. Fernandez, T. Taleb, M. Guizani, and N. Kato, "Bandwidth Aggregation-Aware Dynamic QoS Negotiation for Real-Time Video Streaming in Next-Generation Wireless Networks," *IEEE Transactions on Multimedia*, vol. 11, no. 6, pp. 1082–1093, Oct 2009.

[14] J. C. Fernandez, T. Talebt, K. Hashimoto, Y. Nemoto, and N. Kato, "Multipath Scheduling Algorithm for Real-Time Video Applications in Next-Generation Wireless Networks," in *2007 Innovations in Information Technologies (IIT)*, Nov 2007, pp. 73–77.

[15] D. K. Goldenberg, J. Lin, A. S. Morse, B. E. Rosen, and Y. R. Yang, "Towards Mobility as a Network Control Primitive," in *in Proceedings of the 5th ACM International Symposium on Mobile ad hoc Networking and Computing (MobiHoc' 04)*, Tokyo, Japan, May 2004, pp. 163–174.

[16] M. Grossglauser and D. N. C. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, August 2002.

[17] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *in Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc' 05)*, Illinois, USA, May 2005, pp. 300–308.

[18] M. A. Messous, S. M. Senouci, and H. Sedjelmaci, "Network Connectivity and Area Coverage for UAV Fleet Mobility Model with Energy Constraint," in *2016 IEEE Wireless Communications and Networking Conference (WCNC)*, Nevers, France, April 2016, pp. 1–6.

[19] T. Taleb, J. C. Fernandez, K. Hashimoto, Y. Nemoto, and N. Kato, "A Bandwidth Aggregation-Aware QoS Negotiation Mechanism for Next-Generation Wireless Networks," in *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*, Nov 2007, pp. 1912–1916.

[20] S. Koulali, E. Sabir, T. Taleb, and M. Azizi, "A Green Strategic Activity Scheduling for UAV Networks: A Sub-Modular Game Perspective," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 58–64, May 2016.

[21] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile Edge Computing Potential in Making Cities Smarter," *in IEEE Communications Magazine*.

[22] R. Joseph, "Unmanned Aircraft System (UAS) Traffic Management (UTM)," July 2016, [Online; accessed 10-October-2016]. [Online]. Available: https://utm.arc.nasa.gov/

[23] COMNET, “Department of Communications and Networking,” School of Electrical Engineering, Aalto University, Espoo, Finland, [Online; accessed 07-July-2017]. [Online]. Available: http://comnet.aalto.fi/en/

[24] D. E. Zheng and W. A. Carter, *Leveraging the Internet of Things for a More Efficient and Effective Military*. Washington, DC, USA: Paperback, Rowman and Littlefield Pub Inc, 2015.

[25] H. Mark, “Project Skybender: Google’s Secretive 5G Internet Drone Tests Revealed,” January 2016, [Online; accessed 04-Jun-2016]. [Online]. Available: http://www.theguardian.com/technology/2016/jan/29/project-skybender-google-drone-tests-internet-spaceport-virgin-galactic/

[26] N. Kamesh, Y. W. M. Gomathisankaran, and R. Pendse, “Airborne network: A cyber-physical system perspective,” in *Proceedings of the First ACM MobiHoc Workshop on Airborne Networks and Communications*, ser. Airborne ’12. New York, NY, USA: ACM, 2012, pp. 55–60. [Online]. Available: http://doi.acm.org/10.1145/2248326.2248340

[27] I. Jawhar, N. Mohamed, and J. Al-Jaroodi, “Data communication in linear wireless sensor networks using unmanned aerial vehicles,” in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, May 2014, pp. 43–51.

[28] K. Fall, “A delay-tolerant network architecture for challenged internets,” in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM ’03. New York, NY, USA: ACM, 2003, pp. 27–34. [Online]. Available: http://doi.acm.org/10.1145/863955.863960

[29] K. Fall, K. L. Scott, S. C. Burleigh, L. Torgerson, A. J. Hooke, H. S. Weiss, R. C. Durst, and V. Cerf. (2007) Delay-Tolerant Networking Architecture. [Online; accessed 15-January-2018]. [Online]. Available: https://tools.ietf.org/html/rfc4838/

[30] L. Gupta, R. Jain, and G. Vaszkun, “Survey of important issues in uav communication networks,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1123–1152, Secondquarter 2016.

[31] O. Sahingoz, “Networking models in flying ad-hoc networks (fanets): Concepts and challenges,” *Journal of Intelligent and Robotic Systems*, vol. 74, no. 1, pp. 513–527, April 2014. [Online]. Available: https://doi.org/10.1007/s10846-013-9959-7

[32] R. B. Chiaramonte and K. R. L. J. C. Branco, “Collision detection using received signal strength in fanets,” in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, May 2014, pp. 1274–1283.

[33] I. Bekmezci, I. Sen, and E. Erkalkan, “Flying ad hoc networks (fanet) test bed implementation,” in *2015 7th International Conference on Recent Advances in Space Technologies (RAST)*, June 2015, pp. 665–668.

[34] S. Temel and I. Bekmezc, “On the performance of flying ad hoc networks (fanets) utilizing near space high altitude platforms (haps),” in *2013 6th International Conference on Recent Advances in Space Technologies (RAST)*, June 2013, pp. 461–465.

[35] M. H. Tareque, M. S. Hossain, and M. Atiquzzaman, “On the routing in flying ad hoc networks,” in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sept 2015, pp. 1–9.

[36] K. Kumar, J. Liu, Y. Lu, and B. Bhargava, “A survey of computation offloading for mobile systems,” *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129–140, Feb 2013. [Online]. Available: https://doi.org/10.1007/s11036-012-0368-0

[37] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, “Communicating while computing: Distributed mobile cloud computing over 5g heterogeneous networks,” *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 45–55, Nov 2014.

[38] N. H. Motlagh, M. Bagaa, and T. Taleb, “UAV-based IoT Platform: A Crowd Surveillance Use Case,” *IEEE Communications Magazine*, pp. 128–134, Feb 2017.

[39] N. H. Motlagh, M. Bagaa, and T. Taleb, “UAV Selection for a UAV-based Integrative IoT Platform,” in *Proceedings of IEEE Globecom*, Washington, USA, December 2016.

[40] N. H. Motlagh, M. Bagaa, and T. Taleb, “UAVs Task Assignment Mechanism for UAV-based IoT Platform,” *submitted to IEEE Transactions on Vehicular Technology*, 2018.

[41] C.-A. B. Brezoescu, “Small lightweight aircraft navigation in the presence of wind,” Theses, Université de Technologie de Compiègne, Oct. 2013. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01060415

[42] N. V. Karel, G. Memmi, P. Jouvelot, and F. Coelhot, “Modeling the Temperature Bias of Power Consumption for Nanometer-Scale CPUs in Application Processors,” in *International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation*, Samos Island, Greece, July 2014.

[43] Y.-J. Xu, Z.-Y. Luo, X.-W. Li, L.-J. Li, and X.-L. Hong, “Leakage current estimation of CMOS circuit with stack effect,” *Journal of Computer Science and Technology - Special issue on computer graphics and computer-aided design*, vol. 19, no. 5, pp. 708–717, September 2004.

[44] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, “Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits,” in *Proceedings of the IEEE*, vol. 91, February 2003, pp. 305–327.

[45] A. Goldsmith, *Wireless Communications*. New York, NY, USA: Cambridge University Press, 2005.

[46] M. K. Simon and M.-S. Alouini, *Digital Communication over Fading Channels*, 2nd ed. New Jersey, USA: John Wiley & Sons, 2005.

[47] W. Chan, T.-C. Lu, and R.-J. Chen, “Pollaczek-Khinchin formula for the M/G/1 queue in discrete time with vacations,” *IEE Proceedings - Computers and Digital Techniques*, vol. 144, no. 4, pp. 222–226, Jul. 1997.

[48] A. Chelli, E. Zedini, M.-S. Alouini, J. R. Barry, and M. Pätzold, “Performance and delay analysis of hybrid ARQ with incremental redundancy over double Rayleigh fading channels,” *IEEE Trans. on Wireless Commun.*, vol. 13, no. 11, pp. 6245–6258, Nov. 2014.

[49] J. F. Nash, “The Bargaining Problem,” *Econometrica*, vol. 18, no. 2, pp. 155–162, 1950. [Online]. Available: http://www.jstor.org/stable/1907266

[50] E. Kalai and M. Smorodinsky, “Other Solutions to Nash’s Bargaining Problem,” *Econometrica*, vol. 43, no. 3, pp. 513–518, 1975. [Online]. Available: http://www.jstor.org/stable/1914280

[51] T. Taleb and A. Ksentini, “VECOS: A Vehicular Connection Steering Protocol,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1171–1187, March 2015.

[52] T. Taleb, A. Ksentini, and F. Filali, “Wireless Connection Steering for Vehicles,” in *2012 IEEE Global Communications Conference (GLOBECOM)*, Dec 2012, pp. 56–60.

[53] N. H. Motlagh, M. Bagaa, T. Taleb, , and J. Song, “Connection Steering Mechanism between Mobile Networks for reliable UAV’s IoT Platform,” in *IEEE International Conference on Communications (ICC)*, Paris, France, May 2017.

[54] Arch Linux Development Team (Wiki), “wvdial Configurations (Point-to-Point Protocol Dialer),” 2016, [Online; accessed 17-February-2017]. [Online]. Available: https://wiki.archlinux.org/index.php/Wvdial

[55] M2M-Support-Forum, “AT+CSQ AT command returns the signal strength of the device,” 2017, [Online; accessed 17-February-2017]. [Online]. Available: http://m2msupport.net/m2msupport/atcsq-signal-quality/

[56] J. R. Norris, *Markov Chains*. Cambridge University Press, October 1998.

[57] Microdrones, “UAV/Drone Equipment: Multispectral Camera, Thermal Imaging, GPS,” 2016, [Online; accessed 08-Mar-2016]. [Online]. Available: https://www.microdrones.com/en/products/equipment/

[58] OpenCV Development Team, “FaceRecognizer - Face Recognition With OpenCV,” 2016, [Online; accessed 12-Jun-2016]. [Online]. Available: http://docs.opencv.org/2.4/modules/contrib/doc/facerec/

Aalto-DD 18/2018

The deployment of UAV-based IoT platforms and a central system orchestrator that enables the delivery of value added IoT services from the sky is the key concepts of this dissertation. This deployment and offering IoT Services through a widespread network of flying UAVs requires overcoming many challenges. To seek solutions to some of these challenges, this dissertation contributes to the current state of knowledge about UAVs with the following achievements. The possible communication technologies that can be used on board UAVs, the UAV communication networks, and data processing are discussed. An efficient architecture including a central system orchestrator and a novel UAV-based IoT platform are envisioned. A UAV selection mechanism for selecting a UAV or a set of UAVs to perform IoT tasks and a mechanism for steering connections between multiple mobile networks to guarantee a reliable communication for UAVs are proposed. In addition, to study the UAVs' data processing, a crowd surveillance use case is introduced.

9789526078212

ISBN 978-952-60-7821-2 (printed)
ISBN 978-952-60-7886-1 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Electrical Engineering
Department of Communications and Networking
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS