HAMSTER – healthy, mobility and security-based data communication architecture for unmanned systems

Daniel Fernando Pigatto

Daniel Fernando Pigatto

HAMSTER – healthy, mobility and security-based data communication architecture for unmanned systems

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco
Co-advisor: Profa. Dra. Cristina Dutra de Aguiar Ciferri

USP – São Carlos
May 2017

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

Pigatto, Daniel Fernando
P628h HAMSTER - healthy, mobility and security-based
data communication architecture for unmanned
systems / Daniel Fernando Pigatto; orientadora
Kalinka Regina Lucas Jaquie Castelo Branco;
coorientadora Cristina Dutra de Aguiar Ciferri. --
São Carlos, 2017.
201 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2017.

1. Communications. 2. Computer networks. 3. Data
communication architecture. 4. Safety. 5. Security.
I. Branco, Kalinka Regina Lucas Jaquie Castelo,
orient. II. Ciferri, Cristina Dutra de Aguiar,
coorient. III. Título.

Daniel Fernando Pigatto

HAMSTER – arquitetura de comunicação de dados voltada à verificação de saúde, mobilidade e segurança para sistemas não tripulados

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco
Coorientadora: Profa. Dra. Cristina Dutra de Aguiar Ciferri

USP – São Carlos
Maio de 2017

To the ones who made this achievement possible (and truly amazing).

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Profa. Kalinka Branco, who has always been by my side regarding orientation, supporting, helping, sharing valuable knowledge and promoting my development as a researcher. Thank you very much for absolutely everything!

I would like to thank my British supervisor, Prof. Jim Smith, from UWE Bristol, who had very productive discussions about my project during my year as research visitor.

I would also like to thank my co-supervisor, Profa. Cristina Ciferri, for all valuable suggestions and discussions.

From the bottom of my heart I would like to thank my beloved partner, João Vitor, for being so supportive and dedicated throughout these years.

I would like to thank my family for understanding my professional choices. A special thank you to my parents Dinorá and Agenor, my sister Fernanda, my godparents, uncles, aunts, cousins and grandparents. I love you so much!

I would like to thank my laboratory mates throughout these years: Adimara, Arthur, Emerson, Julio, Matheus, Paulo, Querino and Rodolfo. A very special thank you to my closer lab mates Natássya, Rayner, Mariana and João for all the GREAT moments and experiences we had together and all the invaluable efforts towards the conclusion of this thesis.

I would like to thank my closest friends Alencar, Ana Paula, Danilo, Deluca, Emily, Felipe, Giuliana, João Paulo, Luana, Matheus, Murilo, Rodolfo, Valéria, and so many more, specially from São Carlos and Erechim, for being always there for me.

I would like to thank Fernanda Pigatto for the creative work on developing my architecture's logotype.

I would like to thank all the friends I have met during my year as research visitor in Bristol, specially Angélica, Draylson, Jessica, Jhony, Kamila, Nelissa, and Yin.

I would like to thank the undergraduate students that took part in this project and are equally responsible for the exciting results: Amir, Eduardo, Guilherme, Lucas and Vitor. Also, Guilherme and Leandro (UNESP S. J. do Rio Preto), and Prof. Alex (UFSC).

Finally, I acknowledge ICMC and USP for the infrastructure, and FAPESP for the financial support (process 2012/16171-6 and BEPE process 2014/13713-8).

“We’re all stories in the end.
Just make it a good one.”
(The Eleventh Doctor — Doctor Who)

RESUMO

PIGATTO, D. F. **HAMSTER – arquitetura de comunicação de dados voltada à verificação de saúde, mobilidade e segurança para sistemas não tripulados**. 2017. 201 p. Doctoral dissertation (Doctorate Candidate Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

Os avanços na área de comunicações foram indiscutivelmente essenciais para a obtenção de sistemas e aplicações modernos como os o atuais. A computação ubíqua se tornou realidade, permitindo que sistemas embarcados especializados ganhassem espaço e cada vez mais autonomia. Esse é notavelmente o caso de veículos não tripulados que têm sido criativamente explorados em aplicações inovadoras e avançadas. Entretanto, para o funcionamento eficiente desses veículos e sistemas não tripulados, além de melhorias de comunicação, é altamente desejável que as necessidades relevantes co-relacionadas a comunicação sejam cuidadosamente observadas, levando a uma facilitação na inserção de veículos não tripulados em espaços públicos. Além disso, ao abordar essas demandas de modo integrado, as chances de produzir melhores resultados é maior. Esta tese apresenta a HAMSTER, uma arquitetura de comunicação de dados baseada em mobilidade e segurança para veículos não tripulados, que aborda três tipos principais de comunicação: máquina-para-máquina, máquina-para-infraestrutura e comunicações internas. Quatro elementos adicionais co-relacionados são fornecidos juntamente com a arquitetura HAMSTER de modo a prover abordagens mais precisas em relação a aspectos de segurança física e da informação (plataforma SPHERE), análise de criticalidade (índice NCI), eficiência energética (plataforma NP) e comunicações ad hoc e infraestruturadas orientadas a mobilidade (plataforma NIMBLE). Além disso, são fornecidas três versões especializadas: para veículos aéreos não tripulados (*Flying* HAMSTER), veículos terrestres não tripulados (*Running* HAMSTER) e veículos submarinos e de superfície não tripulados (*Swimming* HAMSTER). A validação da arquitetura é obtida por meio de estudos de caso sobre cada recurso abordado, levando a diretrizes sobre o desenvolvimento de veículos mais preparados para atender a requisitos de certificação, comunicação mais eficiente e segura, abordagens assertivas sobre criticidade e abordagens verdes nas comunicações internas. Por fim, os resultados comprovaram a eficiência da arquitetura HAMSTER e os elementos com ela providos, bem como a flexibilidade em realizar experimentos focados em vários aspectos de comunicação, auxiliando na obtenção de comunicações seguras em veículos autônomos.

Palavras-chave: Comunicação, Redes de computadores, Arquitetura de comunicação de dados, Eficiência energética, Arquitetura HAMSTER, Verificação de saúde, Índice de criticidade, Segurança física, Segurança da informação.

ABSTRACT

PIGATTO, D. F. **HAMSTER – healthy, mobility and security-based data communication architecture for unmanned systems**. 2017. 201 p. Doctoral dissertation (Doctorate Candidate Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

Advances in communications have been unarguably essential to enable modern systems and applications as we know them. Ubiquity has turned into reality, allowing specialised embedded systems to eminently grow and spread. That is notably the case of unmanned vehicles which have been creatively explored on applications that were not as efficient as they currently are, neither as innovative as recently accomplished. Therefore, towards the efficient operation of either unmanned vehicles and systems they integrate, in addition to communication improvements, it is highly desired that we carefully observe relevant, co-related necessities that may lead to the full insertion of unmanned vehicles to our everyday lives. Moreover, by addressing these demands on integrated solutions, better results will likely be produced. This thesis presents HAMSTER, the HeAlthy, Mobility and Security based data communication archiTEctuRe for unmanned vehicles, which addresses three main types of communications: machine-to-machine, machine-to-infrastructure and internal machine communications. Four additional elements on co-related requirements are provided alongside with HAMSTER for more accurate approaches regarding security and safety aspects (SPHERE platform), criticality analysis (NCI index), energy efficiency (NP platform) and mobility-oriented ad hoc and infrastructured communications (NIMBLE platform). Furthermore, three specialised versions are provided: unmanned aerial vehicles (Flying HAMSTER), unmanned ground vehicles (Running HAMSTER) and unmanned surface/underwater vehicles (Swimming HAMSTER). The architecture validation is achieved by case studies on each feature addressed, leading to guidelines on the development of vehicles more likely to meet certification requirements, more efficient and secure communications, assertive approaches regarding criticality and green approaches on internal communications. Indeed, results prove the efficiency and effectiveness of HAMSTER architecture and its elements, as well as its flexibility in carrying out different experiments focused on various aspects of communication, which helps researchers and developers to achieve safe and secure communications in unmanned vehicles.

Keywords: Communications, Computer networks, Data communication architecture, Energy efficiency, HAMSTER architecture, Health checking, Node Criticality Index, Safety, Security.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

A2A	Aircraft-to-Aircraft Communication
A2I	Aircraft-to-Infrastructure Communication
AODV	Ad Hoc On Demand Distance Vector Routing
AP	Access Point
BER	Bit Error Rate
BSP	Broadcast Storm Problem
CAGE	Control and monitoring AGEncy
CaRINA	Carro Robótico Inteligente para Navegação Autônoma
CDMA	Code Division Multiple Access
CNPC	Control and Non-Payload Communication
COTS	Commercial-Off-The-Shelf
CSU	Central Security Unit
DDoS	Distributed Denial of Service
DoS	Denial of Service
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
FAA	Federal Aviation Administration
FANET	Flying Ad hoc NETwork
GCS	Ground Control Station
GPS	Global Positioning System
GUI	Graphical User Interface
HAMSTER	HeAlthy, Mobility and Security-based data communication archiTEctuRe
IAC	Internal Aircraft Communication
ICSP	In-Circuit Serial Programming
IEEE	Institute of Electrical and Electronics Engineers
IMC	Internal Machine Communication
IMU	Inertial Measurement Unit
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IVC	Internal Vehicle Communication

IWC	Internal Water vehicle Communication
LoS	Line-of-Sight
M2I	Machine-to-Infrastructure Communication
M2M	Machine-to-Machine Communication
MAC	Medium Access Control
MANET	Mobile Ad hoc NETworks
MIAS	Mobile Intelligent Autonomous System
MIPv6	Mobile IPv6
MIRACL	Multiprecision Integer and Rational Arithmetic C/C++ Library
MU	Mobile Unit
MUD	Multi-User Detection
NCI	Node Criticality Index
NCSWT	Networked Control System Wind Tunnel
NIMBLE	NatIve MoBiLity platform for unmanned systEms
NP	Navigation Phases
PCB	Printed Circuit Board
PDA	Photo-Diode Array
QoS	Quality of Service
RELIC	Library for Efficient Cryptography
RPAS	Remotely Piloted Aircraft Systems
RPSMA	Reverse-Polarity Sub-Miniature version A
RSA	Rivest Shamir Adleman
SMU	Safety Management Unit
SNR	Signal to Noise Ratio
SPHERE	Security and safety Platform for HEteRogeneous systEms
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
UANET	Underwater Ad hoc NETwork
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UGS	Unmanned Ground System
UGV	Unmanned Ground Vehicle
UML	Unified Modelling Language
UNICAMP	University of Campinas
USB	Universal Serial Bus
USV	Unmanned Surface Vehicle

UUV	Unmanned Underwater Vehicle
UV	Unmanned Vehicle
UWS	Unmanned Water System
V2I	Vehicle-to-Infrastructure Communication
V2V	Vehicle-to-Vehicle Communication
VANET	Vehicular Ad hoc NETwork
W2I	Water vehicle-to-Infrastructure Communication
W2W	Water vehicle-to-Water vehicle Communication
WAN	Wireless Avionics Network
WiMAX	Worldwide Interoperability for Microwave Access

CONTENTS

CHAPTER 1

INTRODUCTION

A brief definition for Unmanned Vehicle (UV) is a vehicle without a person on board. Also seen as uncrewed vehicles, they can be remotely or autonomously controlled, usually applied in a wide range of environmental sensing activities, high risk areas monitoring, driving assistance, monitoring activities and much more. There is massive research interest on this field as smart drones, cars and boats take place, get cheaper, easier to control and even more integrated to everyday situations. However, a major concern on these vehicles' acceptance and certification still relies on safety and security issues, apart from other requirements that must be met as new vehicles are developed.

The particular case of Unmanned Aerial Vehicles (UAVs) was the initial motivation for the development of this thesis due to inherent safety challenges that are closely related to reliable communications. The growing popularisation of UAVs has boosted research in this field and is fostering the use of such technology in many new applications, such as recent integration of UAVs to the Internet of Things. There are periodically published roadmaps written by military and civilian organisations – e.g. United States Army (USArmy), American Department of Defense, European RPAS (Remotely Piloted Aircraft Systems) Steering Group and Federal Aviation Administration (FAA) – that outline the expected advances for UAVs (US Army, 2010; YEARBOOK, 2011; UK Civil Aviation Authority, 2012; DOD, 2013). In summary, there are five challenges of Unmanned Aircraft Systems (UAS) integration into the airspace as stated by Dr. Wilson Felder, the Director of the William J. Hughes Technical Centre of the FAA, reported in Stark, Stevenson and Chen (2013): procedural, technical, aircraft safety, crew credentials and public acceptance. Any person or computer-based system that meets the three mandatory activities to operate an aircraft (flight, navigation and communication) should assume the command of an aircraft, be it manned or unmanned (Baraldi Sesso *et al.*, 2016).

Safety and security topics are underestimated when it comes to critical embedded systems. Instead of being considered as features for later development, these topics must be taken into account as high priority requirements from project design to the system final deployment. Besides that, distinct criticality levels are important characteristics of critical systems that have not been widely addressed by recent researches. Moreover, a challenging trade-off between the provision of such requirements with the necessity for green approaches must be accurately investigated. These areas are closely related to communications and influence each other. They are also very relevant for certification purposes, thus demand assertive efforts towards a global improvement. In regard to communications, there are specific aspects that still demand research efforts and are the focus of this thesis.

In this thesis, the **H**e**A**lthy, **M**obility and **S**ecurity-based data communication archi**TE**ctu**R**e (HAMSTER) is introduced. HAMSTER is a data communication architecture for unmanned vehicles which improves mobility, security and safety of the overall system.

1.1 Motivation and problem statement

In short, FAA requirements demand that UAS meet safety levels equivalent to those of manned aircraft. It includes the frequency of collision of a UAS being operated in a FAA-controlled airspace, which is currently 10^{-7} events per hour of operation for manned aircraft (ASMAT *et al.*, 2006). Moreover, in response to merged approaches that may include both manned and unmanned aircraft at the same time, leading to the necessity of using the non-segregated airspace, new means of meeting safety requirements are needed. Therefore, there is a demand for communication architectures that lead to improvements on overall UAV safety, including both the aircraft itself and the population, helping the UAS to address some of the current main challenges.

In the last few years there has been a growing interest in approaches with multiple heterogeneous vehicles, integrating UAVs with ground and aquatic vehicles for better environment sensing and mission performing. Search and rescue scenarios are good examples on how UAVs can work with Unmanned Ground Vehicles (UGV) and cooperate for more precise results, leading to the necessity of communication architectures that not only provide means of increasing safety, but also make it easier to safely and securely communicate different vehicles.

The aim of this thesis is the specification of a data communication architecture for unmanned vehicles designed with the integration of exclusive platforms for mobility, energy efficiency, criticality determination and assurance of security and safety. Although major challenges appear in the aerial scenario, three types of vehicles are taken as main focus: aerial, ground and aquatic vehicles. However, it is important to point out that the architecture is not limited to these three vehicles, being open for further developments and new vehicles.

1.2 Research question

This thesis' main research question is: **how to enable heterogeneous unmanned vehicles to communicate with improved system safety, information security and reduced internal energy consumption?**

The answer to the main research question can be achieved by answering several related smaller research questions:

1. How to identify potential sensors and actuators that might be safety threats for the entire system and take anticipated appropriate action?
2. How to measure the importance of a single node on the system and make this information available for improving communication, security, safety, mobility and tasks delegation?
3. How to internally save energy during an unmanned vehicle operation by analysing navigation patterns?
4. How to separately handle communications among vehicles and communications with infrastructure elements?
5. How to provide communication standards with safety and security requirements for heterogeneous vehicles?

1.3 Hypothesis

The hypothesis of solving the mentioned gap is the definition of a data communication architecture with well-defined ways of communication security, improved safety management including modules health checking, the possibility of reducing energy consumption specially on wireless communications, mobility-specific platform that independently manages ad hoc and infrastructured communications, criticality analysis for more precise tasks delegation and accurate approaches on aforementioned features.

1.4 Objectives

The main objective of this thesis is to help improving the reliability of UVs through the specification of a data communication architecture aimed at providing heterogeneous unmanned vehicles with secure communication links, safety management, targeted mobility approaches, criticality identification and internal energy saving. The achievement of this main objective is a result of the integration of independent platforms that cooperatively work towards the goals of modern unmanned vehicles.

Specific objectives were also defined, as follows:

1. Define an independent platform to deal with security and safety in both integrated and isolated ways, taking care of UVs' major necessities, specially the ones that are more relevant for certification, but also open for individual needs that might emerge;

2. Identify and formalise general operation phases that UVs will potentially assume throughout missions, helping the achievement of a better energy efficiency by turning off idle nodes and also limiting bandwidth if possible;

3. Provide a formal calculation of criticality for sensors, actuators, vehicles and support systems that may be connected to an unmanned system, allowing developers and researches to develop more precise approaches for a wide variety of purposes by considering a unified criticality index; and

4. Obtain a segmented platform to deal with mobility that takes into consideration the differences between ad hoc and infrastructure-based communications.

1.5 Contributions

The originality of this thesis is the definition of HAMSTER, a data communication architecture that provides an integrated reference model to address one of the main issues faced by unmanned vehicles. However, its design still took into account modularisation, which resulted on the definition of independent platforms to manage the main aspects.

Moreover, this thesis' contributions go towards the requirements of modern unmanned vehicles applications. There are natural limitations in missions performed by a single vehicle either by the restrict set of functions it can execute and the necessity of flexible approaches that imitate and go beyond human capacities. The data communication architecture specified in this thesis helps integrating heterogeneous vehicles into a unique system, keeping high levels of security and safety.

Alongside with the architecture, other relevant contributions and results were provided:

- **HAMSTER architecture**. HAMSTER is the main contribution provided by this thesis. It specifies well defined ways of achieving communication goals through a reference model to assist the development of safety, security, mobility-based, energy-efficient unmanned systems in Unified Modelling Language (UML), openly available for further research and development. (PIGATTO, 2013; PIGATTO *et al.*, 2014; PIGATTO *et al.*, 2016).

- **HAMSTER unit**. This unit turns modules into HAMSTER-ready elements, implementing all the platforms and features provided with the architecture on a well-defined way. The main contribution is the abstraction of physical objects when it comes to communications.

- **Security and safety Platform for HEteRogeneous systEms (SPHERE)**. The need for vehicles aligned to certification requirements has recently increased with the introduction of applications demanding their inherent flexibility. SPHERE provides specialised modules to deal with safety and security requirements both on integrated and independent approaches. This is one of the main contributions which directly meets communication-related requirements of certification processes (PIGATTO *et al.*, 2015; SILVA *et al.*, 2015).

- **Node Criticality Index (NCI)**. This contribution is the specification of a formal criticality classification for network nodes in various levels. The estimated score takes into account modules health, UV cost, manipulated and stored data, mission requirements, field of operation and the importance of fully accomplishing a mission. This set of information contributes with the provision of relevant data for the development of communication protocols, tasks delegation management units and improved system safety and information security (PIGATTO *et al.*, 2016).

- **Navigation Phases (NP)**. This approach contributes towards energy efficiency by using the knowledge on unmanned vehicles' operation phases. NP classifies known operation stages and attributes very specific behaviours that may reduce energy consumption (PIGATTO *et al.*, 2015; PIGATTO *et al.*, 2016).

- **NatIve MoBiLity platform for unmanned systEms (NIMBLE)**. External communications may include different requirements regarding mobility and operation modes. Aiming at individually addressing issues, NIMBLE manages external communications with individual modules permitting requirements-oriented developments towards ad hoc and infrastructured networks improvements (MUNHOZ *et al.*, 2016; MARCONATO *et al.*, 2016; MARCONATO *et al.*, 2017).

1.6 Text organisation

The remainder of this thesis is organised as follows: a review on communications in unmanned vehicles is carried out in Chapter 2; the proposed architecture is discussed in Chapter 3; the validation is presented in several case studies, starting with security and safety aspects in Chapter 4; a node criticality index is empirically analysed in Chapter 5; approaches towards energy efficiency are provided in Chapter 6; external communications are addressed in Chapter 7; and, finally, the conclusion is reported in Chapter 8.

CHAPTER

2

COMMUNICATIONS IN UNMANNED VEHICLES AND SYSTEMS

2.1 Chapter overview

Embedded systems are a combination of hardware and software designed to perform a specific function, usually as part of a larger system (BARR, 1999). Some of these systems are considered critical due to the fact that a malfunction or failure may result in high monetary losses and human risks (JANUZAJ *et al.*, 2010). Examples of embedded systems are found in many applications, including unmanned systems, such as UAVs, UGVs and unmanned surface/underwater vehicles (USVs and UUVs).

Currently, a majority of embedded systems are equipped with wireless adaptor interfaces that enable flexible network reconfigurations and allow these systems to be as mobile as possible. Mobility is a key concept that contributed to embedded systems dissemination, being applied in scenarios where accessibility used to be formerly difficult or even impossible, enabling a wide range of new applications e.g. forest fires detection, volcanic activity monitoring, crops diseases identification and more (YICK; MUKHERJEE; GHOSAL, 2008).

Although embedded systems have operated at limited power, processing and memory resources for many years, recent advances are changing this scenario (SINGH *et al.*, 2013). There are currently both smaller embedded systems that still operate with low-capacity resources in very specific applications (e.g. sensing, automation, processing, communication, etc.) (COLOMINA; MOLINA, 2014) and modern embedded systems ready for advanced applications (e.g. unmanned systems). Despite the fact that limitations have decreased, there are aspects that may still affect the development of embedded systems, such as components cost, demand for green approaches, performance and quality of service guarantees (MOZAFFARI *et al.*, 2016). Moreover, the provision of security and safety becomes challenging as embedded system are applied on critical applications, demanding a rational use of resources. Traditionally, security had not been considered a priority requirement on embedded systems design (KOCHER *et al.*, 2004; RAVI *et al.*, 2004) and the immaturity of security approaches in this domain has been emphasised by recent literature (STUDNIA *et al.*, 2013; GASHI *et al.*, 2014; GREEN; ÇIÇEK; KOÇ, 2016).

Highly connected environments led to a wider exposure to malicious attacks, raising new safety and security concerns e.g. unauthorised entities might be able to invade systems, steal information, make services unavailable and physically or logically damage devices (JAVAID *et al.*, 2012). That has also increased challenges on the design of new communication architectures for embedded systems. In regards of unmanned vehicles, because of their inherent criticality, the design of safety and security is a very important effort that must be devoted.

2.2 Unmanned systems

An unmanned system is a machine or device equipped with necessary data processing units, sensors, automatic control and communications mechanisms that is capable of performing missions autonomously without human intervention (World Scientific, 2017). Unmanned systems include unmanned aircraft (popularly known as drones), ground robots, underwater explorers, satellites and other unconventional structures. Raol and Gopal (2012) named these systems as Mobile Intelligent Autonomous Systems (MIAS), which are meant to comprise theory and practice of several closely related technologies that have some elements of mobility, intelligence and/or autonomy operating and are envisaged not only for robots, but also for other mobile vehicles.

Their applications depend on the type of vehicle. UAVs applications are normally related to precision agriculture, environmental monitoring, safety, military and civil defence (MAZA *et al.*, 2011; LUO *et al.*, 2012; VERMA; FERNANDES, 2013; BOUACHIR *et al.*, 2014). Unmanned ground vehicles are seen in driving assistance, accidents prevention, precision agriculture and industrial applications (SUN *et al.*, 2011; FERNANDES *et al.*, 2014). Unmanned aquatic vehicles have been used for tasks related to oil exploration, hydropower maintenance and marine geoscience (WYNN *et al.*, 2014). Current challenges for these systems include multiple heterogeneous vehicles missions (XIANG *et al.*, 2012; ABBOTT-MCCUNE *et al.*, 2013).

Communication is one of the biggest challenges in designing systems with multiple vehicles and also a crucial aspect for cooperation and collaboration (CHUNG *et al.*, 2011a; BOUACHIR *et al.*, 2014). There are three main types of communications in the context of unmanned vehicles systems: (a) internal machine communications (IMC); (b) machine-to-machine communications (M2M); and (c) machine-to-infrastructure communications

(M2I).

IMC can be wired or wireless. M2I communication comprises the links between the vehicles and the network infrastructure, which can be either a direct link to a central control station or mediated by a satellite (FREW; BROWN, 2008). M2M communication includes the communication among all vehicles. In an infrastructure-based unmanned system, M2M can be performed through such infrastructure. However, this requires a more expensive and complex hardware in all vehicles to enable communication with the control station or satellite. Furthermore, the reliability of communication is also compromised since factors, such as changing environmental conditions, movements by aerial, aquatic or ground vehicles and different characteristics of terrain relief or obstacles, interfere with the ability of vehicles in maintaining the communication link.

An important issue to be addressed is the range restriction between vehicle and control station. If a vehicle is outside the coverage area of the control station, it is consequently disconnected from the network. An alternative communication solution for unmanned systems with multiple vehicles is the use of ad hoc networks to connect vehicles, also known as VANETs (Vehicular Ad hoc NETworks) (BOVEE *et al.*, 2011; RAWAT *et al.*, 2013) and FANETs (Flying Ad hoc NETworks) (TEMEL; BEKMEZCI, 2013; SAHINGOZ, 2014). As long as part of the vehicles are within the range of the control station or satellite, all vehicles constitute an ad hoc network, which enables every vehicle to communicate with the control station.

Next section will present the state of the art in communication architectures for unmanned systems.

2.3 State of the art on data communication architectures for unmanned vehicles

The main goal of this section is to identify and characterise current data communication architectures for unmanned aerial, aquatic and ground vehicles. The systematic review technique was chosen as the methodology. Researches which introduced novelties that could be eventually incorporated to data communication architectures were also considered. Following subsections will present the state of the art in each communication type.

2.3.1 Internal Machine Communications

Internal communication is of fundamental importance in some studies as shown by Sun *et al.* (2011). Their proposal aims to use UAVs to collect information derived from sensors positioned over and underground for border patrol. To facilitate the detection of objects in a timely and accurate manner, effective communication protocols are needed to support mainly two types of transmission: i) sending information on suspicious activities detected by ground sensors to surveillance towers and ii) sending images captured by watchtowers to remote control centres. The challenge in this scenario is the communication many-to-many, not only externally, but mainly internally for communication between data collectors from terrestrial sensors.

Also in the context of ground sensors information collection, Tan and Munro (2007) considered an urban setting with obstacles that affect communication among ground sensors and a UAV flying at about 100 m of altitude. The authors propose the application of Adaptive Probabilistic Epidemic Protocol protocol to enable nodes to send information via broadcast based on the quantity of neighbours, avoiding doing so in cases when many neighbouring nodes are also transmitting, aiming at reducing the level of collisions and increasing communication performance. The protocol considers information, such as Line-of-Sight (LoS), associated with the number of neighbouring nodes. According to the authors, this issue is also seen in internal communication.

The size of the aircraft may also affect internal communications. Mohamed *et al.* (2013) investigated the collaborative aspects and challenges of multiple UAV systems and points out that one of the main issues for these systems is the demand for an effective framework to enable the development of software systems for collaborative UAV operations. One possible approach is to rely on service-oriented computing and service-oriented middleware technologies to simplify development and operations. The paper discusses how the service-oriented middleware approach can help solving some of the challenges of the development of collaborative UAVs and also proposes a service-oriented middleware architecture that can meet requirements of the development and operations of such applications.

On the other hand, Frew and Brown (2008) surveyed the main network requirements for small UAS. They deal with smaller physical integrity concerns, since some techniques may be used in failure situations, such as the use of a parachute. However, these aircraft are also very limited in payload, or carry only on-board sensors and are therefore more limited. There are three operational requirements considered by authors. The first one is the connectivity, which considers that the aircraft may be "connected" or "disconnected" under stress. The second one is the delivery of data, which can be "reliable" (with short delays) or "unreliable" (with long delays). And the third one is the discovery of service that can be "stable" or "unstable", depending on the context and the circumstances. All requirements are very well established in traditional, static and wired networks, but in the context of UAVs they turn into major challenges. These problems apply to internal and external communications in UAS.

According to Dang *et al.* (2012), wireless technology on avionics internal communications has become feasible but also advisable for the following reasons: first, a Wireless Avionics Network (WAN) will allow an inherent weight reduction and an increase of system's flexibility and efficiency through less fuel consumption and better flight autonomy; second, eliminating the wiring-ageing-related problems shall enhance the system scalability and safety thanks to simpler fault allocation process and less fire hazards; third, cable-less avionics implementation will inherently reduce the costs not only during design, production and development process but also for maintenance and overhaul.

Dang *et al.* (2012) also pointed out the disadvantages related to fly by wireless paradigm. There is a new trend to use commercial-off-the-shelf (COTS) technology rather than designing a dedicated solution to reduce the development costs. However, the problem with COTS is reconciling the different requirements between commercial and safety-critical applications. For wireless technologies, the main concerns are related to the system's susceptibility against electromagnetic interferences. This is mainly due to natural phenomena or man-made events that could be internal or external to the plane, e.g. Portable Electronic Devices, satellite communications or Radio Navigation. This results in both data rate and Quality of Service (QoS) degradation or even network collapse. Furthermore, there is system security issue due to the access and manipulation of sent information with Man-in-the-Middle and Denial of Service (DoS) attacks (GOMEZ, 2010).

In general, studies that address IMC communications deal with problems in which several sensors or modules constantly exchange information among themselves and with information centralisers. In some cases, the approaches are different concerning critical parts of the vehicle and may also vary according to the size of the aircraft. There is no data communication architecture in the literature which would help defining the most appropriate characteristics of the IMC communications in vehicles. Such an architecture must address the requirements of wired and wireless communications, dealing with the specific characteristics of each scenario.

2.3.2 Machine-to-Machine Communications

One of the main requirements of M2M communication is ensuring connectivity even with the UAVs operating at high speeds. Kuiper and Nadjm-Tehrani (2006) simulated a reconnaissance mission where some UAVs must fully scan an area of 30 km. Such checking should be done at least once per hour. Some requirements are established, e.g. maintain a constant connection with the control station and do not extrapolate the use of communication bandwidth to exchange routing or mobility information, since it is a scarce resource. Two mobility models were proposed based on these requirements: a simple random pattern and a model in which the movement of a UAV is highly dependent on the mobility of other UAVs. The simulation results demonstrate that the random model scanned nearly 80% of the area in 2 hours. The model of distributed pheromone, in turn, could scan 90% of the area in only 1 hour, proving to be a better approach.

In case of failures, the connectivity must be guaranteed plotting alternative paths of communication among UAVs. Shirazipourazad, Ghosh and Sen (2011) proposed a robust Airborne Network where, in case of failures the surviving nodes of the network can remain connected and able to communicate with other node. The algorithm searches for the shortest distance of transmission needed to ensure network connectivity regardless of failure location.

Regions of hampered access often have more communication failures due to lack of ground support to infrastructured communication. Thus, specific approaches must be developed as shown by Verma and Fernandes (2013). They proposed techniques for setting or retrieving a UAV communication network considering mission requirements in areas where it is not trivial to maintain a terrestrial network to support the aerial network. They suggest a connectivity model where a UAV attempts to connect to the four nearest UAVs, thereby obtaining full connectivity among all UAVs in a predetermined space. Simulations have also shown the possibility of establishing a connection among terrestrial and aerial network elements.

A way of improving connectivity among aircraft is by analysing the optimal placement of antennas. Some works presented efforts towards this direction, which is the case of Temel and Bekmezci (2013). One of the current problems in the area is that most of the publications on MANETs (Mobile Ad hoc NETworks) and VANETs barely explore methods for antenna positioning. In the context of FANETs, high speeds and altitudes require new approaches, as the one proposed by the authors which specified a new Medium Access Control (MAC) protocol, Location Oriented Directional Medium Access Control. It presents changes to the use of directional antennas to provide all UAVs with the exact location information of neighbours with frequent updates of GPS (Global Positioning System) through the assistance of a High Altitude Platform and a vector of three associated antennas.

Bettstetter, Hartmann and Moser (2005) made the assumption that all transmitting directional antennas are oriented in a random direction and do not attempt any adaptation or control. It consists on a reasonable assumption for ad hoc networks with hardware-limited and power-constrained wireless devices.

Alshbatat and Dong (2010) proposed the use of directional antennas embedded in UAVs connected to FANETs. They suggested two new schemes: Intelligent Medium Access Control Protocol for UAV and Directional Optimised Link State Routing. After that, a comparison with IEEE 802.11 and Optimised Link State Routing protocols was carried out, respectively. Results show that both proposed schemes performed better than the well-known protocols used for comparison. These results were simulated using OPNET (Optimised Network Engineering Tools) (RIVERBED, 2014) and showed the importance of establishing new protocols to improve the performance of FANETs.

In addition, the positioning of UAVs into the airspace is addressed by Severinghaus, Tummala and McEachen (2013), merging air and ground scenarios. They showed possible improvements in the coverage of wireless communication between ground vehicles with better placement of antennas. The authors have considered the possibility of integrating a UAV to serve as a central node to the terrestrial network, enabling LoS to improve communication. Related to that, Goddemeier *et al.* (2011) presented an approach based on potential fields to solve the problem of UAVs' air positioning to cover the largest possible ground area, without the existence of "dead" regions. In this situation, LoS is a crucial factor. Towards this direction, approaches like the one presented by Tan and Munro (2007) and discussed in the previous subsection may help to address such issues.

The optimisation of the aircraft positioning as a communication gateway was addressed by Quaritsch *et al.* (2010). The study focused on natural disaster management applications using sensor networks enabled by UAVs. The goal was finding the best UAV position in an area and shooting the largest area with the smallest possible number of captures. Moreover, authors have also attempted to reduce energy consumption, flight time and bandwidth usage.

Some decision making may be performed offline with no need to intermittent communication with the control station. Luo *et al.* (2012) presented an aircraft which collects information from other aircraft performing a mission and forwards to the control station. This approach can be used when there is no requirement for real-time mission processing.

In cases of flight formation, links with the ground station must be constant. Thus, Li *et al.* (2013) proposed a token to detect hidden/lost neighbours, code assignment and cooperative transmission in UAV ad hoc networks based on CDMA (Code Division Multiple Access) with multi-user detection (MUD). Tests presented have shown that the solution with MUD has an average packet delay slightly smaller in relation to the implementation without MUD. Regarding the percentage of packets delivered, the solution with MUD also performed better. On the other hand, Webber and Hiromoto (2006) presented UAVs organised in clusters and controlled by a single leader. This main UAV sends information about the mission to each UAV; after that, each UAV calculates its own trajectories, positions and corrections needed to keep the flight formation.

Regarding connectivity among aircraft, Pandey and Verma (2011) evaluated the performance of AODV (Ad Hoc On Demand Distance Vector) routing protocol concerning network topology changes and different mobility conditions. It was observed that, throughout the simulation, the Jitter value increased linearly. For different mobility conditions, the behaviour of Jitter value did not present linear behaviour. A hundred nodes were used to represent unmanned vehicles and only five nodes acting as ground control stations.

Experiments where the UVs and ground stations were stationary presented the lowest Jitter values. On the contrary, with mobile nodes, the Jitter value reached its highest value.

A discussion on UAVs motion was provided by Riley *et al.* (2011). They have presented the Networked Control System Wind Tunnel (NCSWT), Networked Control Systems integrated simulation environment. The experiments were divided into case studies to address communications among UAVs and were simulated on NS-2 and MATLAB/Simulink simulators. Results proved that NCSWT can achieve times within or below the expected.

Regarding mobility models, Bouachir *et al.* (2014) applied a realistic environment to predict communication problems that may affect UAS performance. As reported, the node mobility has a great effect on network topology and communication protocol performance. They have also presented a realistic mobility model designed for UAV ad hoc networks based on Paparazzi UAVs motion patterns. Indeed, results proved that the mobility prediction is similar to real UAV traces.

Sahingoz (2014) identified main challenges of using UAVs as relay nodes in an ad hoc networks, introduced UAV network models and depicted open research issues. Similarly, Gupta, Jain and Vaszkun (2016) have carried out a complete survey on outstanding issues of UAV networks. Protocols are required to adapt to high mobility, dynamic topology, intermittent links, power constraints and changing link quality. More recently, Jawhar *et al.* (2017) have discussed communication and networking of UAV-based systems by stating possible architectures, however not mentioning the highly needed security and safety requirements.

Researches in regards of unmanned maritime vehicles communications have been published in recent years. Karthik (2014) has used an underwater vehicle for surveillance swarm network communication. Murad *et al.* (2014), in turn, carried out a survey on current underwater acoustic sensor network applications. And Verma and Prachi (2015) discussed various communication methodologies to determine which one suits best to the requirements of Underwater Wireless Sensor Networks.

Once there is an increasing need to assure safer operations, more secure communications, improved mobility and energy efficiency on unmanned vehicles, Pigatto *et al.* (2014), Pigatto *et al.* (2015), Pigatto *et al.* (2016) have published advances with HAMSTER data communication architecture. The most solid version of HAMSTER is presented in this thesis, supported by case studies that validate each inherent platform.

In short, there are many papers approaching M2M communications addressing a variety of issues introduced by complex and sensitive environments. There are several possibilities once a vehicle is in the air, on the ground or over/under water. All variations determine specific and critical requirements that must be thoroughly addressed.

2.3.3 Machine-to-Infrastructure Communications

As previously mentioned, some authors proposed the placement of UAVs as air relay points for terrestrial MANETs. Cemin, Gotz and Pereira (2012) presented a method to optimise the placement of a UAV platform, attempting to obtain the widest possible coverage where a ground-based MANET operates. The accurate positioning of a relay UAV helps increasing the robustness of the system, broadening the average flow of data and reducing the incidence of delays in communication. The proposed algorithm is based on an iterative procedure to compute the best UAV position in order to obtain the best possible situation from communication perspective. Rubin and Zhang (2007) presented an optimisation model for selecting the best location of a UAV over a predetermined region to serve as a relay for ground located nodes, presenting results which proved that the best location is chosen. That includes altitude positioning (the higher the altitude, the greater the coverage area, however the lower the signal strength) and the horizontal positioning to provide better communication to nodes scattered on the ground.

Altitude is a factor that impacts the communication between an aircraft and the ground control station. Hatziefremidis *et al.* (2013) analysed the communication Bit Error Rate (BER) and Signal to Noise Ratio (SNR) between a UAV and a control station at different conditions, e.g. aircraft altitude, visibility, wavelength of communication link and atmospheric conditions. The simulation was performed using MATLAB and results showed that, depending on factors like altitude and visibility, the relative loss caused by atmospheric factors can be determined.

Lee *et al.* (2010) applied Mobile WiMAX (Worldwide Interoperability for Microwave Access) on ground vehicles and ground control stations communication for teleoperation means. It was a review on performance and viability of the technology in a specific scenario, but results refer only to tests with LoS. On the other hand, Durham *et al.* (2009) simulated the physical layer with the discrete event simulator OPNET (RIVERBED, 2014). The OPNET Modeller was selected for experiments due to the large number of different networks ready to use. For the paper simulations, a network with a UAV and a ground control station was created and the OPNET Antenna Editor module was used for modelling transmission/reception antennas.

Papers focused on M2I communications address some particular situations, despite having similar problems to M2M communications. For instance, altitude and large distances are cited as a major factor influencing the performance of communications with infrastructures, as well as the lack LoS and aircraft speeds. As one addresses M2I communications specifically for each type of vehicle, there will be divergent issues arising. UAVs might face challenges due to long distances. UGVs, in turn, have to deal with obstacles. On the other hand, aquatic vehicles will face two different environments that impose distinct challenges: surface vehicles communicate through the air and underwater vehicles communicate through the water, which is even more arduous. To summarise, M2I communications are less critical than M2M communications in most of the cases. However, there are important challenges to be addressed and further explored.

2.3.4 Safety and security

Chien and Lin (2006) proposed a security framework for MANETs with hierarchical structural organisation. The framework has an easy adjustment considering that these networks do not count on a fixed organisation. The proposal aims to use low computational power and does not require key exchange.

Javaid *et al.* (2012) proposed an assessment of security of M2M and M2I communications including vehicles, ground stations and satellites. The authors mentioned that the security solutions applied to wireless sensor networks and MANETs are not appropriate for aircraft networks due to some substantial differences and presented a modelling of major attacks that can be experienced by an aircraft. Furthermore, they assigned severity levels to such attacks and calculations are performed to identify the priority with which they should treat each attack.

Similarly, Puchaty and Delaurentis (2011) presented a study regarding attacks to UAV-based sensor networks. A military scenario composed by UAVs, ground control stations and satellites was tested through a 30-minutes simulation. The environment contained a centre of mission control, three UAVs, injection attacks and trajectories defined by waypoints. Among the cited attacks there was a DDoS (Distributed Denial of Service) attack. The simulation results showed degradations in communication, such as increased latency when under attack. The bigger the number of redundant links, the better the system availability.

Faughnan *et al.* (2013) investigated UAVs kidnapping. The method is divided into two parts. The first one was the risk identification of a UAV attack. To perform this part, a list of risk scenarios was created. The second one consisted on a mechanism to inform the system operator that the UAV was possibly under attack.

Bakar *et al.* (2009) developed secure channels for communication among UAV systems, satellites and base stations and addressed major attacks in UAVs. Initially, the main components of the system were identified, based on a criticality degree. Then, a system model was created and associated to attacks and threats. The simulated results permitted to analyse the behaviour of the network under attack. After a series of attacks, the system had some failing components, especially after the denial of service attack.

Man *et al.* (2009) proposed a way of monitoring health and safety of UAV systems. First, a model with the main components of an aircraft was designed. These components were grouped according to their function within the system, allowing a faster module identification in case of errors. In addition, the authors discussed some techniques to predict when modules should begin to be defective based on the quality of data and experience regarding the use of UAVs.

Raj, SelvaKumar and Lekha (2011) presented a protocol for the authentication of nodes to join a network. To be part of the network, a node must request and be granted with permission from all other nodes in the network using a secure channel of communication.

Iannicca *et al.* (2013) presented a survey on major security requirements (confidentiality, integrity and availability) focused on CNPC system (Control and Non-Payload Communication). This system is used in the United States for communication among control stations and UAVs, being very well accepted for the integration of UAVs into the airspace. By stating the main requirements and reviewing vulnerabilities related to CNPC, a security modelling was presented, determining risks and development recommendations. A UGV vulnerability assessment is also presented by Abbott-Mccune *et al.* (2013).

The safety challenges related to UAVs certification are addressed by Gimenes *et al.* (2013). They proposed guidelines that could support UAS regulations for the future integration into the Global Air Traffic Management System. These guidelines are based on three viewpoints: the aircraft, the piloting autonomous system and the integration of autonomous UAS into non-segregated airspace. It should help UAVs to safely share airspace with manned aircraft.

Papers related to safety and security are focused on specific details of unmanned systems. Although relevant, solutions that cover every aspect of security and safety communications must be the main focus.

2.4 Final remarks

This chapter summarised the most relevant publications in the field of data communication architectures for unmanned vehicles focusing mainly on aerial, aquatic and ground vehicles. To the best of our knowledge, there are no architectures that fully addresses the requirements stated as this thesis' objectives. However, important progress has been seen in the literature that must be considered by data communication architectures for unmanned vehicles, since they are eligible to assist researchers and developers on the implementation of more suitable unmanned systems with certification requirements fully or partially met.

Exclusive approaches must be developed for each type of communication and vehicle. It leads to the need of general approaches to M2M, M2I and IMC communications, but also special targeted studies towards the unique needs of each vehicle. Such diversity of requirements and impacting factors reveal the necessity for an integrated data communication architecture for heterogeneous vehicles and systems to provide means of addressing these issues as accurately as possible.

Then, next chapter presents HAMSTER, a new data communication architecture for unmanned vehicles.

CHAPTER

3

HAMSTER ARCHITECTURE

This chapter introduces a new data communication architecture for unmanned vehicles which is aimed at improving mobility, security and safety for the overall system, named HAMSTER, an acronym for **HeAlthy, Mobility and Security-based data communication archiTEctuRe**. Although it is focused on three main types of vehicles (aquatic, aerial and terrestrial), HAMSTER architecture can also be adapted for current and upcoming unmanned and autonomous vehicles. Parts of this chapter were independently published in Pigatto *et al.* (2014), Pigatto *et al.* (2015), Pigatto *et al.* (2016).

A brief overview on terminology used from now on will be presented in Section 3.1; the complete specification of a new data communication architecture for unmanned systems is on Section 3.2; after that, the HAMSTER units are presented on Section 3.3, making it clear how the architecture can be implemented; following sections will go through the major platforms found on HAMSTER architecture: the platform for security and safety on Section 3.4; the platform for criticality known as node criticality index on Section 3.5; the platform which improves energy usage named navigation phases on Section 3.6; and the mobility platform on Section 3.7.

3.1 Terminology

This section briefly describes the important terminologies used to describe HAMSTER architecture on a general view. Specific nomenclatures will still be used throughout the text to distinguish particular characteristics or implementations as needed.

3.1.1 Vehicles and systems

This thesis will mostly refer to unmanned vehicles (**UV**) in general, but it can also cite specialised vehicles, such as aerial, ground and aquatic vehicles. Moreover, a group of elements that support the vehicle operation will be referred as a system. For that, the main nomenclatures used are: **UAV** (Unmanned Aerial Vehicle) which is part of a **UAS** (Unmanned Aircraft System); **UGV** (Unmanned Ground Vehicle) which is part of **UGS** (Unmanned Ground System); and, finally, **USV** (Unmanned Surface Vehicle) and **UUV** (Unmanned Underwater Vehicle) which in turn are both part of **UWS** (Unmanned Water System).

3.1.2 Communications

There are three main types of communications covered by HAMSTER, which will assume different names and approaches for each vehicle. In general, this thesis refers to them as: **IMC** which stands for Internal Machine Communication; **M2M** which stands for Machine-to-Machine communication; and **M2I** which stands for Machine-to-Infrastructure communication. Additionally, a single reference to M2M and M2I communications can be seen as **M2X**.

Specialised names for communications will be seen as: **IAC** (Internal Aircraft Communication), **A2A** (Aircraft-to-Aircraft communication) and **A2I** (Aircraft-to-Infrastructure communication) for the aerial segment; **IVC** (Internal Vehicle Communication), **V2V** (Vehicle-to-Vehicle communication) and **V2I** (Vehicle-to-Infrastructure communication) for the ground segment; **IWC** (Internal Water vehicle Communication), **W2W** (Water vehicle-to-Water vehicle communication) and **W2I** (Water vehicle-to-Infrastructure communication) for aquatic segment.

3.1.3 Elements

On the inner view of an unmanned vehicle, HAMSTER architecture will manipulate two types of elements: modules and clusters of modules. A **HAMSTER module** consists on a sensor, actuator, or any other module connected inside the unmanned vehicle, including the payload ones, (e.g. cameras, mission-specific sensors). A group of modules can be manipulated all at once by HAMSTER architecture organised as a cluster of modules. A **HAMSTER cluster of modules** consists on a cluster of sensors, actuators, or any other module inside the unmanned vehicle that share similar characteristics or have related functions.

On the external view, HAMSTER manipulates vehicles and support systems as entities. A **HAMSTER entity** consists on an element belonging to an unmanned system which is connected to a network via M2M or M2I (e.g. UAV, car, control station).

3.1.4 Platforms

HAMSTER architecture is organised in specialised platforms for dedicated purposes. The platform for security and safety on HAMSTER architecture is called **SPHERE**, an acronym that stands for Security and safety Platform for HEteRogeneous systEms[1]; mobility aspects are managed under the **NIMBLE** platform, the NatIve MoBiLity platform for unmanned systEms; **NP** stands for Navigation Phases; **NCI** is the Node Criticality Index; and, finally, a special element is envisaged as a platform for ground or control stations purposes, named **CAGE**, an acronym for Control and monitoring AGEncy.

3.2 The architecture

The **HeAlthy, Mobility and Security-based data communication archiTEctuRe** is divided into three main versions according to the most common types of UV: aerial (Flying HAMSTER), ground (Running HAMSTER) and aquatic (Swimming HAMSTER). Moreover, four special elements are defined: i) a platform intended to control security and safety aspects under all architecture versions; ii) an index to evaluate node criticality within a network; iii) a platform that aims at mobility aspects; and iv) a platform for the provision of efforts towards energy efficiency.

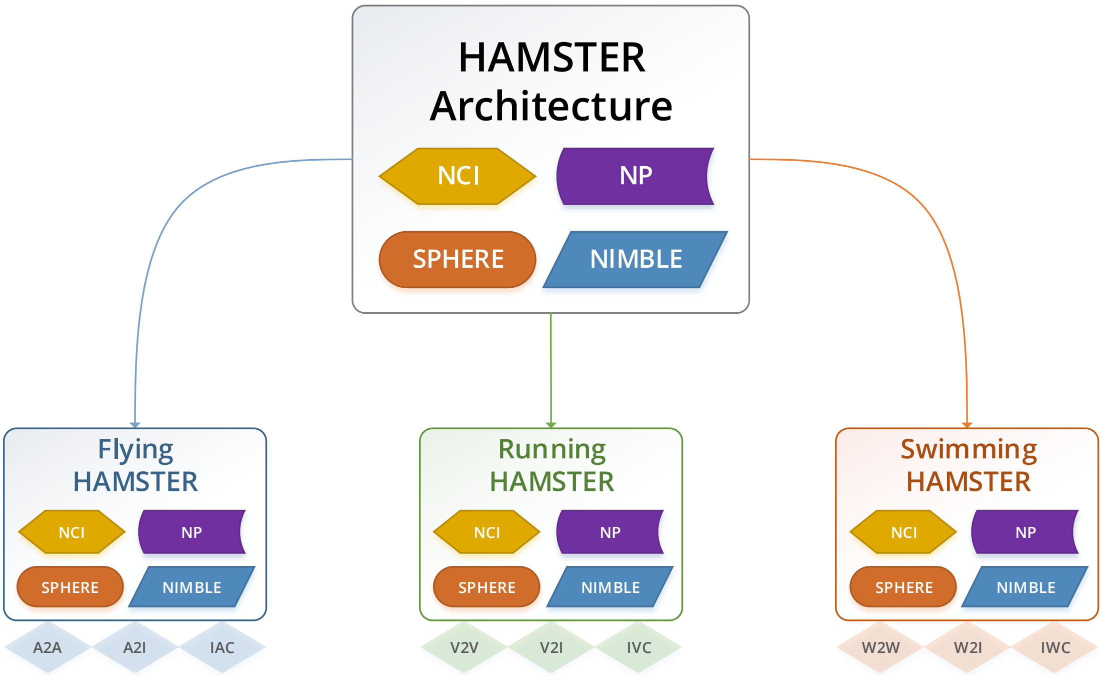
Flying HAMSTER deals exclusively with the aerial segment. It was defined based on specific characteristics and requirements of unmanned aerial vehicles (UAV) and unmanned aircraft systems (UAS). Flying HAMSTER deals specifically with internal aircraft communication (IAC), aircraft-to-aircraft communication (A2A) and aircraft-to-infrastructure communication (A2I).

The main applications of UAVs are related to agricultural and environmental monitoring, safety, military and civil defence. The aircraft is usually able to capture images for processing relevant information about a specific field, which may contribute to improve productivity. There are several cases where they might be applied in environmental and borders monitoring, or even applied as aerial sensors in networks for disaster management (QUARITSCH *et al.*, 2010) and multiple UAV applications (MAZA *et al.*, 2011; BOUACHIR *et al.*, 2014; LUO *et al.*, 2012; VERMA; FERNANDES, 2013).

Running HAMSTER deals specifically with vehicles on terrestrial segment. It was defined based on specific characteristics and requirements of unmanned ground vehicles (UGV) and unmanned ground systems (UGS). Running HAMSTER treats internal vehicle communication (IVC), vehicle-to-vehicle communication (V2V) and vehicle-to-infrastructure communication (V2I).

[1] SPHERE also provides "health" checking for modules. In this thesis, "health" refers to the good functioning of a module/system.

Figure 1 – HAMSTER versions and specific modules for criticality estimation (NCI), energy saving (NP), security and safety (SPHERE) and mobility (NIMBLE).



Source: Elaborated by the author.

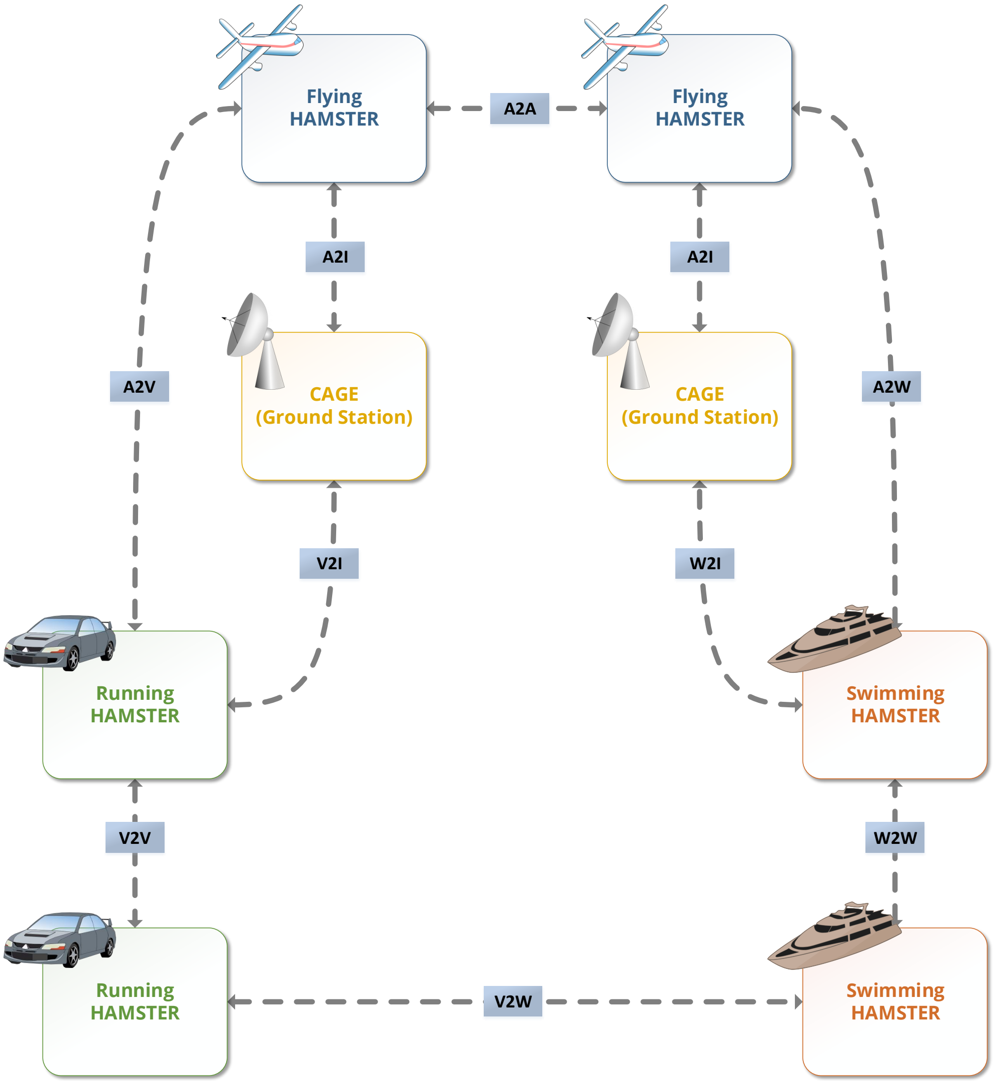
The objective of ground vehicles may vary from driver support in possible dangerous situations with the intention of preventing road accidents, to autonomous driving with no human intervention, which could be used in urban traffic, agriculture, industry and safety applications (WONG, 2008). The sensor fusion technique is used for integration of multiple sensors, such as cameras, digital compasses and GPS, allowing the vehicle to become autonomous in both urban and rural areas (SUN *et al.*, 2011).

Swimming HAMSTER was designed for vehicles that operate on aquatic environments. It was defined based on specific characteristics and requirements of unmanned surface vehicles (UGV), unmanned undersea vehicles (UUV) and unmanned water vehicles systems (UWS). Swimming HAMSTER is composed by internal water vehicle communication (IWC), water vehicle-to-water vehicle communication (W2W) and water vehicle-to-infrastructure communication (W2I).

The aquatic vehicles have been used for various tasks, especially those related to monitoring of oil exploration and maintenance of hydro-power. The current challenges for these vehicles go beyond autonomy, integrating other areas with the distributed and embedded systems, such as computer networks, artificial intelligence, software engineering, electrical, mechanical and mechatronics engineering, among others. The multiple vehicles tasks are also challenging (XIANG *et al.*, 2012; ABBOTT-MCCUNE *et al.*, 2013).

HAMSTER's goals include the integration of different unmanned systems. As previously stated, three main versions of HAMSTER were defined in this thesis. Nevertheless, HAMSTER can be extended for other vehicles and systems, e.g. unmanned trains, domestic robots. Thus, one of the most important features that a system following HAMSTER architecture model guarantees is the compatibility of communications and HAMSTER inherent functions, such as SPHERE, NIMBLE, NCI and NP, which is going to be detailed in next sections.

Figure 2 – Communications among different vehicles and systems enabled by HAMSTER.



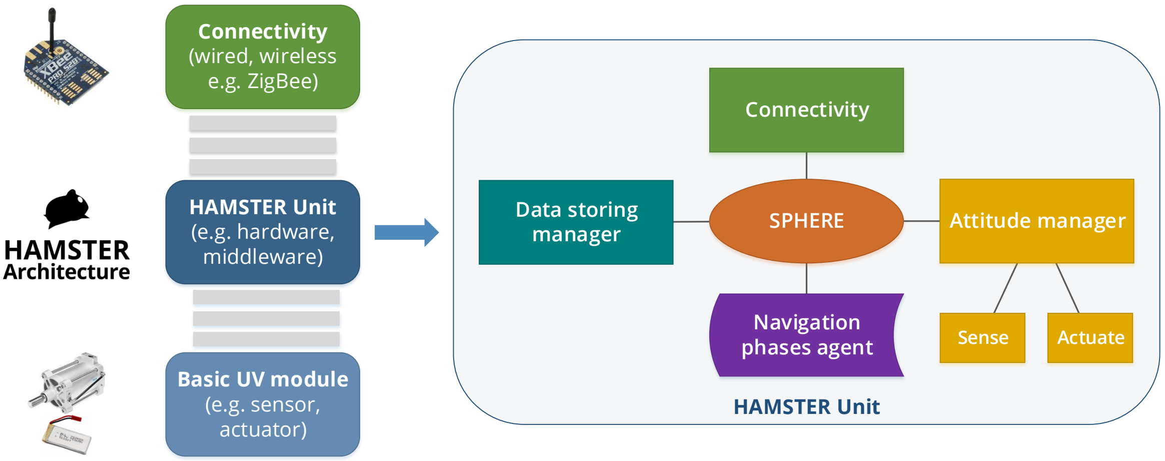
Source: Elaborated by the author.

The main communications covered by HAMSTER are generally defined as IMC, M2M and M2I. Although distinctly implemented in each system, approaches developed for each communication type are compatible. For example, a car and a ship could cooperate and share information if both are adapted with HAMSTER. Moreover, such communication would not be different than the communication with their own category of systems. For the identification of hybrid communications among different vehicles and systems, Figure 2 identifies communication possibilities enabled by HAMSTER. A special platform is also considered in this figure, which identifies a control station within HAMSTER architecture and assumes the name of CAGE (Control and monitoring AGEncy).

3.3 HAMSTER units

The architecture functions are implemented by a HAMSTER unit, which may be a dedicated hardware or software running on a microprocessor. The main functions include security and safety approaches, modules health checking, authentication, secure data storing and energy saving approaches. A HAMSTER unit is an intermediate module inserted between an element and the communication module (wired or wireless). The elements that compose a HAMSTER unit are: SPHERE, transmission manager, data storing manager, attitude manager and navigation phases agent. Figure 3 illustrates the mentioned elements.

Figure 3 – Structure of a general HAMSTER unit connected to an integrated communication module (e.g. an XBee radio that implements IEEE 802.15.4) coupled to an actuator (e.g. a motor).


Connectivity (wired, wireless e.g. ZigBee)
HAMSTER Architecture
HAMSTER Unit (e.g. hardware, middleware)
Basic UV module (e.g. sensor, actuator)
Connectivity
Data storing manager
SPHERE
Attitude manager
Navigation phases agent
Sense
Actuate
HAMSTER Unit

Source: Elaborated by the author.

The main module found on a HAMSTER unit is **SPHERE**. It is the implementation of features detailed in Section 3.4. As it deals with security and safety aspects, every incoming and outgoing message must be treated before being forwarded to other elements. Authorisations are also carried out by SPHERE.

The **transmission manager** deals with incoming and outgoing messages. All incoming messages are transmitted to SPHERE, which will check for the appropriate action to take. On the contrary way, an outgoing message is only transmitted if authorised by SPHERE. This approach ensures the security of transmitted data and the overall UV safety by allowing only well specified actions to proceed. The authorised messages will be sent and/or received by a physical module represented as Connectivity in Figure 3. Such modularisation provides abstraction to the basic UV module.

Some modules might store important data for future use or simply to keep a log of operation. Depending on the sensitivity of data, SPHERE may demand strong, light, or no encryption. Such a task is performed by **data storing manager**, which manages encryption keys together with SPHERE and also securely stores data on an exclusive memory.

As for actuating or sensing, an **attitude manager** module will proceed whether authorised by SPHERE. An exclusive module is used to ensure that actuation and sensing actions will take place concurrently with other inherent activities, reducing the chance of delays. For guaranteeing performance in real-time sensitive applications, better microprocessors should be used on the implementation of HAMSTER units.

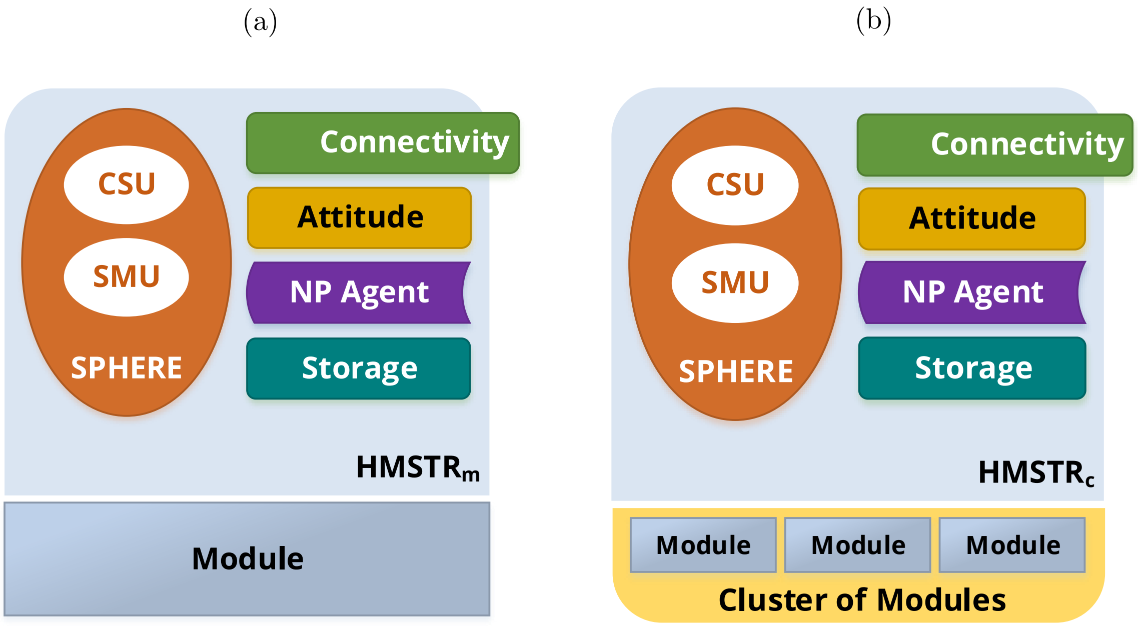
Finally, energy saving mechanisms are managed by the **navigation phases agent**. The operation of a UV can usually be split into several stages with key behaviour changes that can be taken into account while implementing energy saving approaches. For instance, considering the three main navigation phases for a UAV – takeoff, cruise and landing – one can determine that all mission modules should remain turned off during takeoff and landing, since they will probably be idle.

Different versions of HAMSTER units are used to integrate modules, clusters of modules and entities to the HAMSTER network. Three types were defined: HMSTR_{m} for modules, HMSTR_{c} for clusters of modules and HMSTR_{e} for entities. Modules and clusters of modules are usually more than one in a single UV and they will constantly communicate with an internal central for global synchronisation and common resources. Figure 4a presents the structure of a HMSTR_{m}.

The HAMSTER unit on clusters of modules was designed considering that some modules are too simple and do not demand an exclusive unit for operation. They can be grouped into clusters by function or proximity and operate synchronously for better results either in communication and sensing/actuation. Figure 4b presents the structure of a HMSTR_{c}. The HMSTR_{c} is connected to a cluster (in yellow) and coordinates all the modules that are part of the cluster.

Finally, while applied on entities, the HAMSTER unit is also in charge of external communications. That leads to the manipulation of received and transmitted data and its

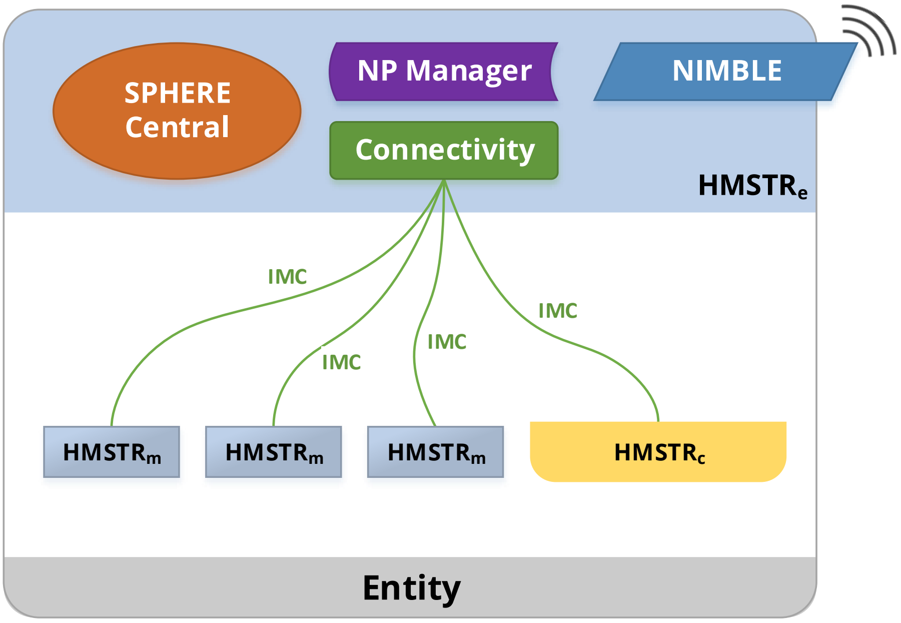
Figure 4 – HAMSTER units: (a) embedded on a UV module, HMSTR$_m$; and (b) embedded on a UV cluster of modules, HMSTR$_c$.



Source: Elaborated by the author.

conversion to the internal/external networks. Figure 5 presents the structure of a HMSTR$_e$.

Figure 5 – HAMSTER unit embedded on a UV entity, HMSTR$_e$.



Source: Elaborated by the author.

3.4 SPHERE, the safety and security platform

The name SPHERE comes from the idea of a hamster wheel that allows the animal to play in a safe way. As an ordinary sphere has the visual concept of wrapping things, it was chosen as the name of the safety and security platform for HAMSTER architecture. It concentrates all the safety and security aspects of the main architecture and all derivative versions. The aim is to ensure safety and security, allowing every unmanned vehicle running a HAMSTER-based architecture to safely share information, even when different scenarios are involved, e.g. to permit the safe communication between an unmanned surface vehicle and an unmanned aerial vehicle. It is also a goal of SPHERE to centralise the modules health checking, which guarantees a safer operation for the vehicle and, consequently, the entire system. Although the platform may have centralised modules, it is not a centralised platform. SPHERE is present in several parts of the system according to its inherent necessities.

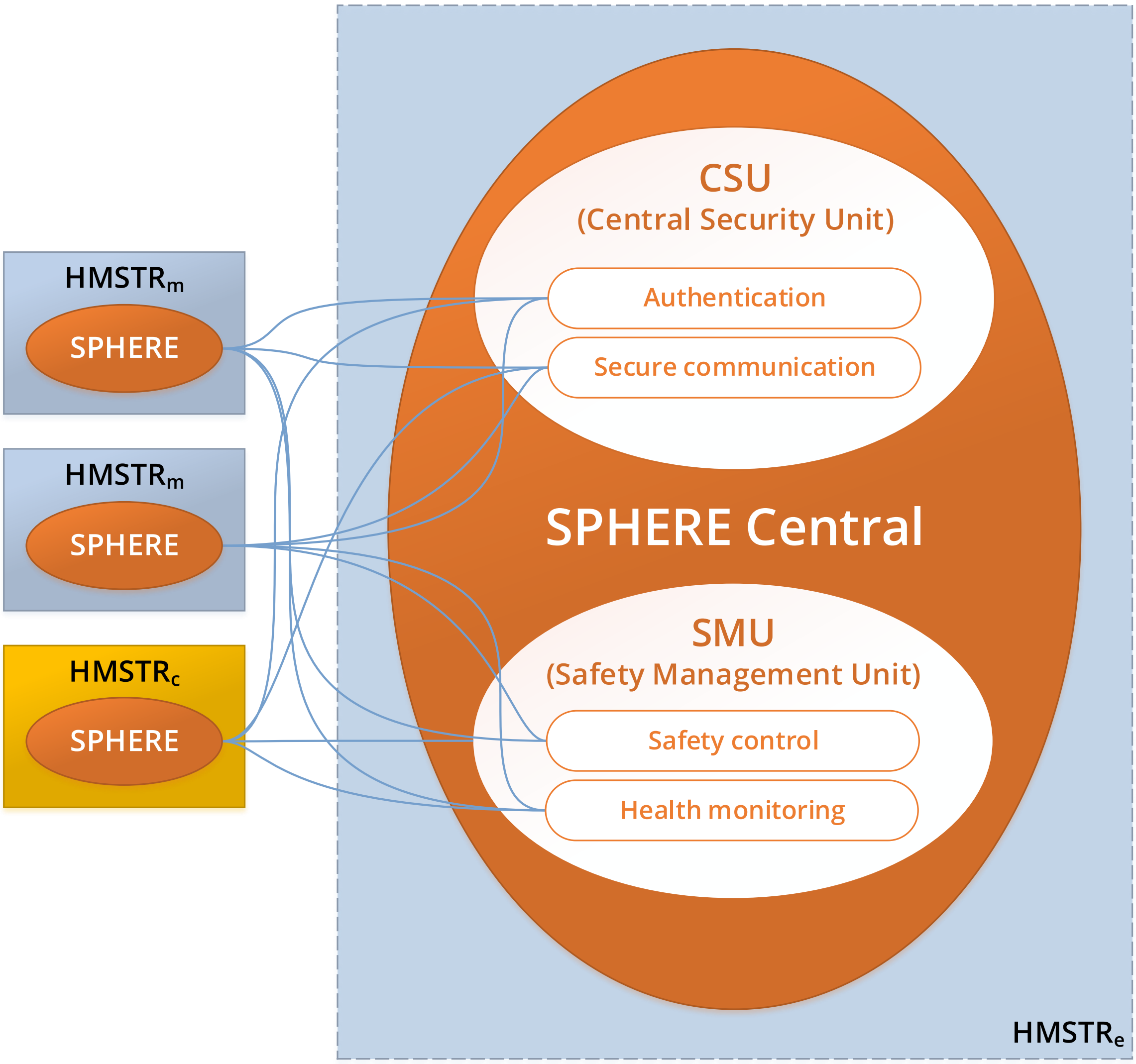
SPHERE platform can be explained in three main topics. First, a components usage policy is addressed (Subsection 3.4.1). Then, SPHERE modules are presented (Subsection 3.4.2). Finally, an authentication protocol which is in charge of components health checking and authentication will be discussed (Subsection 3.4.3). It is important to point out that such functions will be performed by different SPHERE subsystems. Figure 6 presents SPHERE and its subsystems. CSU (Central Security Unit) and SMU (Safety Management Unit) share the implementation of SPHERE's main functions and will be detailed in the end of this section.

3.4.1 Components usage policy

Critical systems in domains such as aviation, railway and automotive are often subject to a formal process of safety certification. The goal of this process is to ensure that these systems will operate safely without posing undue risks to the user, the public, or the environment (NAIR *et al.*, 2014). Therefore, one of the first steps on the direction of ensuring the safe operation of a UV and facilitate its integration into the space of actuation (e.g. a UAV into the airspace) must be the redefinition of its components usage policy. In some cases, only a few (critical) parts of a UV are properly treated to ensure that all connected modules are authentic and have not been replaced or tampered with by a third party. The current policy adopted by manufacturers uses a concept of "Accept all" which trusts in all components embedded in a UV. SPHERE assumes an "Almost Deny All" approach, which denies the authenticity of all mechanical components and peripherals attached to the vehicle until the opposite is proved, which may result in safer vehicles.

The categorisation of every module is therefore crucial for such a new security model to be applied to UVs. There are various peripheral devices embedded in a UV that

Figure 6 – SPHERE central platform and subsystems.



Source: Elaborated by the author.

require different levels of security, which leads to the necessity of a module categorisation according to the criticality of their performed functions. The SPHERE platform uses an index for criticality classification called Node Criticality Index (NCI). This index independently considers aspects of security and safety, but also provides a merged score in cases of interdependence. Although NCI was originally defined in early SPHERE proposal, it became a very relevant score which started being applied not only for security and safety purposes, but also demanding for more independence as a consequence. Section 3.5 will provide a wider explanation of NCI.

Despite the fact that NCI considers other classifications, in this chapter we will consider only main and mission-specific modules for a better explanation. Main modules are those considered essential components for the UV to operate, be aware of its location and be able to perform an emergency operation abort safely, even when the mission was not entirely concluded. An autopilot, a GPS receiver and an inertial unit are examples of modules classified as main, since they might cause big issues in case of failure. In contrast, modules not considered as essential to the UV operation are classified as mission-specific modules. Whether abnormal behaviours are detected in any mission-specific module, the operation of main nodes should not be affected. However, the problematic mission-specific module should remain disabled or at least isolated from the network not to compromise any other module or subsystem. It implies that all main modules must be authenticated before the operation begins. In contrast, the mission-specific modules do not necessarily need an authentication before the UV operation, even though it is highly recommended.

Additionally, in order to protect the UV against malicious attacks, there is the possibility of identifying anomalies due to operating time. For instance, repetitive collisions and pressure effects on a UV may cause natural degradations in components integrity. Therefore, mechanisms to identify the existence of unusual behaviours should help increasing UV safety, even with a consequent abort of a mission in order to keep the UV's physical integrity. These concepts are strongly connected to sense and avoid area, which are not yet addressed by HAMSTER, but could be further explored in future works.

3.4.2 SPHERE modules

SPHERE is implemented by two modules (as seen in Figure 6):

- The **Central Security Unit (CSU)** provides security services on SPHERE. Two main services are provided by the CSU:
  - **Authentication** deals with one of the most important phases to guarantee the genuineness of UV modules. In this phase, it is important to verify all the modules and clusters before starting the UV operation, including SPHERE Central, ensuring no intruder is allowed to communicate with the system; and
  - **Secure communications** takes care of cryptographic techniques to guarantee confidentiality for communications. Different levels of importance may be attributed to data, which leads to appropriate approaches regarding cryptography managed by CSU.
- The **Safety Management Unit (SMU)** concentrates safety-related tasks on SPHERE. Two main services are provided by SMU:
  - **Health monitoring** is constantly verifying UV modules for misbehaviour due to time of operation or anomalies that may happen due to unexpected failures. For instance, this verification may compare a component datasheet with its

real behaviour history, triggering an appropriate action, such as component substitution or repair; and

- **Safety control** which frequently verifies the overall UV for anomalies and takes appropriate action for UV's and nearby systems' or people's safety.

3.4.3 Protocol structure

To protect the UV against attacks that may come from malicious modules, SPHERE implements a security policy. It is not just necessary to ensure that all modules are authentic, but also monitor their health status, taking appropriate actions if needed, e.g. stop communication with a failing module or return home. Furthermore, such policy must be applicable even during UV operation, considering that external factors may affect the components behaviour, e.g. climate or weather changes.

On start-up, SPHERE must also be authenticated just like any other ordinary module. It will be storing tables of public keys of all UV modules, operating as a Certification Authority (STALLINGS, 2008), ensuring that a public key belongs to a module. Each module will store a hash table of the keys for integrity checking. The process starts with a mutual authentication phase among modules and CSU's Authentication unit. It checks a database for information about all known modules, getting access to their criticality, and if there is any access restriction associated to that module. There is also the possibility of deciding whether a module should be initialised or not during the verification stage. The initialisation depends on the Navigation Phases status, which will be detailed in Section 3.6. The following steps on CSU's Authentication process will be the authorisation and, if positive, the exchange of encrypted messages to establish a secure channel for communication among modules and CSU.

After such handshake, three situations are expected:

1. Either the module that is trying to authenticate and the SPHERE have not been tampered with;

2. The module has been tampered with and therefore has not been authenticated, leading to two situations:

   a) If it is a main module, the UV must not operate;

   b) If it is a mission-specific module, all other modules must stop communicating with the attacked module.

3. The module that is trying to authenticate may notice that SPHERE is not authentic and must notify other components about it.

From the point of view of communication security, an ideal situation would be if all modules could authenticate with others. However, this method would cause a system overload, since the increase of modules in the UV would cause an exponential increase in the number of exchanged messages. To solve such problem there have been proposed e-voting protocols (LIAW, 2004). In case of non-authentic SPHERE, protocols such as those presented in Kikuchi and Nakazato (2004) might be used. This model can be further expanded according to the needs of UVs, including a negotiation mediated by CSU to create a secure channel of communication among modules.

3.5 NCI, the node criticality index

The Node Criticality Index (NCI) is a key feature provided with HAMSTER and consists on a rich index to help determining single and global priorities for nodes within a network. It is applied to M2M, M2I and IMC communications aiming at the provision of QoS, security, safety and prioritisation approaches for modules, clusters of modules and entities. This approach is flexible enough to encompass different sets of goals based also on mission information.

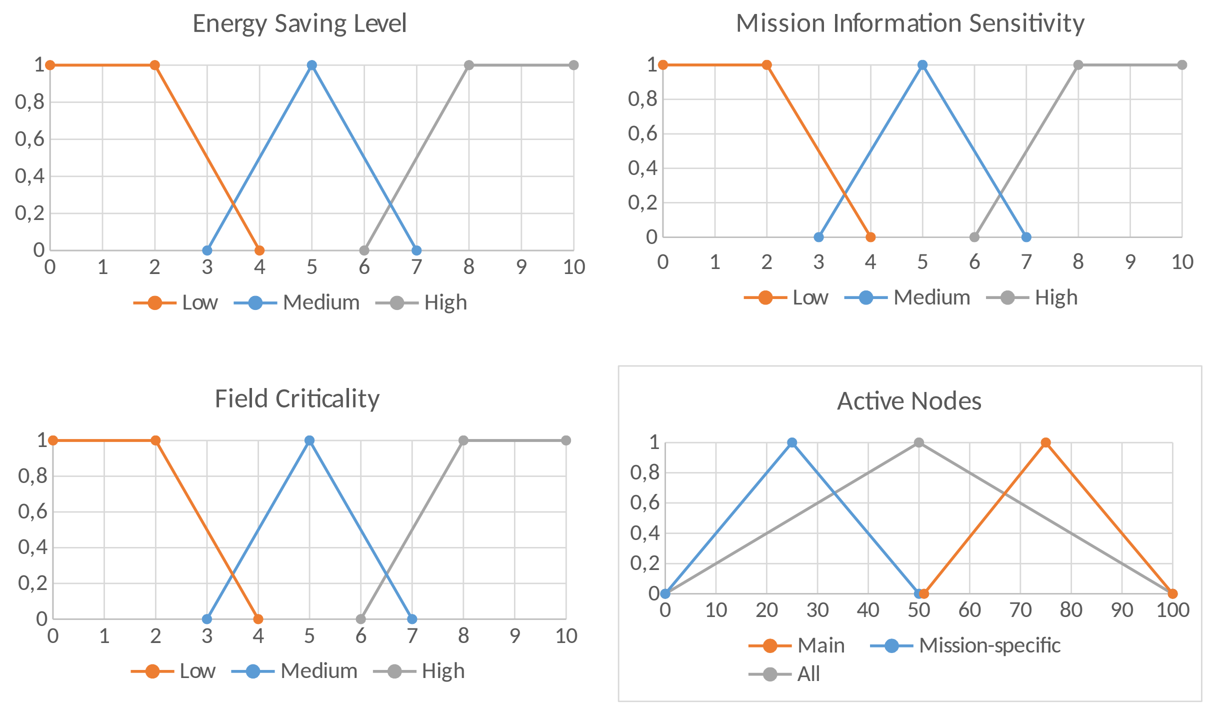
An initial investigation towards energy savings on internal communications was inspired by Fuzzy logic, which let the creation of Navigation Phases platform (Section 3.6 will address the subject). Although early several analyses were conducted on how to save energy by turning off idle nodes, the main issues identified were related to the difficulty of specifying the criticality of a mission field, the mission information sensitivity and the level of energy saving in order to have results on the output variable related to active nodes, as illustrated by charts in Figure 7. However, it was later identified that there was a relevant necessity for a formal way to measure criticality on unmanned systems and also that, in the field of small UVs, a naive, Fuzzy logic-inspired approach would suit better due to inherent simplicity of the vehicle. Thus, NCI was defined towards the criticality identification.

This section details the methodology used to define NCI index. NCI is designed to work in three different situations: i) the internal network connecting basic and mission-specific modules individually, ii) the internal network connecting clusters of modules and iii) the external network among unmanned vehicles and eventual infrastructure entities.

3.5.1 NCI on HAMSTER modules

As presented in Section 3.1, a HAMSTER module consists on a sensor, an actuator, or any other module connected to the unmanned vehicle inner network. Such modules may denote different levels of importance regarding security and safety of specific unmanned vehicles and elements that interact with or share their operation field. Thus, the deter-

Figure 7 – Definition of input and output (highlighted) variables on a Fuzzy logic system that inspired the creation of NCI.



Source: Adapted from Pigatto *et al.* (2016).

mination of an index, and even more importantly, several sub-indices, will provide the system with valuable information that may influence tasks decision making.

First of all, every module has to be assigned with independent security and safety scores. In this context, security is the maximum score obtained by measuring two types of data: *storedData* (data stored by a module) and *temporaryData* (data manipulated by a module, but not stored). Both *storedData* and *temporaryData* must have independent approaches since eventual safety and security related issues will impact the system in different ways, e.g. stored data becomes a potential security concern if an unmanned vehicle is eventually stolen or captured; on the other hand, temporary data is a relevant safety concern for an under-attack unmanned vehicle as it will probably contain control messages that must override the autopilot assuming it is the attacked module.

The determination of each score takes into account the necessity for different approaches. The security score is a sensitive task which must be attributed by a specialist via a complete formal analysis based on appropriate datasheets and usage statistics. Although this thesis' scope does not include the automatic attribution of such score, it is an open research topic for future integration. The fact that it demands a human intervention is not necessarily an issue, since it is performed only once before the unmanned vehicle operation start. Basically, the score attribution is first performed during a setup phase and then automatically updated as a consequence of changes and events during the system operation. The score attribution must be a number within a range from 0 to 1, meaning ordinary and critical data, respectively. Equation 3.1 shows the main formula for a module's NCI security sub-index, represented as $NCIm_i^{sec}$.

$$NCIm_i^{sec} = max(storedData_i, temporaryData_i) \tag{3.1}$$

where m refers to a HAMSTER module; i indicates the module; sec identifies the index as security-related only; $storedData$ is a score between $[0,1]$ that represents the sensitivity of stored data; and $temporaryData$ is a score between $[0,1]$ that points out the sensitivity of manipulated data. A suggested classification is provided in Table 1.

The second score is related to safety. In this context, safety calculation is based on the average mean of two scores: $health$ (a score that represents whether a module is properly working or experiencing issues) and $modulePriority$ (the importance of a single module to the system).

These scores must be numbers from 0 to 1, meaning ordinary and critical impact, respectively. Equation 3.2 represents the general formula for module's NCI safety sub-index, represented as $NCIm_i^{saf}$.

$$NCIm_i^{saf} = average(health_i, modulePriority_i) \tag{3.2}$$

where saf identifies the index as safety-related only; $health$ is a score between $[0,1]$ that represents the health status of a module; and $modulePriority$ is a score $[0,1]$ that identifies the importance of a module to the overall system safety.

Now, a HAMSTER module's NCI ($NCIm_i$) can be found by calculating the average mean between security and safety sub-indices, as represented in Equation 3.3.

$$NCIm_i = average(NCIm_i^{sec}, NCIm_i^{saf}) \tag{3.3}$$

Table 1 – Suggested criticality statuses for selected variables on the NCI calculation.

	Criticality statuses			
	Minor	**Marginal**	**Critical**	**Catastrophic**
storedData and *temporaryData*	There will be no damage or business harm if data happens to be accessed by unauthorised parties.	Small financial loss may be experienced in case of unauthorised data access. Basic data encryption should be considered.	Significant financial loss may be experienced in case of unauthorised data access. Strong data encryption should be considered.	Huge financial loss may be experienced in case of unauthorised data access. Strong data encryption associated with other backup/ security techniques are strongly recommended.
health	The module is operating as expected. The module's datasheet or a history of operation are usually taken as reference.	A secondary function provided by the module is temporarily unavailable. This does not compromise its main operation.	A secondary function provided by the module is definitely unavailable. Although it does not compromise its main operation, a repair should be carried out very soon.	The module is not operating as it should. The vehicle must not proceed with operation or find an alternative source of information.
modulePriority	The module is not important for the system operation at all.	The module is important, but its function can be performed by another one.	The module is very important for specific phases of the system's operation.	The module is very important to the system and the vehicle will not operate without it.

Source: Elaborated by the author.

3.5.2 NCI on HAMSTER clusters of modules

HAMSTER clusters of modules are groups of sensors, actuators, or any other module located inside an unmanned vehicle with similar or related function or set of functions, as stated in Section 3.1. The NCI calculation for clusters of modules ($NCIc_j$) is obtained by finding the maximum $NCIm$ among all modules in a specific cluster, as seen in Equation 3.4. The range is from 0 (ordinary impact) to 1 (critical impact).

$$NCIc_j = max(NCIm_i) \mid m_i \in c_j \tag{3.4}$$

where c refers to a HAMSTER cluster of modules; and j identifies the cluster.

Similarly to the modules, there is the possibility of determining clusters' security and safety sub-indices, as seen in Equations 3.5 and 3.6, respectively.

$$NCIc_j^{sec} = max(NCIm_i^{sec}) \mid m_i \in c_j \tag{3.5}$$

$$NCIc_j^{saf} = max(NCIm_i^{saf}) \mid m_i \in c_j \tag{3.6}$$

3.5.3 NCI on HAMSTER entities

Although NCI was designed for all SPHERE domains, the most complex one is the NCI for HAMSTER entities. As defined in Section 3.1, a HAMSTER entity is an element of an unmanned system which is connected to a network via M2M and/or M2I. The NCI for entities impacts external activities mainly.

There are two important elements that must be taken into account while determining an entity's NCI ($NCIe$). Firstly, an evaluation of the mission criticality must be conducted, aiming at reflecting the real implications of a mission ($missionPenalty$) to the UV criticality. In this case, the $field$ of execution and the importance of a mission fully $accomplishment$ are used, as stated by Equation 3.7.

$$missionPenalty_k = max(field_k, accomplishment_k) \tag{3.7}$$

where k indicates the entity; $field$ is a number between $[0,1]$ that represents the sensitivity of a geographic region or environment where the mission is performed; and $accomplishment$ is a number between $[0,1]$ that reflects the importance of fully accomplishing a mission.

Secondly, a UV's estimated financial cost must also be known. That leads to the calculation of how ***worth*** a vehicle is within the scenario and other UVs nearby, which is presented in Equation 3.8. It is important to point out that a specialist is needed in this case, since there should not be much discrepancy among UVs' estimated costs, allowing NCI to be in fact useful.

$$worth_k = cost_k / max(cost) \tag{3.8}$$

where $cost_k$ is a financial value that is usually the acquisition price or production value invested with an entity k and $cost$ is a vector composed by all costs from all entities involved.

Hence, being aware of *missionPenalty* and *worth* values, it is possible to generally determine NCI for entities, as represented in Equation 3.9.

$$\begin{aligned} NCIe_k &= 0.9 * ((1 - missionPenalty_k) * average(NCIm_i, NCIc_j) \\ &+ missionPenalty_k * max(NCIm_i, NCIc_j)) \\ &+ 0.1 * worth_k \mid m_i, c_j \in e_k \end{aligned} \tag{3.9}$$

As done for modules and clusters of modules, individual NCI's security and safety sub-indices are also available, allowing targeted, specific applications. Equations 3.10 and 3.11 show how they are calculated.

$$\begin{aligned} NCIe_k^{saf} &= (1 - missionPenalty_k) * average(NCIm_i^{saf}, NCIc_j^{saf}) \\ &+ missionPenalty_k * max(NCIm_i^{saf}, NCIc_j^{saf}) \mid m_i, c_j \in e_k \end{aligned} \tag{3.10}$$

$$\begin{aligned} NCIe_k^{sec} &= (1 - missionPenalty_k) * average(NCIm_i^{sec}, NCIc_j^{sec}) \\ &+ missionPenalty_k * max(NCIm_i^{sec}, NCIc_j^{sec}) \mid m_i, c_j \in e_k \end{aligned} \tag{3.11}$$

3.6 NP, the navigation phases platform

Towards the reduction of energy consumption and better activity control of UV modules, HAMSTER provides the navigation phases (NP) concept. A navigation phase is a very well defined UV operation stage where it is attributed at least an ON/OFF state and different transmission rate permissions for each single module. The most common classification splits modules into two categories: “mission-specific” and “main” modules. For instance, a UAV would have at least three main phases: takeoff, cruise and landing. Each phase would prioritise different modules over others, considering their individual demand for operation.

A general classification of a UV navigation phases is presented in Table 2. There are five main phases and an emergency one. Main phases will usually follow a predictable order. On the other hand, emergency phases can be started at any time to treat adverse conditions. Each navigation phase defines which category of nodes is allowed to be working at that specific situation.

Table 2 – General navigation phases for unmanned vehicles.

Navigation phases		Sub-navigation phases		Active modules
1	Start-up	1.1	Modules health checking	All
		1.2	Energy supply verification	All
		1.3	Authentication	All
2	Operation initialisation	2.1	Operation start	Main
		2.2	Moving to the target field	Main
3	Mission execution	3.1	Positioning on the field	Main
		3.2	Performing mission	All
4	Shutdown	4.1	Moving back home	Main
		4.2	Preparing to stop operation	Main
		4.3	Turning off vehicle	Main
5	Post-operation	5.1	Modules health checking	All
		5.2	Mission data acquisition	Mission-specific
E	Emergency situations	E.1	Operation abortion and home returning	Main
		E.2	Operation abortion and vehicle turning off	Main
		E.3	Data self-destruction (wipe data)	Mission-specific
		E.4	Stabilising (after non-predicted disturbances)	Main

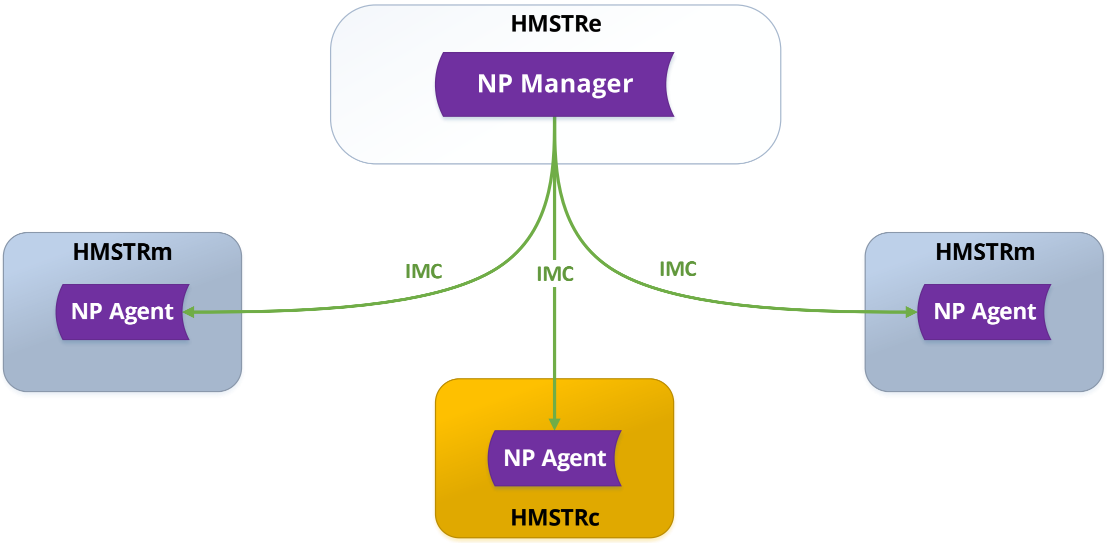
Source: Elaborated by the author.

The **Start-up** phase is dedicated for several preoperational tasks, e.g. modules health checking, energy supply verification and authentication. Probably, the transmission rates allowed for each node will be similar, considering that performed tasks are not differently implemented for adversary classes of nodes. That is not the case of **Operation initialisation** phase, which must prioritise UV main modules over mission-specific ones to guarantee that the UV will start operating and will successfully move to the target field. This can be considered a safety-critical operation that should meet restrict time requirements.

Next, a natural phase is the **Mission execution**. Unmanned vehicles will usually be designed mainly (often exclusively) for that purpose. Therefore, the prioritisation of mission modules is an approach that might take place. Although very important, mission-specific nodes cannot operate by themselves; thus, all nodes should work in this phase with different transfer rates. Depending on the module, transfer rates may vary. From mission execution, the UV will likely move on to a **Shutdown** phase. The operation is similar to the Operation initialisation in criticality terms.

Finally, the UV must be put in a **Post-operation** phase. This stage can have different operation patterns. For instance, while in mission data acquisition sub-phase, distinct transfer rates will be allowed to each mission-specific module depending on the importance or amounts of data they have stored throughout the operation.

Figure 8 – Navigation phases interaction: NP Agent and NP Manager.



Source: Elaborated by the author.

Emergency situations phase concentrates adversary and unpredictable situations that may be experienced by the system. For instance, if batteries are in critically low levels, the UV just collided, or a possible unauthorised entity is trying to steal sensitive information from the UV, an emergency situation might be started. Perhaps, one can consider this as the most SPHERE-connected phase.

Navigation phases are centrally managed by a NP Manager placed on a $\mathrm{HMSTR_e}$ unit. Each $\mathrm{HMSTR_c}$ and $\mathrm{HMSTR_m}$ unit is provided with a NP Agent that will proceed with orders originated from NP Manager. Figure 8 presents the interaction between NP Agent and central NP Manager. On a modern version of NP, one can consider NCI for determination of modules restrictions table.

3.7 NIMBLE, the mobility platform

Be it an infrastructured network or a mobile ad hoc network, mobility is an important requirement for UVs. In the air, mobility can be even more challenging than in roads and in the ocean, but all of them have somehow a demand for mobility. Naturally, ad hoc is the most challenging operating mode, since it does not rely on a fixed centralised access point and nodes move at high speeds and (sometimes) in unpredicted paths. Although seeming not so challenging, infrastructure-based communications may demand mobility approaches as well. That way, NIMBLE was conceived with the view to encompass mobility in M2X communications on HAMSTER architecture.

Communication is a crucial aspect of the design of multiple-vehicle systems and one of their biggest challenges (BOUACHIR *et al.*, 2014; CHUNG *et al.*, 2011b). In the simplest scenario, all vehicles are directly connected to a common infrastructure and this can act as an intermediary for all communications among them. However, this strategy has several problems. Firstly, each vehicle must be equipped with expensive and complex hardware in order to perform the long-distance communication with the control station or satellite. Secondly, many factors may compromise communication reliability, such as changing environmental conditions, the high mobility of vehicles, different terrain topologies or obstacles. Finally, the typical use of a ground control station (GCS) to provide the communication infrastructure limits the mission target locations to the GCS coverage area, since beyond that vehicles disconnect from the network and become unreachable.

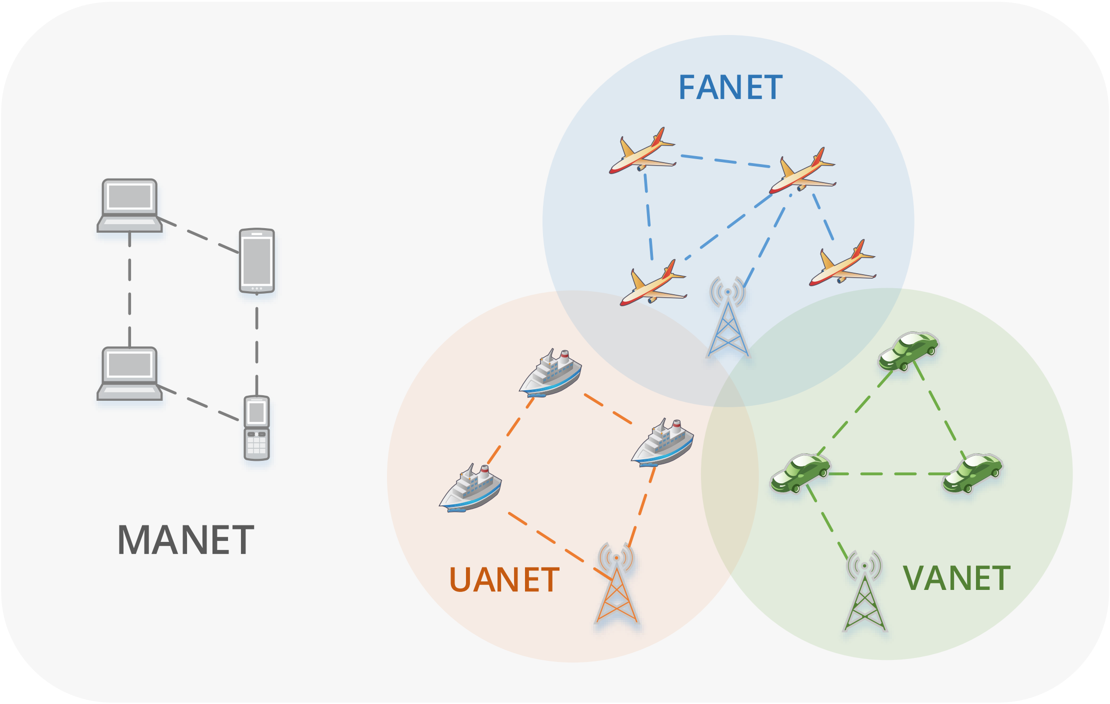
The implementation of an ad hoc network connecting all vehicles is one of the most feasible alternatives to infrastructure-based communication. An ad hoc network is composed by nodes that also act as routers, forming a temporary network with no fixed topology or centralised administration (SARKAR; BASAVARAJU; PUTTAMADAPPA, 2008). This approach increases the mission target area, since communications among vehicles and the GCS can be routed through other vehicles in a series of hops. Also, even if there is no connection to a GCS, the nodes can form an ad hoc network to share information or work in cooperation.

Ad hoc networks are classified according to their implementation, utilisation, communication and mission objectives. If the nodes that compose an ad hoc network are mobile, the network is classified as MANET (Mobile Ad hoc NETwork). For vehicle-specific applications, MANETs are sub-divided into UANET (Underwater Ad hoc NETwork) for aquatic vehicles, VANET (Vehicular Ad hoc NETwork) for terrestrial vehicles, or FANET (Flying Ad hoc NETwork) for aerial vehicles (BEKMEZCI; SAHINGOZ; TEMEL, 2013; SAHINGOZ, 2014), as illustrated by Figure 9.

Each type of vehicular network faces different, unique challenges: for instance, a UANET must deal with an underwater transmission medium and VANETs often encounter unexpected road obstacles. However, it has been recognised that FANETs have to address more challenging issues than other ad hoc networks (BEKMEZCI; SAHINGOZ; TEMEL, 2013; SAHINGOZ, 2014), because of the following specific characteristics:

- **Higher node mobility**. FANET nodes typically have higher mobility than those

Figure 9 – Relationships among different types of mobile ad hoc networks (MANET): underwater ad hoc networks (UANET), vehicular ad hoc networks (VANET) and flying ad hoc networks (FANET).



Source: Elaborated by the author.

in other types of MANET. As a result, a FANET's network topology can change more frequently, which increases the overhead caused by connecting and routing operations.

- **Multiple connections**. In many applications, the nodes in FANETs collect environmental data and then retransmit it to the control station, similarly to wireless sensor networks (RIEKE; FOERSTER; BROERING, 2011). Therefore, FANETs have to manage multiple communications between UAVs and ground control stations, as well as providing support to peer-to-peer connections among UAVs.

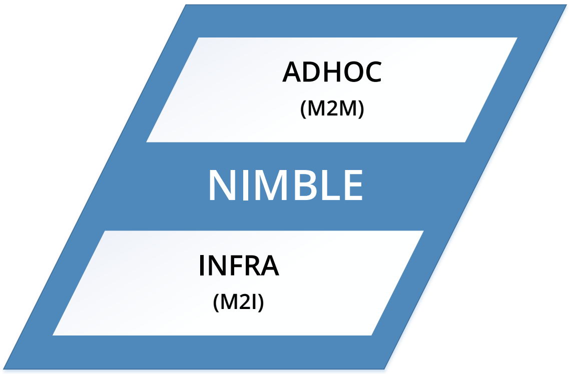
- **Very low node density**. Typical distances among nodes in FANETs are usually longer than in MANETs and VANETs (CLAPPER *et al.*, 2007); thus, the communication range in FANETs must also be greater than in other networks. This imposes more demanding requirements for radio links and other hardware elements.

- **Heterogeneity**. UAV systems may include heterogeneous sensors and each of them may require different strategies for data distribution.

- **Obstacles**. Due to the higher node mobility, obstacles may randomly block links among UAVs, which must be addressed in order to provide different temporary communication paths, avoiding the disconnection of nodes.

External communications on HAMSTER architecture are dealt by NIMBLE platform. Figure 10 illustrates NIMBLE's sub-modules: ADHOC and INFRA. M2M communications, including mostly MANET and derivative networks, are managed by ADHOC. On the other hand, INFRA is aimed at the management of infrastructured communications, e.g. satellites and GCS (in HAMSTER case, identified as CAGE). These sub-modules also concentrate efforts on mobility models improvements.

Figure 10 – NIMBLE mobility platform is composed by ADHOC and INFRA sub-modules for external communications.



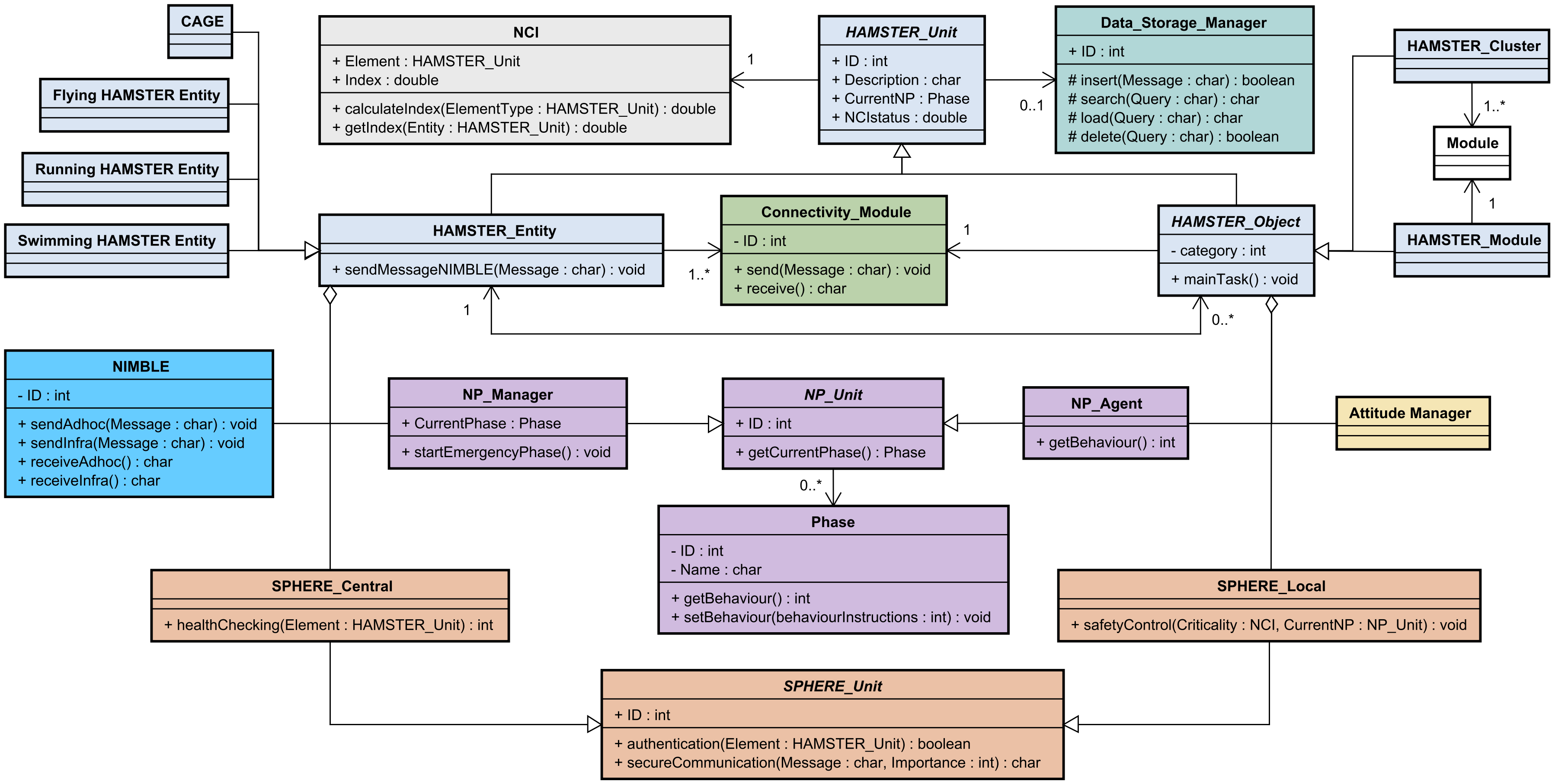
Source: Elaborated by the author.

The separation into two different modules provides the advantage of allowing better approaches to each external communication. For instance, while communicating with an infrastructure, INFRA sub-module will most certainly need to transmit with higher signal strength due to longer distances from the UV to the infrastructured element. On the contrary, while communicating with others UVs, ADHOC should use appropriate ad hoc routing protocols for better message delivery.

3.8 Modelling

The overall proposal of HAMSTER architecture is presented as a reference model with UML that helps illustrating the relationships among its components (see Figure 11).

Figure 11 – Class diagram for HAMSTER Architecture.


CAGE
Flying HAMSTER Entity
Running HAMSTER Entity
Swimming HAMSTER Entity
NCI
+ Element : HAMSTER_Unit
+ Index : double
+ calculateIndex(ElementType : HAMSTER_Unit) : double
+ getIndex(Entity : HAMSTER_Unit) : double
1
HAMSTER_Unit
+ ID : int
+ Description : char
+ CurrentNP : Phase
+ NCIstatus : double
0..1
Data_Storage_Manager
+ ID : int
# insert(Message : char) : boolean
# search(Query : char) : char
# load(Query : char) : char
# delete(Query : char) : boolean
HAMSTER_Cluster
1..*
Module
1
HAMSTER_Module
HAMSTER_Entity
+ sendMessageNIMBLE(Message : char) : void
1..*
Connectivity_Module
- ID : int
+ send(Message : char) : void
+ receive() : char
1
HAMSTER_Object
- category : int
+ mainTask() : void
1
0..*
NIMBLE
- ID : int
+ sendAdhoc(Message : char) : void
+ sendInfra(Message : char) : void
+ receiveAdhoc() : char
+ receiveInfra() : char
NP_Manager
+ CurrentPhase : Phase
+ startEmergencyPhase() : void
NP_Unit
+ ID : int
+ getCurrentPhase() : Phase
NP_Agent
+ getBehaviour() : int
Attitude Manager
0..*
Phase
- ID : int
- Name : char
+ getBehaviour() : int
+ setBehaviour(behaviourInstructions : int) : void
SPHERE_Central
+ healthChecking(Element : HAMSTER_Unit) : int
SPHERE_Local
+ safetyControl(Criticality : NCI, CurrentNP : NP_Unit) : void
SPHERE_Unit
+ ID : int
+ authentication(Element : HAMSTER_Unit) : boolean
+ secureCommunication(Message : char, Importance : int) : char

Source: Elaborated by the author.

Abstract class *`HAMSTER_Unit`* is the architecture core, providing an abstraction to any element supporting HAMSTER architecture. Each unity has a unique ID and a NCI value, represented by attributes identification `ID` and `NCIStatus`, respectively, and may or may not store data through a `Data_Store_Manager`.

`HAMSTER_Unit` specialisation of `HAMSTER_Entity` abstracts vehicles and other elements that compose the unmanned system and may be specialised to different versions (Flying HAMSTER, Running HAMSTER, Swimming HAMSTER or any other new version) if needed. Abstract class *`HAMSTER_Object`* abstracts the sensing/actuating modules and clusters of modules and is implemented respectively by `HAMSTER_Module` and `HAMSTER_Cluster` classes.

SPHERE platform and Navigation Phases Manager are represented respectively by abstract classes *`SPHERE_Unit`* and *`NP_Unit`*. Due to the different tasks performed by those platforms depending whether the HAMSTER unit is an entity or a module or cluster of modules, their abstraction is specialised in different implementations that are associated with the specialised HAMSTER Unit classes rather than with super class, facilitating the implementation of different features for different units.

Therefore, SPHERE platform is implemented in a HAMSTER entity through class `SPHERE_Central` and in a HAMSTER object through class `SPHERE_Local`; the Navigation Phases Manager is implemented by classes `NP_Manager` and `NP_Agent`, as discussed in Figure 8.

`Nimble` class abstracts the platform and is associated with HAMSTER entity, since only entities have communication with the outer world. On the other hand, only HAMSTER objects aggregate an Attitude Manager, since they are the unities performing tasks in the system.

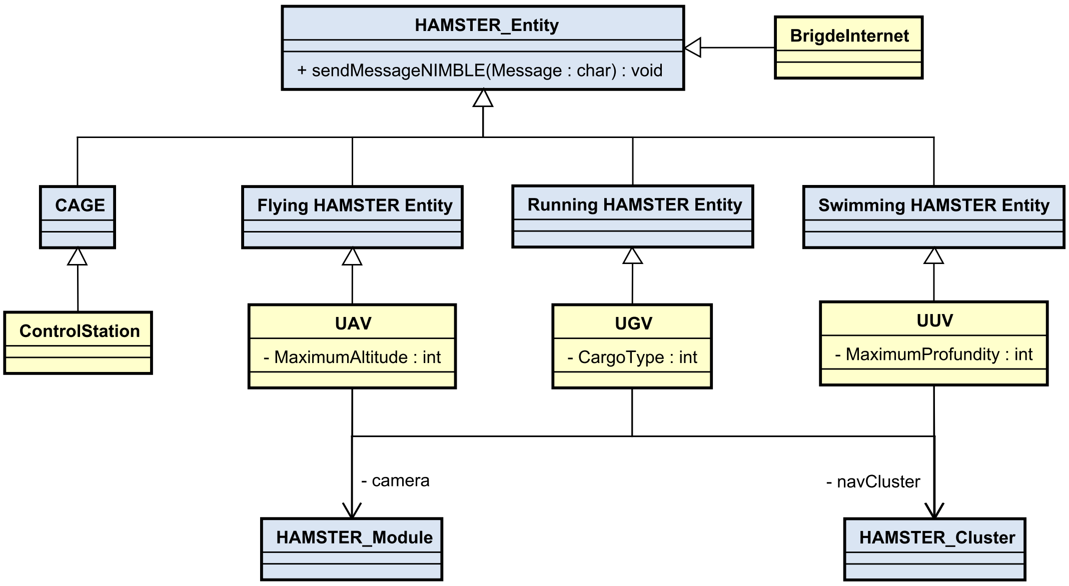
In order to illustrate the implementation of HAMSTER modelling, a possible application of unmanned systems is described and its respective object diagram presented.

The scenario used in this example is the supervision of a suspension bridge which holds a road above a river. Five HAMSTER unities are used: a UAV for inspecting bridge structure above the water, a UUV for inspecting bridge structure underwater, a UGV to inspect road asphalt, a floating control station on the river that functions as an access point between the UUV and the above water network and a pre-existing, fixed on the bridge, access point connected to the Internet. Each vehicle holds a camera used for bridge inspection. For simplicity purposes, main modules from each vehicle (such as GPS, battery sensor, inertial unit, moving actuators) are grouped in a single cluster.

Figure 12 presents the application modelling. Each vehicle is specialised from a specific version of HAMSTER unit. The floating control station is specialised from `CAGE` and the pre-existing bridge infrastructure is treated as a general HAMSTER entity.

Since all components are HAMSTER unities, the communication will take hold in a secure, seamless way among them. Each vehicle has a `navCluster`, a main HAMSTER cluster aggregating all sensors and actuators necessary for navigation and a `camera`, a mission-specific HAMSTER module that controls camera operation.

Figure 12 – Class diagram for HAMSTER architecture example.

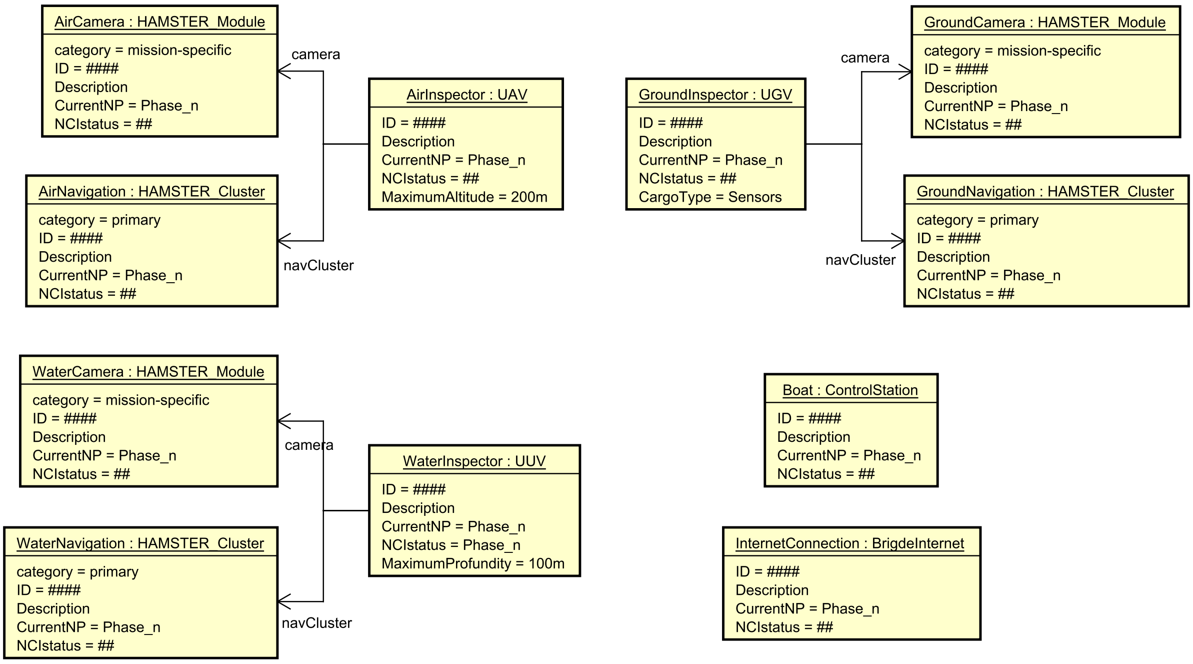


Source: Elaborated by the author.

Finally, Figure 13 presents the object diagram of the example. Each HAMSTER entity has one instance and, in case of vehicles, each one has also a navigation cluster and camera module. SPHERE and Navigation Phase Manager Platform instances were suppressed for readability purposes.

The detailed UML documentation is provided on Appendix A.

Figure 13 – Object diagram for HAMSTER architecture example.



Source: Elaborated by the author.

3.9 Final remarks

HAMSTER architecture provides an innovative way of connecting unmanned vehicles respecting their needs for mobility and heterogeneity, but also guaranteeing that there will be clear ways of implementing security and safety along with other desired modern features. The maturity achieved by HAMSTER during its development has led to the modularisation into four important and equally innovative platforms that can be individually explored in other systems as future work.

SPHERE platform provides well-defined ways of implementing security and safety on unmanned vehicles and systems to cover a majority of these systems models. Similarly, NIMBLE splits external communications into two units that help clarify the differences between ad hoc and infrastructured modes, prioritising approaches that take into consideration all their particularities.

On the field of energy efficiency, a small but promising contribution has been made with NP, which analyses the behaviour of navigation phases and tries to save power as much as possible, respecting the vehicle safety and security. Following a similar path, NCI proposal takes into consideration a set of characteristics and builds a trustworthy criticality index that can now be applied to communication, security, safety and any other application that one might find out relevant within an unmanned vehicle network.

In summary, HAMSTER is aligned with some of the most important aspects towards the achievement of more reliable and efficient unmanned aerial, ground and aquatic vehicles. It also supports the growing necessity of integration among heterogeneous vehicles to allow novel, complex, complete missions to be performed with guarantees required by governmental agencies.

Fundamentally, the study and development of HAMSTER architecture shall not end with this thesis. The architecture must be kept openly available for contributions by other researchers and developers around the globe[2]. Following chapters will carry out several case studies with experimental validations of each part of HAMSTER architecture pointing out how one can implement the provided ideas and proposals and how promising they can be.

[2] HAMSTER architecture is openly available at <www.lsec.icmc.usp.br/hamster>

CHAPTER 4

CASE STUDIES ON SPHERE

4.1 Chapter overview

Aiming at the global validation of HAMSTER architecture, several experimental case studies in each individual platform are developed. This chapter will go through several case studies on SPHERE platform for security and safety, described in Section 3.4. These results were partially published in: Silva *et al.* (2015), Pigatto *et al.* (2015) and Pigatto *et al.* (2016). The chapter organisation is: Section 4.2 presents the implementation of SPHERE's CSU authentication protocol using an embedded system prototype; Section 4.3 describes experiments and results obtained with Elliptic Curve Cryptography algorithms for SPHERE's CSU secure communications; and Section 4.4 provides methods to analyse safety with SPHERE's SMU.

4.2 Case study A: experiments on SPHERE's CSU authentication

This case study is a generic implementation of SPHERE's CSU authentication protocol.

4.2.1 Background

In this case study it is assumed that SPHERE's CSU is aware of the following information about each module: if it is on the list of authorised modules, the Internet Protocol (IP) address, the ID, the public key and the file descriptor. Each module (named Terminal for these experiments) is aware of its own identifier and pair of keys, IP, port and the CSU public key. All communication must be encrypted in order to ensure the authenticity of both parties, as it will be discussed in Section 4.3. Altogether, there are five operations on the SPHERE's CSU authentication protocol: Access Request; Information Request; Authentication Request; Notification; and Data Request.

The **Access Request** is the simplest and most important operation. On the vehicle initialisation, all primary modules must be authenticated to prevent fraud and ensure secure operations. It works as follows: the module sends a message to CSU containing its ID, which is double encrypted with the module's private key and the CSU's public key in this order. CSU receives and decrypts the message with its own private key and then the module's public key, extracting the ID value to be authenticated. If the decrypted ID is equal to the one associated to the IP address, then the module is authentic and may be granted with network access. Otherwise, permission is denied. Different approaches may be taken at this point, such as a mandatory authentication of main modules before UV operation and an optional authentication of secondary modules. Thus, groups of modules might be created, which leads to the next operation.

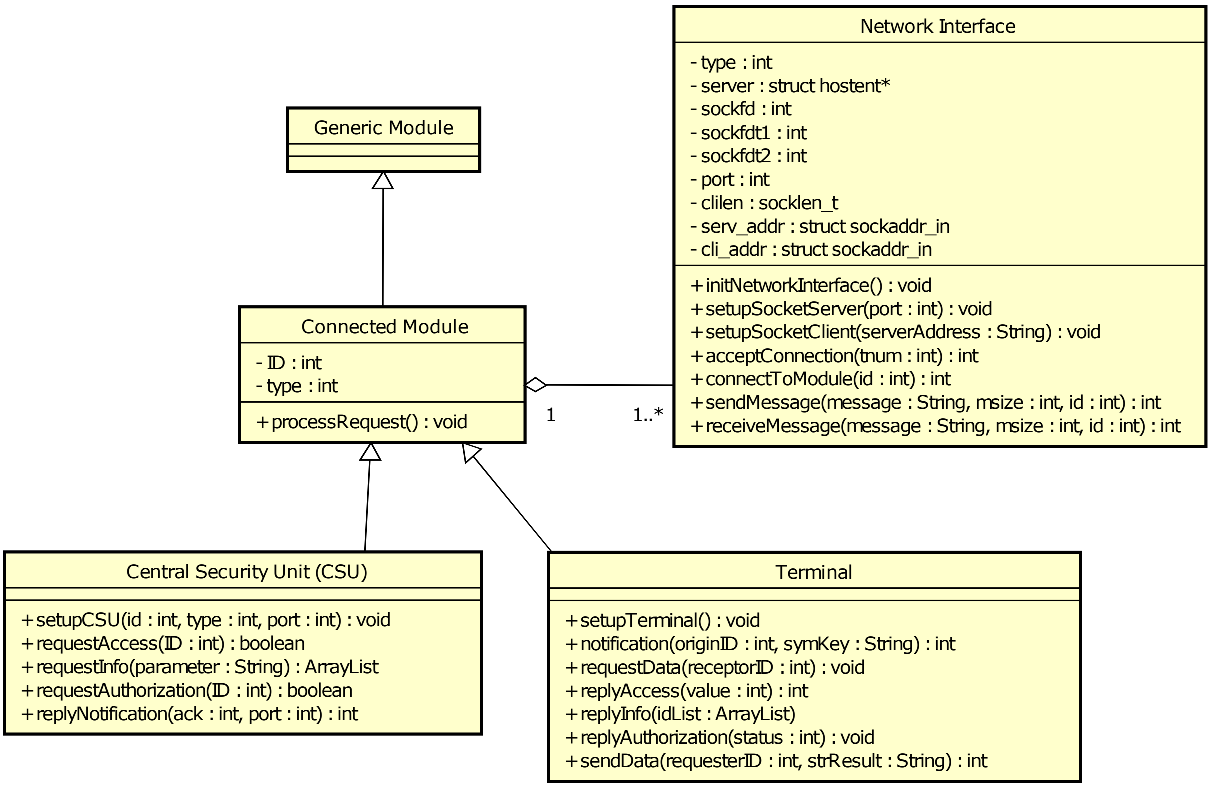
The **Information Request** allows a module to be aware of which modules are part of a category. For instance, a UAV is launched to collect aerial images of an area. The aircraft is set to fly over the region and begin to take photos as soon as it arrives on site. Such action can only be performed if the camera is aware of UV's location, demanding a communication with the GPS module. To meet this requirement, all modules must be previously categorised. The Information Request operation searches and returns a list of modules' IDs accordingly.

The **Authorisation Request** is sent by a module asking authorisation to communicate with other module. The **Notification** occurs during the Authorisation Request. CSU sends a Notification to the destination module, along with the public key of the module which has requested to communicate. The destination module starts a socket connection and returns a port number to CSU. In turn, CSU provides the IP address and the incoming port for the module that originally requested to communicate. Finally, a direct connection can then be established between modules, which is done by **Data Request** operation.

4.2.2 Material and methods

According to the UML class diagram used for the authentication protocol implementation, as seen in Figure 14, CSU and Terminal are subclasses of *Connected Module*, which in turn is a subclass of *Generic Module*. Both CSU and Terminal can process requests by invoking *processRequest* method. However, only CSU attends access requests (*requestAccess CSU* method) and only the Terminal will perform a set of activities after getting a notification (*Notification* method) from CSU. All *Connected Modules* have network interfaces that keep file descriptors and data types required by operations.

Figure 14 – Class Diagrams of SPHERE's internal modules.


Generic Module
Network Interface
- type : int
- server : struct hostent*
- sockfd : int
- sockfdt1 : int
- sockfdt2 : int
- port : int
- clilen : socklen_t
- serv_addr : struct sockaddr_in
- cli_addr : struct sockaddr_in
+ initNetworkInterface() : void
+ setupSocketServer(port : int) : void
+ setupSocketClient(serverAddress : String) : void
+ acceptConnection(tnum : int) : int
+ connectToModule(id : int) : int
+ sendMessage(message : String, msize : int, id : int) : int
+ receiveMessage(message : String, msize : int, id : int) : int
Connected Module
- ID : int
- type : int
+ processRequest() : void
1
1..*
Central Security Unit (CSU)
+ setupCSU(id : int, type : int, port : int) : void
+ requestAccess(ID : int) : boolean
+ requestInfo(parameter : String) : ArrayList
+ requestAuthorization(ID : int) : boolean
+ replyNotification(ack : int, port : int) : int
Terminal
+ setupTerminal() : void
+ notification(originID : int, symKey : String) : int
+ requestData(receptorID : int) : void
+ replyAccess(value : int) : int
+ replyInfo(idList : ArrayList)
+ replyAuthorization(status : int) : void
+ sendData(requesterID : int, strResult : String) : int

Source: Elaborated by the author.

Each of the operations described in Subsection 4.2.1 produces two messages paths: the request and the response. The first four bytes identify the message type and will determine how the rest of the message will be interpreted. There are ten codes defined by the following macros: REQACC, REQINFO, REQAUTH, REQDATA, NOTIFY, REPACC, REPINFO, REPAUTH, SNDDATA, REPNOTIFY. Table 3 shows message codes, type and procedure description.

Table 3 – Message codes, types and associated procedures.

Message code	**Type**	**Procedure**
REQACC	request	Access
REQINFO	request	Information
REQAUTH	request	Authorisation
REQDATA	request	Data
NOTIFY	request	Notification
REPACC	response	Access
REPINFO	response	Information
REPAUTH	response	Authorisation
SNDDATA	response	Data
REPNOTIFY	response	Notification

Source: Elaborated by the author.

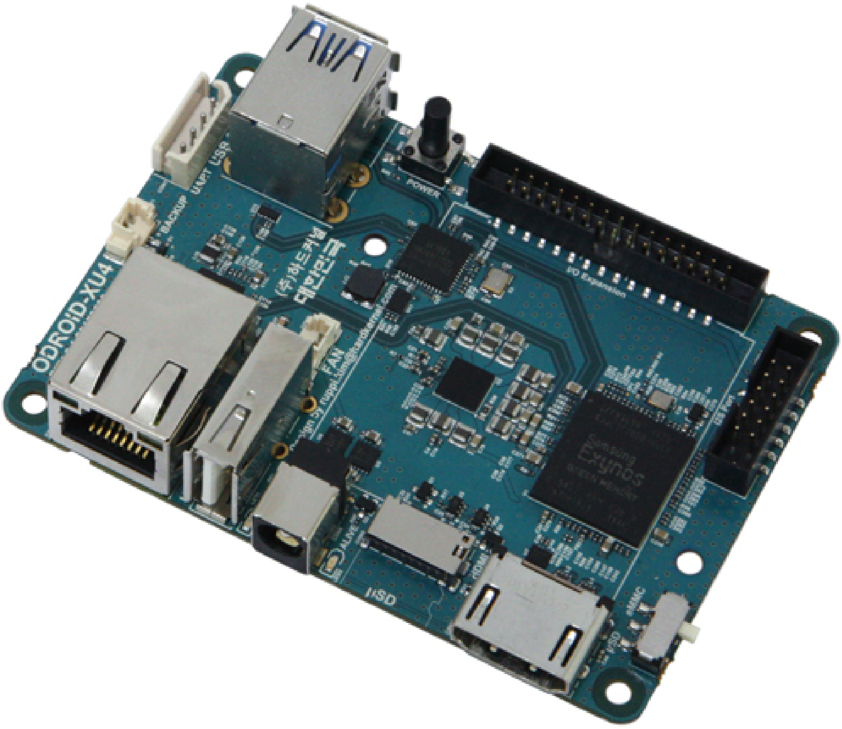
The structure of each message is described as follows:

- REQACC is followed by a 4-byte integer containing the Terminal ID to be authenticated. The total size is 8 bytes;
- REQINFO is followed by a 4-byte integer identifying the type or requested module category. The total size is 8 bytes;
- REQAUTH is followed by a 4-byte integer containing the Terminal ID to which a communication is requested to be established with. Fixed size of 8 bytes;
- REQDATA is followed by a 4-byte integer with the size of requested information, which is used in the following step to perform the action. The total size is variable;
- NOTIFY is followed by a 4-byte integer with the terminal ID that requested to establish a communication. Then, a fixed 32-byte private key is shared. Fixed size of 40 bytes;
- REPACC is followed by a 4-byte integer with the response. It may vary from OK (authorised) or NOK (not authorised). Fixed size of 8 bytes;
- REPINFO is followed by a 4-byte integer with the quantity of identifiers that belong to a category. This quantity times 4 is the amount of bytes that composes the rest of the message. The total size is variable;
- REPAUTH is followed by a 4-byte integer with the destination Terminal state code. If its status is OK, then 4 bytes with the IP number are sent, followed by 4 bytes containing the port number and more 32 bytes with the symmetric key. The total size may vary between 8 and 48 bytes;
- SNDDATA is followed by a 4-byte integer with the size of information to be returned, followed by such size times 4 bytes for the requested information. The total size is variable;
- REPNOTIFY is followed by a 4-byte integer with the module state. If OK, then 4 bytes are sent containing the port number to connect. Otherwise, the message is ended. The total is variable from 8 to 12 bytes.

These experiments were performed using a laptop computer to run the Terminals and an ODROID-XU4[1] to run the SPHERE's CSU. A picture of an ODROID-XU4 is provided in Figure 15.

[1] ODROID-XU4 is powered by ARM big.LITTLE technology, an Heterogeneous Multi-Processing (HMP) solution.

Figure 15 – ODROID-XU4 features an octa-core Exynos 5422 big.LITTLE processor, advanced Mali GPU and Gigabit ethernet.

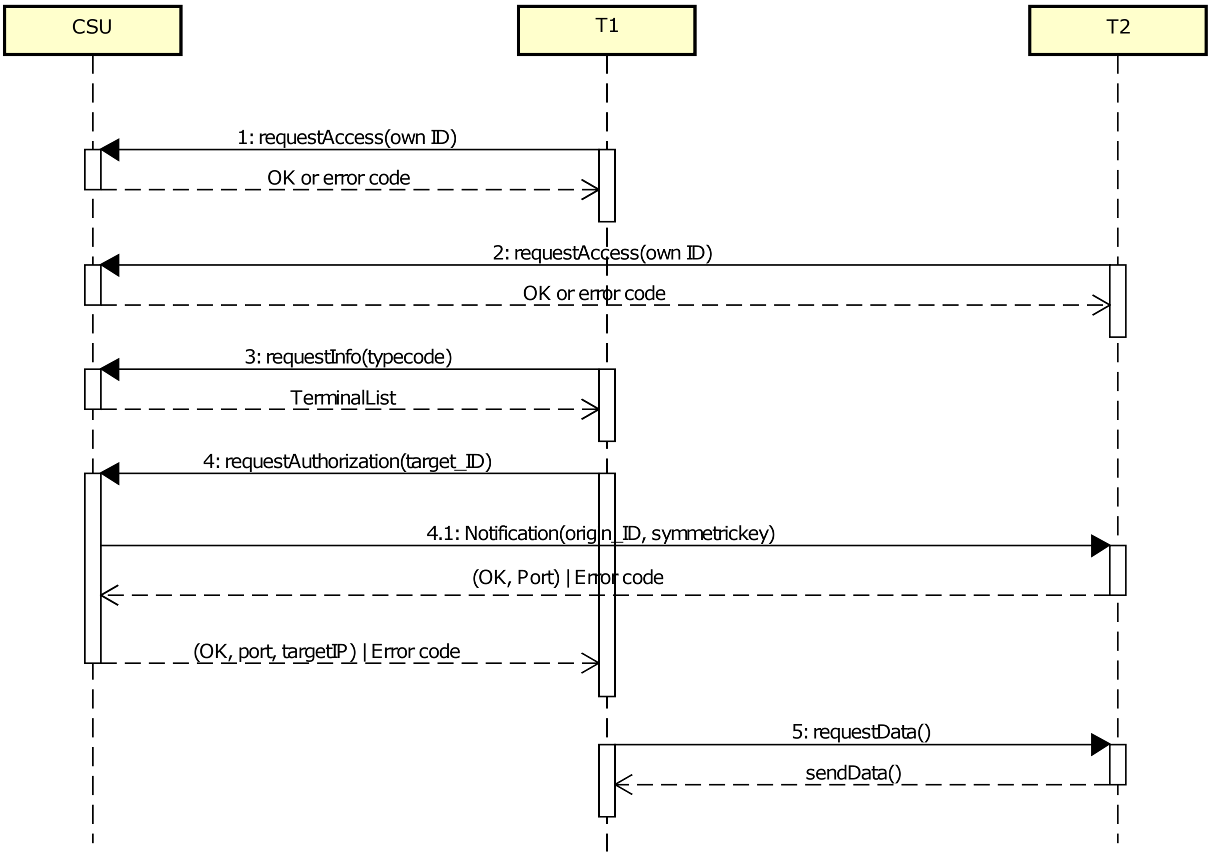


Source: Elaborated by the author.

4.2.3 Results and discussions

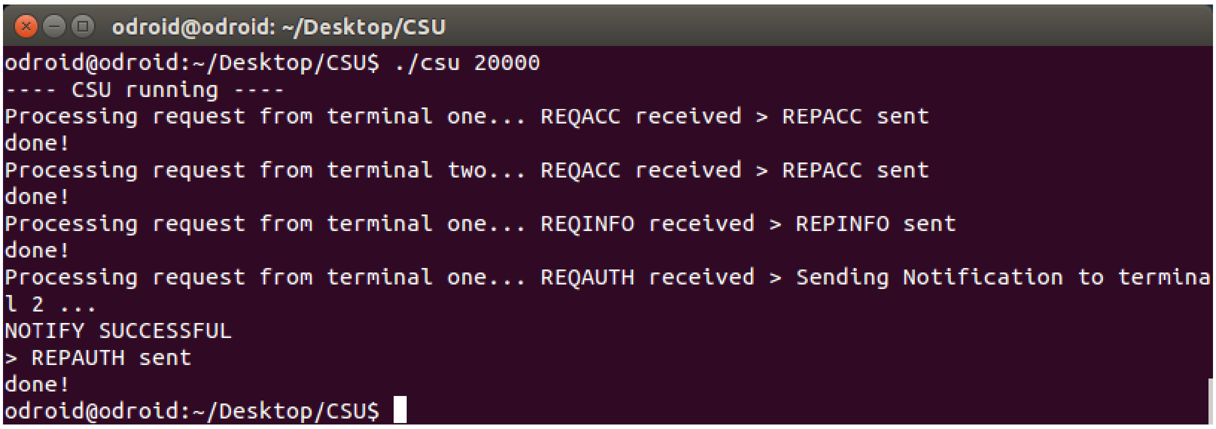
The results are related to the actions shown in the sequence diagram in Figure 16. The goal is to represents all the protocol operations by making Terminal T1 communicate with Terminal T2. First, both T1 and T2 authenticate with CSU by using REQACC and REPACC. T1 requests information to CSU with REQINFO and REPINFO in order to find out about the existence of T2. Next, T1 requests authorisation via REQAUTH to the CSU, which notifies T2 using NOTIFY and REPNOTIFY messages, followed by a REPAUTH. Finally, T1 establishes a connection with T2 and they can communicate with REQDATA and SNDDATA. Figures 17, 18 and 19 present the execution of CSU, T1 and T2 commands, respectively.

Figure 16 – Sequence diagram used to perform experiments.



Source: Elaborated by the author.

Figure 17 – CSU running on an ODROID-XU4 (IP: 10.70.1.232 / Port: 20000).

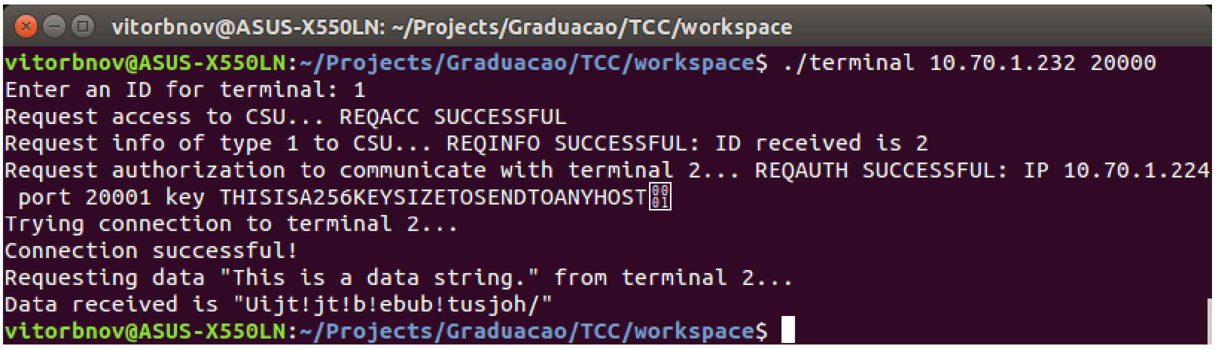


```
odroid@odroid:~/Desktop/CSU$ ./csu 20000
---- CSU running ----
Processing request from terminal one... REQACC received > REPACC sent
done!
Processing request from terminal two... REQACC received > REPACC sent
done!
Processing request from terminal one... REQINFO received > REPINFO sent
done!
Processing request from terminal one... REQAUTH received > Sending Notification to termina
l 2 ...
NOTIFY SUCCESSFUL
> REPAUTH sent
done!
odroid@odroid:~/Desktop/CSU$
```

Source: Elaborated by the author.

Although SPHERE's CSU authentication protocol provides a way of verifying modules authenticity, light approaches for providing secure communications must also be implemented by SPHERE's CSU. Next section will carry out an evaluation of a cryptographic algorithm that meets embedded systems requirements.

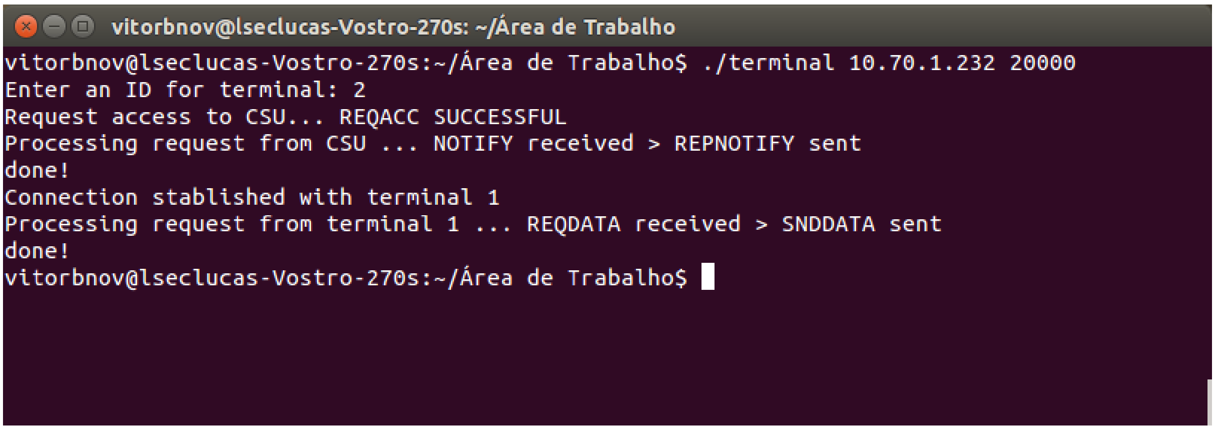
Figure 18 – Terminal 1 running on a laptop (IP: 10.70.1.173).



```
vitorbnov@ASUS-X550LN:~/Projects/Graduacao/TCC/workspace$ ./terminal 10.70.1.232 20000
Enter an ID for terminal: 1
Request access to CSU... REQACC SUCCESSFUL
Request info of type 1 to CSU... REQINFO SUCCESSFUL: ID received is 2
Request authorization to communicate with terminal 2... REQAUTH SUCCESSFUL: IP 10.70.1.224
 port 20001 key THISISA256KEYSIZETOSENDTOANYHOST▯
Trying connection to terminal 2...
Connection successful!
Requesting data "This is a data string." from terminal 2...
Data received is "Uijt!jt!b!ebub!tusjoh/"
vitorbnov@ASUS-X550LN:~/Projects/Graduacao/TCC/workspace$ 
```

Source: Elaborated by the author.

Figure 19 – Terminal 2 running on a laptop (IP: 10.70.1.224 / Port: 20001).



```
vitorbnov@lseclucas-Vostro-270s:~/Área de Trabalho$ ./terminal 10.70.1.232 20000
Enter an ID for terminal: 2
Request access to CSU... REQACC SUCCESSFUL
Processing request from CSU ... NOTIFY received > REPNOTIFY sent
done!
Connection stablished with terminal 1
Processing request from terminal 1 ... REQDATA received > SNDDATA sent
done!
vitorbnov@lseclucas-Vostro-270s:~/Área de Trabalho$ 
```

Source: Elaborated by the author.

4.3 Case study B: evaluation of Elliptic Curve Cryptography for SPHERE's CSU secure communications

This case study provides guidelines for the development of SPHERE's CSU secure communications. Real experiments were performed aiming at providing security on all IMC, M2M and M2I communications.

4.3.1 Background

Kumar and Kumar (2008) performed the study of a protocol for the exchange of keys on a mobile ad hoc network. It was stipulated that the network would be composed by three layers. A military scenario was chosen in which the first layer is composed of nodes that communicate with a central unit through the backbone. In the example, nodes are considered soldiers sending information and the backbone is a vehicle with more computational power. The second level consists on several backbones that compose a wireless network. Finally, the third level is a UAV flying in the area of the backbone, helping to centralise the network.

Eissa, Razal and Asringadi (2009) focused on creating an authentication mechanism in MANETs. For this, four different keys are generated: an identity key, a public key, a private key and a symmetric key. The first step is to check the level of confidence of every node. For this, neighbouring nodes are consulted using the identification key. If at least *n* nodes confirm the confidence level of the others, the communication starts. Both nodes agree on a session key using the public key and hold in their databases of reliable keys. Thereafter, such nodes use their private key to encrypt messages in communications. As a result, cryptanalysis attacks do not work as it is necessary to have a public key.

Faughnan *et al.* (2013) focused on kidnapping of UAVs. The method is divided into two parts. The first one is the risk identification of an attack to the UAV. To perform this step, a list of risk scenarios was created. The second one is the creation of a mechanism to inform the system operator that the UAV is under attack. There are two systems embedded on the aircraft measuring system speed. If there is great variation in the speed measurement, this may mean that the system is under attack. For the tests, it was used a car to simulate the UAV. The result was a framework able to detect attacks through the described process.

Kashikar and Nimbhorkar (2013) studied the exchange of messages among nodes in a MANET. As it is done by exchanging a data packet at a time, such networks are affected by DoS attacks. The proposed method aims to block malicious nodes to access the network. If the target node of the attack begin receiving packages in quantities larger than the network supports, such node will prompt the attacker node to reduce its transmission. If it does not, the communication among nodes is stopped and the target node will put the attacker node in a list of unreliable identifications. As a node attempts to join the network, the nodes belonging to the network check their lists of unreliable identifications. If in any of these lists, it will not get access.

Faughnan *et al.* (2013) and Kashikar and Nimbhorkar (2013) have shown the possible attacks a UAV might face during operations. Javaid *et al.* (2012) aimed the creation of secure channels for communication among UAV systems, satellites and base stations. After a test with attacks, the system had some failed components, especially after the DoS attack.

Man *et al.* (2009) carried out a study on health and safety monitoring in UAV systems. As a basis for the study, a model with the main components of an aircraft was designed. Such components have been grouped according to their functions in the UAV system. Thus, in case of errors in a module, the path of propagation of this error can be noticed. In addition, the author addressed some techniques to predict when the modules might begin to malfunction, based on the quality of data and experience regarding the use of UAVs. The paper presents no results; however addresses important concepts related to the health of the UAV.

Raj, SelvaKumar and Lekha (2011) studied a protocol for nodes admission to a network in a decentralised manner. At the start it was assumed that the network is composed by trusted nodes only. This group of nodes owns a shared secret key. To join the network, a node must request and receive permission from all nodes using a secure communication channel. If a node is approved, the other nodes create a new shared secret key that will be used for communications between pairs of nodes.

The traditional cryptographic algorithms are usually enough for most of the computing applications. The asymmetric RSA (Rivest Shamir Adleman), for instance, is well tested and sometimes considered a synonym of public-key cryptography. However, critical embedded systems with strict limitations would work at better conditions if smaller cryptographic keys were used, still providing high reliability to the results. The investigation performed in this case study is the eligibility of using a different public-key algorithm to replace RSA on SPHERE's CSU secure communications. Specifically, the ECC (Elliptic Curve Cryptography) was chosen instead of RSA since it can present several advantages for computational systems with restricted resources.

4.3.2 Material and methods

The three accepted encryption schemes are based on three mathematical problems (JENA; JENA, 2011): integer factorisation problem, discrete logarithm problem and elliptic curve discrete logarithm problem. The latter offers a higher level of security, since it operates with smaller key sizes in comparison to RSA and DSA (Digital Signature Algorithm). To achieve an appropriate level of security, RSA and DSA should use 1024-bit key size based on the time needed to break their cipher code, while ECC needs to operate with only 160-bit keys to provide the same level of security.

Besides the much smaller key size, ECC algorithm has specific advantages, such as the fact that only exponential-time attacks may be applied if the curve is carefully chosen. Even if factoring and multiplicative group discrete logarithms are broken, the elliptic curve discrete logarithm can still be difficult to compute (JENA; JENA, 2011). Establishing a comparison, the security level of an implementation of elliptic curves with 160-bit key is equivalent to RSA 1024-bit key size (LENSTRA; VERHEUL, 1999).

Two algorithms were developed in this case study, both published in Pigatto, Silva and Branco (2012). The method chosen for the implementation of ECC is El-Gamal, that combines the properties of El-Gamal elliptic curve encryption method of exchanging messages. Its operation is given as follows: two users must share the same elliptic curve and a point P. Each one must choose a random number that acts as its private key and multiply the known point, obtaining aP, which becomes its public key. At the beginning of the communication, the public key is transmitted to the other user, which has bP as public key and thus the private key b. For the exchange of messages the user needs to multiply its own private key by the public key of the other user, obtaining $b(aP)$, and then add this result to the message encoded in an M number. Therefore, the message will be $M + b(aP)$. When the message is delivered to the receiver, it will be able to decode it by multiplying its own private key by the other user's public key, $(a(bP))$, and subtracting the total content, $M + b(aP) - a(bP) = M$.

In the implementation, two libraries were used as tools to perform mathematical operations: MIRACL (Multiprecision Integer and Rational Arithmetic C/C++ Library) due to its performance as reported by Ramachandran, Zhou and Huang (2007) and RELIC 0.2.3 (Library for Efficient Cryptography) developed by Aranha and Gouvêa (2011). The MIRACL library produced by Shamus Software is proprietary, but free for educational use. It is intended to behave as a tool for developers of encryption systems and offers the necessary operations to handle large numbers and a full support for elliptic curves. RELIC, on the other hand, has been developed by researchers at University of Campinas (UNICAMP) in order to provide cryptographic tools based on flexibility and efficiency. It has implementations of large integers arithmetic, binary and prime fields arithmetic, elliptic curves over prime fields, among others.

The algorithm developed with MIRACL library operates on fixed size blocks of 18 characters and the algorithm based on RELIC library operates on blocks of 40 characters. Parameters that define the elliptic curve and the point used in common by the users are fixed. These definitions have been established according to some experiments that have proven their efficiency while operating with these block sizes.

The algorithms were developed in C language and run through three actions that must be informed as parameters: key creation, encryption and decryption. The latter two also require input and output files paths. The implementations of the two algorithms have similar structures, containing four main functions responsible for creating keys, encrypting, decrypting and calling other functions. In addition, both algorithms have defined structures for the public and private keys.

The function responsible for creating the keys first generates a random integer as the private key and then multiplies a known point of the curve by that number to generate the public key. After being generated, keys are stored in two different files to be exchanged over the network, if necessary.

The function that encrypts a text block starts transforming the message in a point on the curve. This is the main difference between the developed algorithms in this case study. The MIRACL library maps a number of bytes, but it does not work when there is a NULL byte. It is therefore necessary to previously treat the text block and indicate the places where this byte is present. After all these actions, it is necessary to map the sequence of bytes in a number. From this number it is possible to find the point that is part of the curve where the x-axis is closer to that number.

RELIC library demands a mapping of bytes sequence, with no previous treatment for numbers. Through transactions between the number and the structure of a known point of the curve, the sequence of bytes is transformed into a point.

Using the messages mapped at points, the encryption is performed by multiplying the private key with the receiver's public key and adding to the point of the message. In contrast, the function that decrypts the encrypted block subtracts the multiplication of the private key by the sender's public key, obtaining the message encoded at a point. After the reverse operation, it is possible to obtain the original sequence of bytes.

The function that calls other functions simply reads the parameters of the execution and operations are defined according to the action. When the action is to generate keys, an appropriate function is executed. However, for encryption and decryption, input and output operations must be performed. In encryption, it is necessary to read the bytes from the original file and write the encrypted blocks at points in an output file. In decryption, the program reads the points of the encrypted file and writes the byte sequences in the output file, restoring the original one.

4.3.3 Results and discussion

The experiments were set up according to techniques for performance evaluation of computing systems (JAIN, 1991). A variety of terms, such as response variable, factors and interaction levels is used during the stages of design and analysis of experiments. Response variable represents the result (output) of an experiment and is often the variable selected to measure the system's performance. Factors are the variables that affect the system response; levels are the values that a factor may assume; and interaction indicates the dependency between the factors evaluated (JAIN, 1991).

The first steps were the definitions of a response variable to be evaluated, the amount of replication required for the experiments and the testing environment. The environment used to run the tests was a Pentium Dual-Core CPU T4300 2.10 GHz with 2 GB of RAM and Linux Ubuntu operating system. To evaluate the efficiency of encryption and compare the results of the developed algorithms, the response variable selected was the average response time. The whole process performed in the experiments is the encryption and decryption of each input file. Each experiment was performed 15 times, ensuring a statistical validation since there was no large standard deviation among results.

There are a few ways to accomplish the design of experiments. In this case study, we have used the full factorial design (JAIN, 1991). In this type of planning, all combinations are used considering all factors and levels. Thus, it is possible to evaluate all factors,

Table 4 – Combinations of experiments.

Exp.	Library	Key size (bits)	Message size (kilobytes)
1	MIRACL	160	50
2	MIRACL	160	100
3	MIRACL	256	50
4	MIRACL	256	100
5	RELIC	160	50
6	RELIC	160	100
7	RELIC	256	50
8	RELIC	256	100

Source: Elaborated by the author.

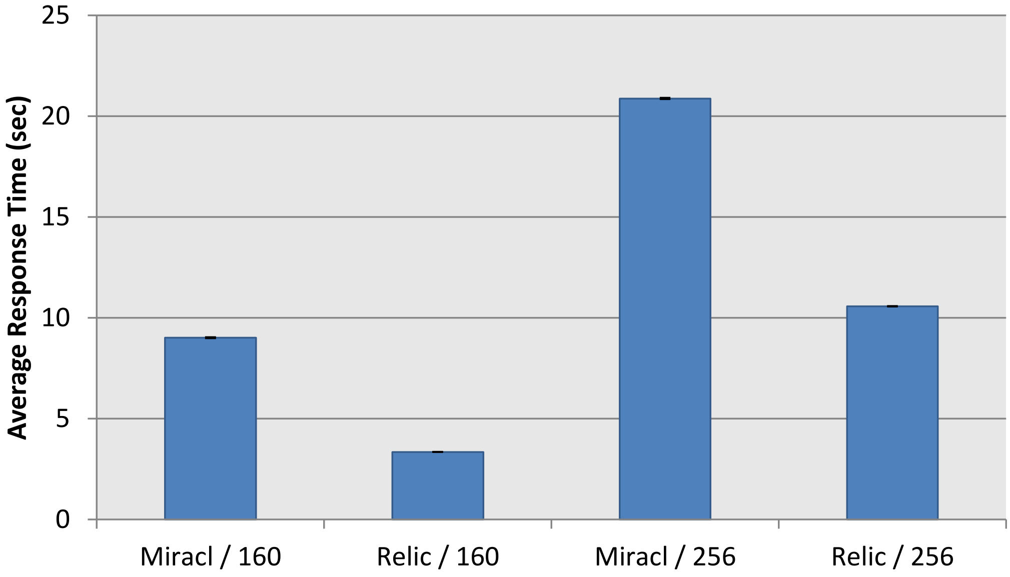
determine the effect of each factor on the experiments and verify the interactions between them. Table 4 shows the possible combinations of the experiments. The first factor is the library used, which has two levels: MIRACL and RELIC. The second factor is the message size, which may vary between 50 and 100 kilobytes (KB). Usually, command messages will be within this range. Finally, the third factor assumed is the key size, also with two variations: 160-bit and 256-bit. These key sizes are used as equivalent to RSA 1024 and 2048-bit, respectively, which is the recommended key size for majority of applications (BAKER, 2006). Therefore, it is possible to generate eight different combinations for the experiments.

Figure 20 shows the comparison between the average response times achieved by each of the algorithms run in the first message size (smaller), considering the two key sizes. The algorithm based on MIRACL library has a considerably higher time than the one based on RELIC, in both cases. The times obtained are approximately 9 seconds (MIRACL) and 3.3 seconds (RELIC), in which the key size is 160-bit. When using 256-bit key size, the times are 20.9 seconds (MIRACL) and 10.6 seconds (RELIC).

Figure 21 shows the second comparison chart with the performance of algorithms to encrypt and decrypt the second message size (larger). There was a natural elevation in the response time due to increased data to be processed while maintaining the same characteristics of the previous comparison. The times obtained are approximately 18.1 seconds (MIRACL) and 6.6 seconds (RELIC), in which the key size is 160-bit. When using 256-bit key size, the times are 41.7 seconds (MIRACL) and 21.1 seconds (RELIC).

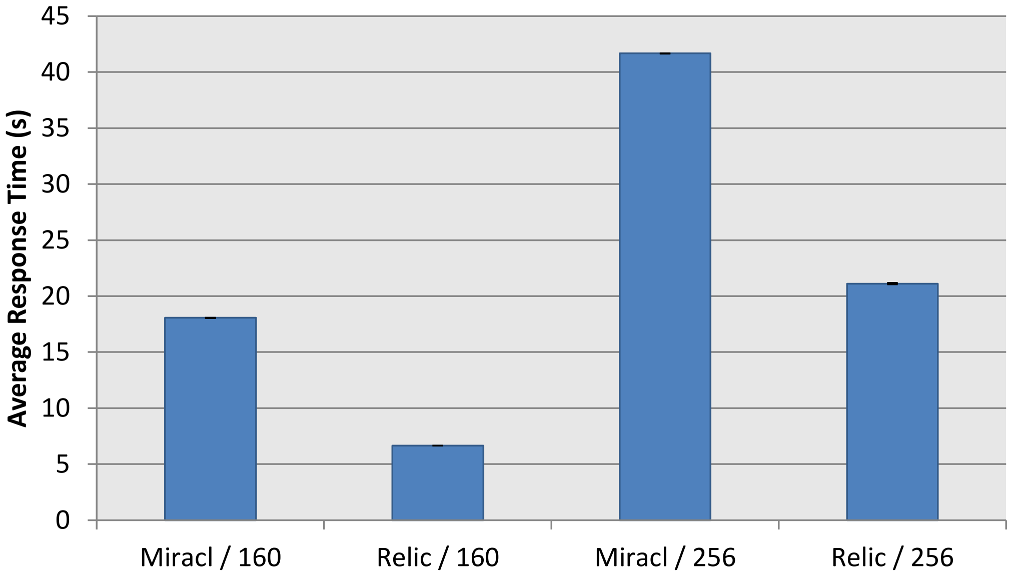
The charts show a better performance of the algorithm based on RELIC in both cases (Figures 20 and 21). The percentage of influence of each factor of the performance evaluation was calculated, as well as the influence of the associated factors over the response time. The chart in Figure 22 shows that factor A (algorithm) exerted a 28% influence on the results, which is relevant to the comparison presented. Factor B (key length) exerted

Figure 20 – Comparison between MIRACL and RELIC libraries with the first message size (50 KB).



Source: Adapted from Silva *et al.* (2015).

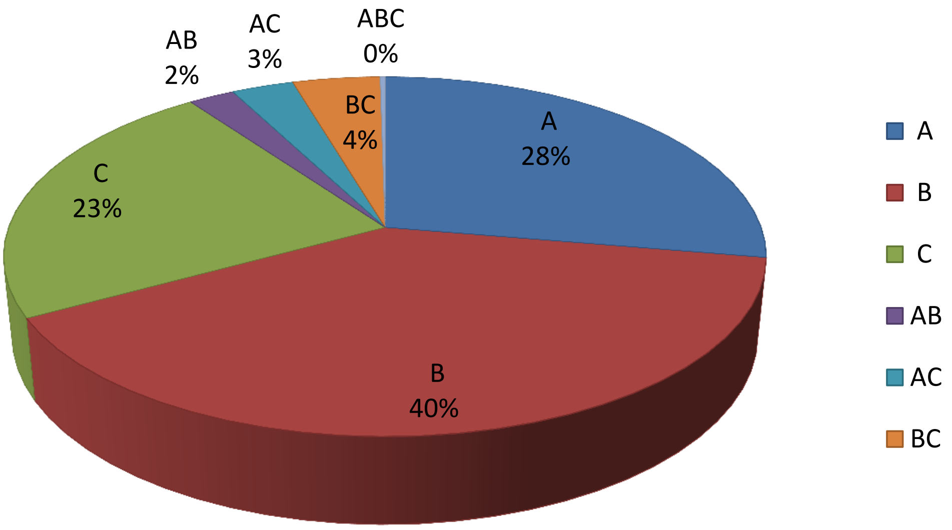
Figure 21 – Comparison between MIRACL and RELIC libraries with the second message size (100 KB).



Source: Adapted from Silva *et al.* (2015).

a greater influence on the response variable, with a total of 40%. Factor C (message size) influenced the results in 23%. The associated factors exerted small influences: BC influenced 4%, AC only 3%, AB only 2% and ABC associated exerted no influence.

Figure 22 – Influences of each factor on the response time (A - Algorithm; B - Key Size, C - Size of message).



Source: Adapted from Silva *et al.* (2015).

These results have shown that the message size influences the outcome of the application due to the difference in the amount of data to be encrypted. However, the aim of this case study was to show that the influence of the algorithm used is quite considerable. As the response time is crucial for critical embedded systems that often work with real-time tasks, RELIC is more suitable for the implementation of the ECC algorithm, considering the conditions of the environment used for the experiments. It was also possible to identify several variations when the experiments were conducted in an environment with similar characteristics to a critical embedded system.

It is important to notice that, initially, the time obtained may be classified as a big issue. However, considering that the assumed key size is relatively large and critical embedded systems require the application of cryptography in most cases to send short commands, such as changing routes or missions, the performance presented meets the expectations, obtaining very short response times, which may be considered a solution for real-time systems, the focus of this work. It is also important to point out that these algorithms should be applied in association with symmetric key algorithms to significantly reduce response times. However, ECC implemented in hardware could achieve considerably better performance if compared with software, being more applicable to embedded systems in general.

4.4 Case study C: measuring safety on avionics on-board wireless networks with SPHERE's SMU

This case study is related to SPHERE's SMU. Safety is a very important concern of unmanned vehicles and requires exclusive approaches to be fully addressed. This case study investigates safety issues related with security vulnerabilities and threats in UAVs, following the RTCA/DO178B (RTCA, 1992) and DO178C (RTCA, 2011) standards related to UAV security and safety.

4.4.1 Background

The availability of ubiquitous connection with higher transmission rates has contributed to the growing numbers of safety problems as scams, worms, denial of service attacks, etc. Safety almost always involves additional costs; these are costs that do not yield direct returns, which must always be justified with regards to the financial world. Risk management automatically generates direct reasons for such recommendations in terms of safety. Several kinds of attacks can be triggered, taking advantage of the vulnerabilities or failures of system components or communications systems. Identify possible failure points of critical communications systems, which may generate such vulnerabilities and be hence exploited, seeking to compromise the integrity of the communications system, can avoid the safety issues.

Future avionics safety applications based on A2A and A2I communications are aimed at reducing the number of fatal accidents, leading to a new era of air traffic safety. Meanwhile, new security requirements must be revised and considered as a manner to prevent attacks to the inner side of such systems.

Modern aircraft are usually equipped with a high amount of sensors and actuators. That may be observed in wireless approaches for aircraft internal communications which benefit from clustering techniques to reduce the communication traffic by grouping sensors and actuators. Then, local analysis of sensor data within small clusters of nodes can be carried out, allowing the extraction of relevant data features locally (TOVAR *et al.*, 2012).

There has not been much concern regarding the inner connections of aircraft, since components are connected to various communication buses, which are hardly accessed from the outside of the aircraft. However, the introduction of wireless approaches is changing this scenario, e.g. the security and privacy concerns in fly by wireless systems (SAMPIGETHAYA *et al.*, 2011). The on-board electronics are threatened by attacks originated from both inside and outside the aircraft.

The basis for secure and safe deployment of A2A and A2I communications relies on trusted elements, secure storage of secret keys and trustworthy communications within aircraft. However, when it comes to the internal aircraft communication (IAC), there is a strong need to protect components allegedly relevant for the overall system security against tampering. Moreover, sensitive data must also be protected from unauthorised changes (AKRAM *et al.*, 2015). A thorough analysis of security and safety requirements is relevant to determine security measures that are effective and cost-effective.

There are two kinds of standards to consider for unmanned aerial vehicles safety and security: i) process standards describe the development processes to be followed to ensure that the finished product is written in a safe (RTCA, 1992; RTCA, 2011) and/or a secure manner (ISO14508, 2006); ii) coding standards describe a high-level programming language subset that ensures the software is written as safely (MISRA, 2004) and securely (DEFENSE, 2007) as possible. Safety is clearly important in UAV development, but a UAV can only be considered safe if it cannot be controlled by a hostile intruder.

This case study outlines a security requirements analysis applied to UAVs internal networks equipped with A2X[2] communication interfaces and associated with safety issues. This approach uses SPHERE's SMU Safety control. Moreover, the research published in Henniger *et al.* (2009) was an inspiration for this assessment.

4.4.2 The target system

This subsection details the target system which is going to be investigated in this research. The system is a generic unmanned aircraft system consisting of ECUs (Electronic Control Units), sensors and actuators connected to each other via several buses.

4.4.2.1 Network architecture

HAMSTER is used to define the UAV on-board network architecture shown in Figure 23. More details about a UAS architecture can be find in Marconato *et al.* (2014). The communication control unit and the mission unit are able of communicating to the outside via dedicated interfaces, e.g. wireless interfaces for A2X communications, such as NIMBLE platform (detailed in Section 3.7).

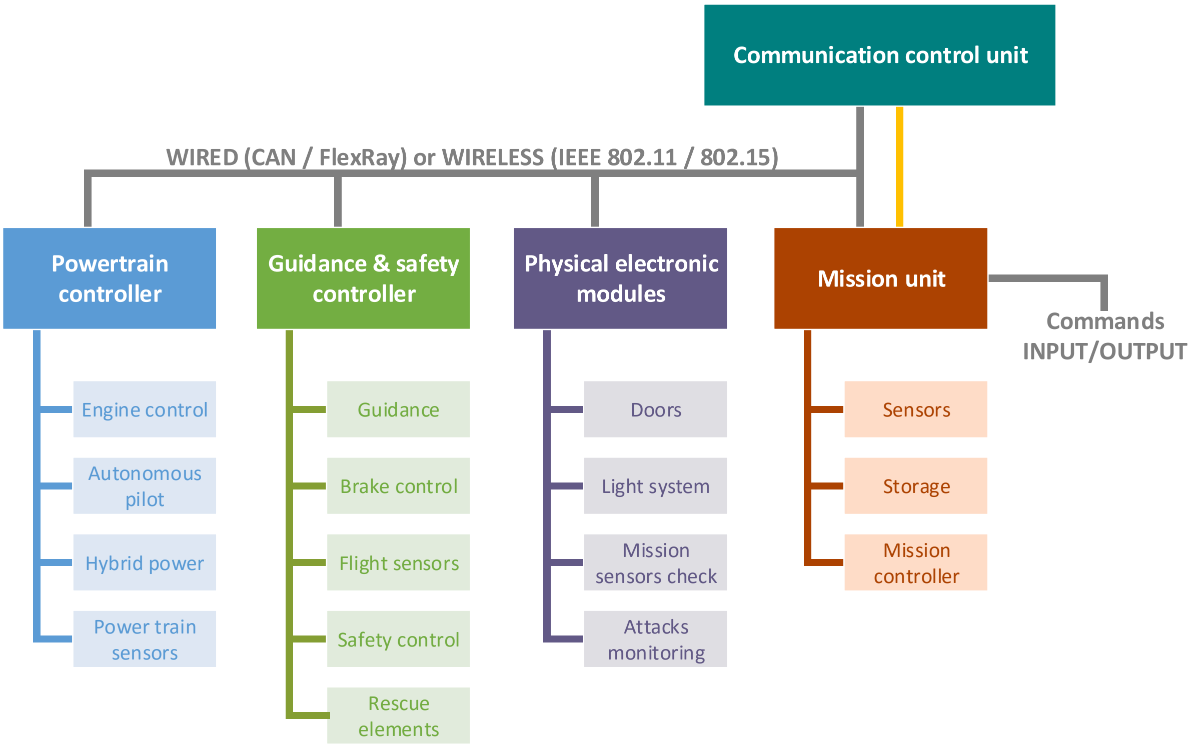
Frequently, avionics on-board systems operate in an uncontrolled environment, exposed to a variety of threats, against which their assets must be protected.

4.4.2.2 Assets

The main components of an avionic on-board network that may become targets of attacks are: i) on-board electronic components, such as ECUs, sensors and actuators; ii) the communication links among components and within ECUs, specially if these are wireless links; and iii) the software running on ECUs.

[2] A2X will be used to refer to A2A and A2I communications in this case study.

Figure 23 – The assumed UAV on-board network architecture.



Source: Elaborated by the author.

4.4.2.3 Use cases

Use cases to describe a system's behaviour as it responds to various stimuli from the outside were chosen. The following are considered as use cases, covering a range of future avionics functions with possible security implications: i) A2A communication; ii) A2I communication; iii) new UAV or mission-specific modules; and iv) other external elements that might be connected to the UAV.

Examples of A2A communication use cases can be pointed out:

- In a swarm, if a UAV identifies an obstacle that leads to changes in routes and may be the case of others UAVs, an emergency notification should be broadcast including accurate position data;

- On the other hand, if a UAV receives an emergency notification as previously described, then it should first check the received information and compare it with its own information about position, route and anything else that matches the situation. If it is a recognised dangerous situation, then the UAV must avoid the obstacle.

4.4.3 Threats

A comprehensive list of threats which are potential motivations for attacks on avionics on-board networks, which may affect the safety, can be stated as: i) actions that aim to gain advantages (e.g. identity or information theft); or ii) simply make UAVs unusable, steal UAVs, or harm others (e.g. people, animals).

A list of attacks can be extensive. It can range from jamming the wireless communication over replaying wireless messages to manipulation of input sensor data and/or changing of control parameters in the engine control unit. An exhaustive list was presented by Javaid *et al.* (2012).

4.4.4 Security issues in fly by wireless

Due to the fact that fly by wireless (DANG *et al.*, 2012) paradigm is strongly based on wireless networking, it is vulnerable to most of the known attacks to such type of network. However, as it introduces a new paradigm and is inserted in an even more critical environment, some exclusive attacks may be faced. In this subsection, the major attacks to fly by wireless concerning security that may also affect safety, are introduced. We also point out the common countermeasures that have been currently applied.

4.4.4.1 Physical layer

- **Jamming attack**: Jamming is a well-known attack to physical layer. It interferes with the radio frequencies used by network's nodes (Elaine Shi; PERRIG, 2004; MODARES; SALLEH; MORAVEJOSHARIEH, 2011; KHAN; PATHAN; ALRAJEH, 2012; DENER, 2014). The attacker sequentially transmits over the wireless network refusing the underlying MAC protocol. Jamming attack can interrupt the network if a single frequency is used throughout the network. In addition, jamming may cause excessive energy consumption at a node by injecting impertinent packets. The receiver's nodes will as well consume energy by getting such packets (MODARES; SALLEH; MORAVEJOSHARIEH, 2011). Typical defences against jamming include variations of spread-spectrum communication, such as frequency hopping code spreading (KHAN; PATHAN; ALRAJEH, 2012; JINDAL; MAINI, 2014).

- **Tampering attack**: In some situations, an adversary can physically tamper nodes. A tampering attacker may damage, replace and electronically 'interrogate' the nodes to acquire information (SALEEM; ULLAH; YOO, 2009; SASTRY; SULTHANA; VAGDEVI, 2013). If a physical access is given to a node, an attacker can draw sensitive information, such as cryptographic keys or other data on the node. The node may also be altered or replaced to create a compromised node controlled by the attacker. Tamper-proofing the node's physical package is one of the defences to this attack (JAIN; KANT; TRIPATHY, 2012; KHAN; PATHAN; ALRAJEH, 2012; DENER, 2014).

4.4.4.2 Data Link layer

- **Collision attack**: A collision occurs when two nodes attempt to simultaneously transmit on the same frequency. It results in packets disruption either completely or partially, which will cause an erroneous data transmission through a communication channel (DENER, 2014; SINGH, 2015). A typical defence against collisions is the use of error-correcting codes (KHAN; PATHAN; ALRAJEH, 2012; SINGH, 2015).

- **Exhaustion attack**: Repetitive collisions can also be used to cause resource depletion (QADRI *et al.*, 2013; SASTRY; SULTHANA; VAGDEVI, 2013; BILAL *et al.*, 2014; DENER, 2014; JO *et al.*, 2015; BONAB; MASDARI, 2015). A feasible solution is to impose rate limits to the MAC admission control such that the network can disregard excessive requests, thus preventing the energy drain resulting from repeated transmissions (DENER, 2014).

- **Unfairness attack**: Rather than blocking access to a service outright, an attacker can degrade it for gaining an advantage, such as causing other nodes in a real-time MAC protocol to miss their transmission deadline (QADRI *et al.*, 2013; DENER, 2014; JO *et al.*, 2015; BONAB; MASDARI, 2015; SALEEM; ULLAH; YOO, 2009; SUN *et al.*, 2014). The use of small frames reduces the effect of such attacks by decreasing the amount of time with which an attacker can take hold of the communication channel (DENER, 2014).

4.4.4.3 Network layer

- **Selective Forwarding attack**: In a simple form of selective forwarding attack, malicious nodes try to stop the packets in the network by refusing to forward or drop the messages passing through them (Zada Khan *et al.*, 2012). It is known as Gray Hole attack. In addition, the malicious node may send the messages to the wrong path so that it can create unfaithful routing information in the network (VENKATRAMAN; DANIEL; MURUGABOOPATHI, 2013). Applying multiple paths to send data is a common defence. Another defence is to detect the malicious node or presume that it has failed and taken a different route (Zada Khan *et al.*, 2012; WALLGREN; RAZA; VOIGT, 2013; SASTRY; SULTHANA; VAGDEVI, 2013; DENER, 2014; BONAB; MASDARI, 2015; SINGH, 2015).

- **Sinkhole attack**: An attacker makes the compromised node look more attractive to surrounding nodes. So selective forwarding becomes very simple and data transfer hence takes place through the affected node (WALLGREN; RAZA; VOIGT, 2013; SASTRY; SULTHANA; VAGDEVI, 2013; DENER, 2014; BONAB; MASDARI, 2015; SINGH, 2015; SUN *et al.*, 2014). The solution to such a problem involves techniques like authentication, monitoring and redundancy (SINGH, 2015).

- **Sybil attack**: Often, sensors in a network might need to work together to accomplish a task, hence they can use distribution of sub-tasks and redundancy of information (SASTRY; SULTHANA; VAGDEVI, 2013). In a Sybil attack, a single node exhibits multiple identities to other nodes in the network (DENER, 2014). Authentication and encryption techniques can hinder an outsider from starting a Sybil attack on the network (PADMAVATHI; SHANMUGAPRIYA, 2009).

- **Wormhole attack**: A wormhole is an out of band connection between two nodes using wired or wireless links. Wormholes can be used to forward packets faster than via normal paths. A wormhole in itself is not necessarily a breach security; for example, a wormhole can be used to forward mission critical messages where high throughput is important and the rest of the traffic follows the normal path (WALLGREN; RAZA; VOIGT, 2013). However, a wormhole created by an attacker and combined with another attacks, such as sinkhole, is a serious security threat (SASTRY; SULTHANA; VAGDEVI, 2013; DENER, 2014; PADMAVATHI; SHANMUGAPRIYA, 2009).

- **HELLO Flood attack**: According to (WALLGREN; RAZA; VOIGT, 2013), the HELLO message refers to the initial message a node sends when joining a network. By broadcasting a "HELLO" message with strong signal power and a favourable routing metric, an attacker can introduce himself as a neighbour to many nodes, possibly the entire network; however, in some of the nodes in the attacker's vicinity, when trying to join the attacker, their messages may get lost because the attacker might be out of range (WALLGREN; RAZA; VOIGT, 2013; SASTRY; SULTHANA; VAGDEVI, 2013; DENER, 2014; BONAB; MASDARI, 2015; SINGH, 2015). Cryptography is mainly the current solution to these types of attacks (DENER, 2014).

4.4.4.4 Transport layer

- **Flooding attack**: The attacker can also cause immense traffic of useless messages on the network. This is known as the flooding. Such action may result in congestion and eventually lead to nodes exhaustion. It is considered a form of Denial of Service attack (SASTRY; SULTHANA; VAGDEVI, 2013). A solution for this problem is to require each connecting client to evidence its dedication to the connection by solving a puzzle (DENER, 2014).

- **Desynchronisation attack**: The adversary repetitively pushes messages, which convey sequence numbers to one or both of the endpoints (DENER, 2014). Requiring authentication of all packets communicated between hosts is one of the possible solutions to this type of attack (DENER, 2014; SALEEM; ULLAH; YOO, 2009; SUN *et al.*, 2014).

4.4.5 Safety issues in fly by wireless

Safety has a long tradition, being considered a mature area. There are several standards that can be used to create safe systems, such as RTCA/DO-178C (for UAV software) (RTCA, 2011), DO-254 (for UAV hardware) (RTCA Inc., 2000), ISO 26262 (for cars) (ISO26262, 2011). Safety deals with minimising the frequency of accidents or failures in a system, mainly when related with loss of life, high-value assets and it is related with undeliberated actions or events (SCHOITSCH, 2005).

Although the last years have presented a growth in new tools and techniques to the development of safe unmanned vehicles, challenges still remain mainly. Regardless all the issues related to safety in critical embedded system hardware and/or software (MARWEDEL, 2010; PATSAKIS; DELLIOS; BOUROCHE, 2014; KOOPMAN, 2004), this case study focuses in fly by wireless safety, not less important than others widely studied in the open literature.

Providing a safe wireless communication is to ensure that the information transmitted is received without any transmission error and loss of the information. Due to noise, interference and fading effects, wireless network cannot have zero transmission error once there is no system with zero risk. For wireless network, transmission error and loss of the information cannot be avoided, but they can be overcome by reducing or by detecting them.

Safety parameters required for wireless application differ considerably for different types of applications. Safety application with higher safety necessities, such as the fly by wireless in UAVs application, requires the communication link to be more reliable even at the cost of optimised throughput and performance when the communication is among the sensors and actuators inside the UAV.

Different parameters can be taken into account if the communication is between two UAVs. Thus, these parameters must to be adjusted based on the necessities to guarantee communications with safety (PENDLI, 2014).

- **Reliability**: communication links should be reliable and immune against noise, jamming signal, interference and fading effects. These provide a link without errors and losses. In the case of fly by wireless, the communication needs to be uninterrupted continuous mode to assure continuous data transmission.
- **Availability and Timely delivery of Information**: the link communication

availability and timely delivery information without failure must be treated in safety critical systems, such as UAVs. Thus, the technology used or designed to be used in fly by wireless vehicles must take into consideration the delay during information transmission and retransmission.

- **Real-Time Performance**: a typical problem is a real-time performance with burst errors. The main constraints providing real-time services are timely delivery of the information and reliability. Real-time performance and mobility management schemes are important parameters in safety critical systems. Once FANET pattern mobility is 3D based, the real-time performance parameters must be carefully taken into account.

- **Robustness**: the communication links must be robust even under adverse conditions against channel fading, low SNR (Signal to Noise Ratio) conditions and channel losses.

- **Optimised Throughput and Latency**: latency (maximum delay accepted for the data transmitted and received) must be low while not depleting batteries. Whereas, throughput (the amount of data transferred per unit time) must take into account the capability of the technology used. The packet size must be related with the latency, always considering reduced delays in transmission.

- **Optimised Power Consumption**: energy constraints plays a very important role in wireless communication, thus it is important to assure energy supply to the wireless nodes inside the UAV. Noises, interferences and fading effects can interfere in the power consumption. The design of the communication system must take these parameters into account to avoid unnecessarily and excessive amount of power consumption consumed. This parameter must be taken into account to assure the integrity of the vehicle.

- **Broadcast Storm Problem**: nodes in FANETs can be fixed or not. The distance between nodes can vary while in movement. To assure the route among nodes, FANETs rely on broadcast techniques. However, using flooding becomes a problem (can be interpreted as an attack to the communication system). To solve the problem of delivering packets to all nodes and avoiding packet redundancy and its associated problems (named Broadcast Storm Problem - BSP), techniques to mitigate BSP must be taken into account (PIRES *et al.*, 2016).

Safety issues need to be assured above the transport layer, complementing the security issues in other layers. As already mentioned, security impacts safety and vice-verse. It is possible to notice that in case of massively deployed unmanned vehicles, security issues have severe safety impact, thus, security gaps may become safety critical and safety problems or measures to maintain safety integrity levels may open breaches for security attacks.

4.4.6 Risk assessment

First of all, the most relevant security requirements must be identified, allowing the assessment of risk levels posed by potential attacks, thus prioritising the identified security requirements. According to Henniger *et al.* (2009), "the risk of an attack is seen as a function of the possible severity (i.e. the gain and loss) of the attack for the stakeholders and the estimated probability of occurrence of a successful attack." In case of threats to safety, the risk assessment also includes an additional controllable parameter. It is not trivial to quantify all factors influencing the risk of an attack, but the relative severity, success probability and controllability of attacks can be assessed, allowing a ranking of attacks based on their relative risk.

There are five aspects in which the severity of an attack is considered:

- Safety of people nearby the aerial vehicle;
- Privacy of aircraft modules, their manufacturers and suppliers;
- Financial losses that may be experienced by individuals or operators;
- Interference with operational performance of aircraft;
- Data loss in cases of sensitive missions.

Using the severity classification in Henniger *et al.* (2009) with adaptations for UAVs, a range of qualitative severity levels is defined in Table 5. The severity of the outcome is estimated for attacks with high-level goals.

The probability that a launched attack will succeed depends on the attack potential of the attacker and the attack potential that the target system is able to withstand. Essentially, the attack potential corresponds to the minimum effort required to create and carry out an attack. The higher the attackers' motivation, the higher efforts they may be willing to exert. Table 6 presents the attack potential ratings. Table 7 presents the rating of attack potential and attack probability.

For the safety component of the severity vector, the risk assessment includes an additional probability parameter that represents the potential for the pilot (automatic or not) to influence the severity of the outcome. It is referred to as "controllability" in Table 8.

Table 5 – Severity classification scheme for security threats.

Security threat severity class	Aspects of security threats			
	Safety	Privacy	Financial	Operational
0	No injuries	No unauthorised access to data	No financial loss	No impact on operational performance
1	Light or moderate injuries	Anonymous data only (no specific driver of vehicle data)	Low-level loss (≈ €10)	Impact not discernible to driver
2	Severe injuries (survival probable); light/moderate injuries for multiple vehicles	Identification of vehicle or driver; anonymous data for multiple vehicles	Moderate loss (≈ €100); low losses for multiple vehicles	Driver aware of performance degradation; indiscernible impacts for multiple vehicles
3	Life threatening (survival uncertain) or fatal injuries; severe injuries for multiple vehicles	Driver or vehicle tracking; identification of driver or vehicle for multiple vehicles	Heavy loss (≈ €1000); moderate losses for multiple vehicles	Significant impact on performance; noticeable impact for multiple vehicles
4	Life threatening or fatal injuries for multiple vehicles	Driver or vehicle tracking for multiple vehicles	Heavy losses for multiple vehicles	Significant impact for multiple vehicles

Source: Adapted from Henniger *et al.* (2009).

Table 6 – Rating of aspects of attack potential.

Factor	Level	Value
Elapsed time	$\leq$ 1 day	0
	$\leq$ 1 week	1
	$\leq$ 1 month	4
	$\leq$ 3 months	10
	$\leq$ 6 months	17
	$>$ 3 months	19
	not practical	∞
Expertise	Layman	0
	Proficient	3
	Expert	6
	Multiple experts	8
Knowledge of system	Public	0
	Restricted	3
	Sensitive	7
	Critical	11
Window of opportunity	Unnecessary/unlimited	0
	Easy	1
	Moderate	4
	Difficult	10
	None	∞
Equipment	Standard	0
	Specialised	4
	Bespoke	7
	Multiple bespoke	9

Source: Adapted from Henniger *et al.* (2009).

Table 7 – Rating of attack potential and attack probability.

Values	Attack potential required to identify and exploit attack scenario	Attack probability
0–9	Basic	5
10–13	Enhanced-Basic	4
14–19	Moderate	3
20–24	High	2
≥	Beyond High	1

Source: Adapted from Henniger *et al.* (2009).

Table 8 – Classification for controllability of safety hazards.

Controllability	Meaning
1	Avoidance of an accident is normally possible with a normal human response.
2	Avoidance of an accident is difficult, but usually possible with a sensible human response.
3	Avoidance of an accident is very difficult, but under favourable circumstances some control can be maintained with an experienced human response.
4	Situation cannot be influenced by a human response.

Source: Adapted from Henniger *et al.* (2009).

4.4.7 Risk

Table 9 maps severity of outcome, probability of attack, and controllability of the situation to risk level. The risk level is considered to be the higher, the more likely the success of the attacker is, the more severe the outcome is judged to be, and/or the more uncontrollable by the driver the situation is.

The risk class 7+ that is used in Table 9 for controllability classes $C = 3$ and $C = 4$ denotes levels of risk that are unlikely to be considered acceptable, such as safety hazards with the highest severity classes and threat levels, coupled with very low levels of controllability. For non-safety related risks, however, the mapping for controllability class $C = 1$ of Table 9 provides the relative risk level, ranging from 0 (lowest) to 6 (highest).

The risk levels are associated with the possible attacks by assessing relative severity at the higher levels of the attack trees and working up relative probabilities from the leaf nodes.

The proposed analysis process may support the development of future avionics applications based on A2X communications. It can be used in combination with the aircraft manufacturer's security policy, in order to decide whether to accept or transfer the identified security risks or to take measures to reduce or avoid specific risks. This approach can be integrated to unmanned vehicles by SPHERE's SMU, the safety management unit.

Table 9 – Security risk level as a function of attack probability P, threat severity class S_S and controllability C.

Security risk level		$P = 1$	$P = 2$	$P = 3$	$P = 4$	$P = 5$
$C = 1$	$S_S = 1$	0	0	1	2	3
	$S_S = 2$	0	1	2	3	4
	$S_S = 3$	1	2	3	4	5
	$S_S = 4$	2	3	4	5	6
$C = 2$	$S_S = 1$	0	1	2	3	4
	$S_S = 2$	1	2	3	4	5
	$S_S = 3$	2	3	4	5	6
	$S_S = 4$	3	4	5	6	7
$C = 3$	$S_S = 1$	1	2	3	4	5
	$S_S = 2$	2	3	4	5	6
	$S_S = 3$	3	4	5	6	7
	$S_S = 4$	4	5	6	7	7+
$C = 4$	$S_S = 1$	2	3	4	5	6
	$S_S = 2$	3	4	5	6	7
	$S_S = 3$	4	5	6	7	7+
	$S_S = 4$	5	6	7	7+	7+

Source: Adapted from Henniger *et al.* (2009).

4.5 Final remarks

This chapter presented three case studies on HAMSTER's SPHERE. The SPHERE platform provides safety and security for unmanned vehicles that use HAMSTER architecture. This is one of the most complete platforms presented, which has originated publications in the field of unmanned aerial vehicles due to a growing interest in ensuring security and safety mechanisms.

SPHERE is mature enough to be independent. Results in this chapter showed that it is possible to apply its authentication mechanism and secure communications adapted to a wide range of applications. Moreover, new approaches can consider new ways of measuring criticality and saving energy associated to SPHERE to provide safer, more secure and even more efficient unmanned vehicles enabled for heterogeneous communications. Next chapter will address case studies on criticality estimation by NCI.

CHAPTER 5

CASE STUDIES ON NCI

5.1 Chapter overview

HAMSTER's NCI index aims at the provision of a unified criticality index to identify unmanned vehicles that are more suitable for specific tasks, taking into consideration safety and security aspects in real-time. NCI index goes further with an evaluation of criticality that independently analyses security and safety evidences, as presented in Section 3.5, and covers not only unmanned vehicles, but also their internal elements, providing a trustworthy solution.

This chapter provides two hypothetical scenarios for NCI empiric evaluation. A precision agriculture application is the subject addressed by the first case study, detailed in Section 5.2. Later, environmental protection is the focus of the second case study, developed in Section 5.3, which engages aerial and ground vehicles on a joint mission.

5.2 Case study D: analysis of NCI in a precision agriculture scenario

This case study will evaluate how NCI could improve security and safety on a mission related to the acquisition of agricultural field imagery. NCI index is empirically applied to allow discussions on probable implications.

5.2.1 Background

Precision agriculture is one of the most important applications of unmanned vehicles, specially UAVs. In fact, recent works have widely explored big data (KSHETRI, 2014), computer vision (MINERVINI; SCHARR; TSAFTARIS, 2015), image processing techniques (HONKAVAARA *et al.*, 2013; VASUDEVAN; KUMAR; BHUVANESWARI, 2016), diseases identification (PONTI *et al.*, 2016), remote sensing (MULLA, 2013) and much more applied to precision agriculture. With growing interest and investments towards the development of more accurate approaches, techniques to support security, safety and tasks delegation should also be carefully investigated. On that direction, NCI can provide relevant information for a range of applications in the field.

High resolution images from crops are taken in order to observe, measure and respond to changes in agricultural fields. These images must have high quality and accurate geolocation to meet basic requirements for efficient issues identification. In fact, UAVs are flexible enough to perform such tasks, reason why they have been widely applied.

5.2.2 Empirical analysis

A popular example of UAV that meets the needs of fields image acquisition is the senseFly eBee (see Figure 24), an autonomous fixed wing aircraft projected to collect crops images in a fast and low-cost way if compared to traditional techniques, e.g. manned aircraft and satellite imagery acquisition.

Figure 24 – senseFly eBee UAV.

Source: Adapted from senseFly (2017).

In this scenario, three eBees are used to capture images of a crop. They communicate with a base station via a radio transmitter. Each eBee has an IMU (Inertial Measurement

Unit) and a GPS to identify its location, a camera to capture the images, an autopilot that also activates the camera, a propeller and servomotors.

This case study is composed by two situations: i) normal operation with three eBees acquiring data; ii) an eBee has a failure and requires a decision from a human operator that monitors the entire operation.

Both situations are analysed by defining each UAV module's ***NCIm*** and then the ***NCIe*** for each UAV. The comparison among UAVs' ***NCIe*** in both situations is an important resource that may help proceeding to an immediate solution during task execution. According to the formulae and definitions presented in Section 3.5, the ***NCIm*** for each eBee module in normal functioning are assumed as shown in Table 10.

Table 10 – *NCIm* for each module of a normal functioning eBee.

Module	$NCIm^{sec}$			$NCIm^{saf}$			*NCIm*
	storedData	*temporaryData*	total	*health*	*priority*	total	
GPS	0	0.3	0.3	0	0.5	0.25	0.275
IMU	0	0.3	0.3	0	1	0.5	0.4
Camera	0.5	0	0.5	0	0	0	0.25
Autopilot	0.3	0.3	0.3	0	1	0.5	0.4
Motor	0	0	0	0	0.5	0.25	0.125
Servomotor1	0	0	0	0	1	0.5	0.25
Servomotor2	0	0	0	0	1	0.5	0.25
Radio transmitter/receiver	0	0.3	0.3	0	0.3	0.15	0.225

Source: Elaborated by the author.

Considering that sensors (GPS and IMU), actuators (motor and servomotors) and the radio transmitter/receiver do not store data, thus ***storedData*** are set to 0. The camera stores images of the overflown region to identify assets and vulnerable areas, which leads to a score of 0.5 for ***storedData***. The autopilot stores information about the positioning of the aircraft when pictures are taken. This module's ***storedData*** is set to 0.3 due to the importance of stored information. Although the GPS log is important for the mission, it is not as important as acquired images, which justifies the difference in scores between these modules.

GPS, IMU, autopilot and radio transmitter/receiver manipulate data related to the aircraft positioning, thus ***temporaryData*** is set to 0.3. Remaining modules deal with no data that could be considered risky for the UAV, being set to 0 on ***temporaryData***. Regarding health, in a normal operation, all modules are properly working, thus ***health*** is set to 0.

The most critical modules for a proper functioning are IMU, autopilot and the servomotors. These modules are set to the highest value for ***priority***, 1. GPS and motor's ***priority*** score are set to 0.5, because it is still possible to land the UAV even if one of these modules fails. The radio transmitter/receiver is not necessary to the accomplishment of the eBee's task, i.e. if eBee loses connection with the base station, it can finish the task by itself. However, if eBee is forced to land, it is necessary to establish a communication via radio in order to locate and rescue the UAV, which justifies its value of 0.3 for ***priority***. Finally, regarding camera's ***priority***, it is set to 0 because if it fails, the UAV can safely go back home.

The definition of entities' ***worth*** measure is the relation among their costs. Since all UAVs are identical in this scenario, the ***worth*** is considered as 1. The variable $\boldsymbol{field}$ is set to 0 due to the fact that the covered area is a crop and presents no risk to the environment or people in case of an accident. The ***accomplishment*** is set to 0 since the mission can be restarted at any time and a deadline was not specified. Indeed, the ***NCIe*** for the three eBees happen to be the same in this situation (Table 11). Thus, the communications among them and the base station are equally implemented.

Table 11 – *NCIe* for each entity represented by eBees.

Entity	*worth*	*missionPenalty*			*NCIe*	$NCIe^{saf}$	$NCIe^{sec}$
		field	*accomplishment*	total			
eBee 1	1	0	0	0	0.345	0.331	0.213
eBee 2	1	0	0	0	0.345	0.331	0.213
eBee 3	1	0	0	0	0.345	0.331	0.213

Source: Elaborated by the author.

After this setup phase, a change in any ***NCIe*** may represent an issue and must be treated as an alert that triggers changes in the communication's behaviour and decision making. For instance, if the ***NCIe*** values change, the system has to prioritise the communication between the base station and the entity which currently has the highest ***NCIe***. Alternatively, the mission operator can send a new UAV to conclude the task.

The second situation analysed takes into consideration a failure on eBee 2's motor. That results in a change of the ***health*** value of the damaged UAV's motor to 1, which reflects in its ***NCIm*** that increases to 0.375 (see highlighted values in Table 12). As a consequence, it affects the ***NCIe*** and an alert to prioritise its communication is triggered. Therefore, a new mission is defined for the damaged eBee to maintain the communication as long as possible. Consequently, the ***priority*** for the radio transmitter/receiver is changed to 1. An updated scenario is shown in Table 12.

From now on, eBee 2's ***accomplishment*** variable is the highest on the network. Considering that it is a small field of operation, the ***accomplishment*** value is set to 0.5 (see Table 13). With these new values, ***NCIe*** increased by 30%, $\boldsymbol{NCIe^{saf}}$ by 75% and $\boldsymbol{NCIe^{sec}}$ by 65%. One can conclude that, based on that, the damaged eBee needs a prioritised communication, which will help the mission operator to rescue this UAV.

Table 12 – *NCIm* for each eBee module.

Module	$NCIm^{sec}$			$NCIm^{saf}$			*NCIm*
	storedData	*temporaryData*	total	*health*	*priority*	total	
GPS	0	0.3	0.3	0	0.5	0.25	0.275
IMU	0	0.3	0.3	0	1	0.5	0.4
Camera	0.5	0	0.5	0	0	0	0.25
Autopilot	0.3	0.3	0.3	0	1	0.5	0.4
Motor	0	0	0	**1**	0.5	**0.75**	**0.375**
Servomotor1	0	0	0	0	1	0.5	0.25
Servomotor2	0	0	0	0	1	0.5	0.25
Radio transmitter/receiver	0	0.3	0.3	0	**1**	**0.5**	**0.4**

Source: Elaborated by the author.

Table 13 – *NCIe* for each eBee after eBee 2 fails.

Entity	*worth*	*missionPenalty*			*NCIe*	$NCIe^{saf}$	$NCIe^{sec}$
		field	*accomplishment*	total			
eBee 1	1	0	0	0	0.345	0.331	0.213
eBee 2	1	0	**0.5**	**0.5**	**0.426**	**0.594**	**0.356**
eBee 3	1	0	0	0	0.345	0.331	0.213

Source: Elaborated by the author.

In summary, HAMSTER's NCI is an index that can be applied not just for communication prioritisation, but also for safety and security purposes due to its sub-indices. Decisions related to such aspects can be taken by HAMSTER's NIMBLE and SPHERE. For instance, the prioritisation of M2I communications due to a failure on an entity can be dealt by NIMBLE. On the other hand, when it comes to ensure the safety of an entity, SPHERE's SMU might take an appropriate action based on the safety index increase.

Furthermore, approaches for decision making on tasks delegation can also be based on NCI, allowing a more efficient resources usage.

5.3 Case study E: analysis of NCI in an environmental protection scenario

This case study verifies the applicability of NCI to an environmental protection scenario mixing UAVs and UGVs on a unique mission. The goal is to analyse NCI behaviour on an heterogeneous network, one of the benefits provided by HAMSTER architecture.

5.3.1 Background

Environmental protection has been a topic of research and application of unmanned vehicles, e.g. UAVs (NGUYEN, 2016; COVENEY; ROBERTS, 2017), USVs (YAN *et al.*, 2010) and UGVs (BONADIES; LEFCOURT; GADSDEN, 2016). The criticality of such tasks is inherent due to the necessity of trustworthy vehicles and systems, in order not to jeopardise any environmental protection area, but still operate with accuracy to identify suspicious activities. Thus, an empirical analysis of an assumed scenario will be carried out to highlight NCI benefits.

5.3.2 Empirical analysis

This scenario will assume the same ***NCIm*** values set to eBee modules on case study D (Section 5.2). Table 14 reproduces these values.

Table 14 – *NCIm* for each eBee module.

Module	$NCIm^{sec}$			$NCIm^{saf}$			*NCIm*
	storedData	*temporaryData*	total	*health*	*priority*	total	
GPS	0	0.3	0.3	0	0.5	0.25	0.275
IMU	0	0.3	0.3	0	1	0.5	0.4
Camera	0.5	0	0.5	0	0	0	0.25
Autopilot	0.3	0.3	0.3	0	1	0.5	0.4
Motor	0	0	0	0	0.5	0.25	0.125
Servomotor1	0	0	0	0	1	0.5	0.25
Servomotor2	0	0	0	0	1	0.5	0.25
Radio transmitter/receiver	0	0.3	0.3	0	0.3	0.15	0.225

Source: Elaborated by the author.

In this case study, an eBee flies over a field searching for illegal actions on an environmental protection area. If anything suspicious is identified, a 3DR Solo quadrotor UAV (see Figure 25) is sent to the specific position to acquire more detailed information. Then, it is assumed that Solo UAV happens to be captured, triggering CaRINA (**Ca**rro **R**obótico **I**nteligente para **N**avegação **A**utônoma) UGV (FERNANDES *et al.*, 2014) to rescue Solo UAV and take appropriate action regarding the environmental related issue.

A 3DR Solo UAV is composed by GPS, IMU, camera, autopilot, motors and the Wi-Fi transmitter/receiver. The GPS, IMU, motors and Wi-Fi transmitter/receiver do not store any data. The camera store images of the overflown area, so the ***storedData*** for this model is set to 0.5. The autopilot ***storedData*** is set to 0.3 due to the fact that it saves the UAV position data.

The GPS and IMU modules manipulate information about the Solo position, thus their ***temporaryData*** are defined as 0.3. The camera, autopilot and the transmitter send

Figure 25 – 3DR Solo UAV.

Source: Adapted from 3DR (2017).

collected images in real-time to the base station, then their ***temporaryData*** are set to 0.5.

All ***health*** scores are defined as 0 because all modules are properly working. IMU, autopilot and motors have ***priority*** equal to 1 since failures in these modules can lead to accidents. GPS is a special case that is set to 0.7 that, in case of failure, the system switches to manual mode, allowing a remote pilot to make an emergency landing. The communication between base station and Solo needs to be available at all times; otherwise it will fly back home, leading to a change on Wi-Fi transmitter ***priority*** to 0.5. If the camera is damaged, the UAV can be controlled automatically or manually without any problems, so its ***priority*** is 0. Table 15 presents these values.

Table 15 – ***NCIm*** for each Solo module during normal conditions.

Module	$NCIm^{sec}$			$NCIm^{saf}$			***NCIm***
	storedData	***temporaryData***	total	***health***	***priority***	total	
GPS	0	0.3	0.3	0	0.7	0.35	0.325
IMU	0	0.3	0.3	0	1	0.5	0.4
Camera	0.5	0.5	0.5	0	0	0	0.25
Autopilot	0.3	0.5	0.5	0	1	0.5	0.5
Motors	0	0	0	0	1	0.5	0.25
Wi-Fi transmitter/receiver	0	0.5	0.5	0	0.5	0.25	0.375

Source: Elaborated by the author.

The ***worth*** of an entity is the relation between its cost and the highest entity's cost. In this case study, the most expensive entity is CaRINA, which costs approximately US\$46,500, followed by eBee, which costs approximately US\$26,000, and Solo, which costs US\$470.

Since this case study is about an environmental protection, the mission area is a place that cannot be damaged by the entities, so the ***field*** is set to 0.5 for all entities. At the time that eBee is executing a preliminary search for any warning points, its ***accomplishment*** is set to 0.2. When a warning point is detected, Solo starts operating with an ***accomplishment*** set to 0.4, since the accomplishment now is more important. These values are presented in Table 16.

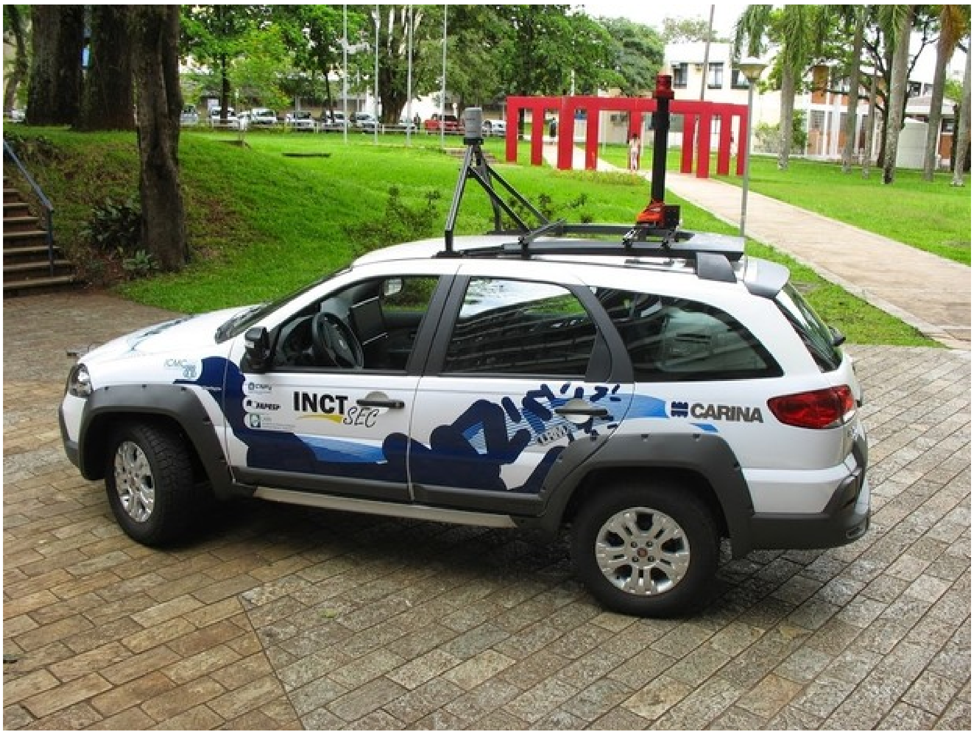
Table 16 – *NCIe* of eBee and Solo UAVs.

Entity	*worth*	*missionPenalty*			*NCIe*	$NCIe^{saf}$	$NCIe^{sec}$
		field	*accomplishment*	total			
eBee	0.565	0.5	0.2	0.5	0.359	0.416	0.356
Solo	0.01	0.5	0.4	0.5	0.383	0.425	0.425

Source: Elaborated by the author.

At some point, Solo UAV happens to be hijacked, leading to a change of its modules ***health*** to 1. Table 17 highlights updated values. These changes force an emergency rescue and a search by intruders on the protected area. This task is assumed to be performed by CaRINA UGV (see Figure 26), which is composed by GPS, IMU, stereo camera, velodyne, autopilot, motor, steering wheel, brake and Wi-Fi transmitter/receiver.

Figure 26 – CaRINA UGV.



Source: Adapted from LRM (2017).

Table 17 – ***NCIm*** for each module after Solo UAV hijacking.

Module	$NCIm^{sec}$			$NCIm^{saf}$			*NCIm*
	storedData	*temporaryData*	total	**health**	*priority*	total	
GPS	0	0.3	0.3	**1**	0.7	**0.85**	**0.575**
IMU	0	0.3	0.3	**1**	1	**1**	**0.65**
Camera	0.5	0.5	0.5	**1**	0	**0.5**	**0.5**
Autopilot	0.3	0.5	0.5	**1**	1	**1**	**0.75**
Motors	0	0	0	**1**	1	**1**	**0.5**
Wi-Fi transmitter/receiver	0	0.5	0.5	**1**	0.5	**0.75**	**0.625**

Source: Elaborated by the author.

Only the autopilot ***storedData*** is set to 0.3, since it stores position data, and all the other modules are set to 0. Regarding ***temporaryData***, GPS and IMU are set to 0.3 considering that they will be providing location information about the UGV. Velodyne, autopilot, stereo camera and Wi-Fi transmitter/receiver deal with driving assistance data, having a ***temporaryData*** score slightly superior, set to 0.5. Finally, ***temporaryData*** for motor, steering wheel and brake are set to 0.

If a failure occurs with GPS, IMU, motor, steering wheel and Wi-Fi transmitter/receiver, no damage is expected, leading to a ***priority*** score of 0.5. On the other hand, if stereo camera or velodyne fail, the UGV might or might not jeopardise people or property, thus ***priority*** is set to 0.8. Finally, the most critical modules are autopilot and break, both set to 1 since failures will probably lead to accidents. These values can be seen in Table 18.

Table 18 – ***NCIm*** of each module of CaRINA UGV.

Module	$NCIm^{sec}$			$NCIm^{saf}$			*NCIm*
	storedData	*temporaryData*	total	*health*	*priority*	total	
GPS	0	0.3	0.3	0	0.5	0.25	0.275
IMU	0	0.3	0.3	0	0.5	0.25	0.275
Stereo camera	0	0.5	0.5	0	0.8	0.4	0.45
Velodyne	0	0.5	0.5	0	0.8	0.4	0.45
Autopilot	0.3	0.5	0.5	0	1	0.5	0.5
Motor	0	0	0	0	0.5	0.25	0.125
Steering wheel	0	0	0	0	0.5	0.25	0.125
Brake	0	0	0	0	1	0.5	0.25
Wi-Fi transmitter/receiver	0	0.5	0.5	0	0.5	0.25	0.375

Source: Elaborated by the author.

CaRINA's ***field*** is set 0.5 since it is an environmental protection field and ***accomplisment*** is set to 0.8, considering that the UGV is rescuing Solo UAV and taking appropriate action over the identified issue. ***worth*** value does not change. Table 19 presents updated entity scores.

Table 19 – *NCIe* for all the entities.

Entity	*worth*	*missionPenalty*			*NCIe*	$NCIe^{saf}$	$NCIe^{sec}$
		field	*accomplishment*	total			
eBee	0.565	0.5	0.2	0.5	0.359	0.416	0.356
Solo	0.01	0.5	0.4	0.5	0.609	0.925	0.425
CaRINA	1	0.5	0.8	0.8	0.517	0.468	0.458

Source: Elaborated by the author.

This case study helps with the identification of which unmanned vehicle is the most critical in a situation that also considers the context of application. As the chosen field was an environmental protection area, there is an increase on overall criticality, since accidents may lead to important damages to the environment. Furthermore, this case study shows that NCI index is applicable to heterogeneous missions and its sub-indices for safety and security are important tools for the development of accurate decision making approaches regarding mission performing.

5.4 Additional result: a GPS spoofing attack

A relevant application of NCI can be seen on GPS spoofing attacks (MIXON, 2013; Inside GNSS, 2013). An attacker transmits a stronger GPS-like signal nearby the unmanned vehicle, overriding the authentic one. Such situation induces to wrong paths and consequent accidents, depending on the field of operation. Experiments were carried out on that field using an Android smartphone[1]. Figure 27a shows the desired trajectory between two points in the city of São Paulo and Figure 27b presents the FakeGPS app being set to spoof Parque Ibirapuera's location to the smartphone.

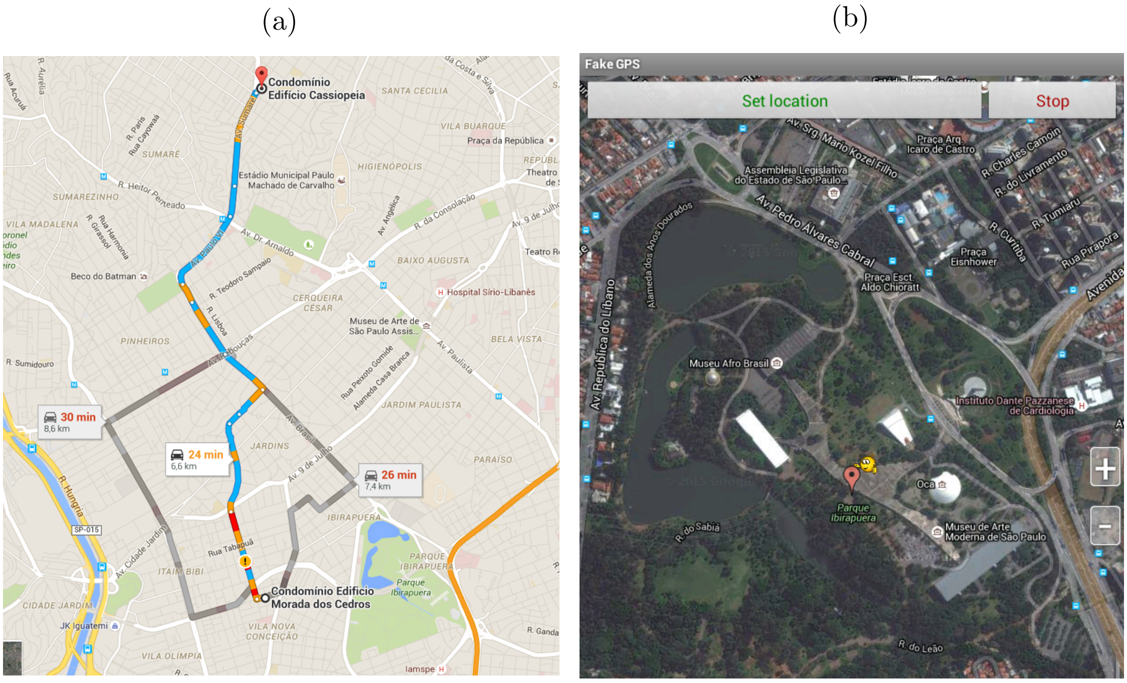
This experiment was performed first with the authentic GPS signal received by the smartphone. Figure 28a presents the trajectory performed inbound (blue line) and outbound (red line). Later, the same experiment was run with FakeGPS app spoofing a fake location to the smartphone. Results can be seen in Figure 28b, where location had suspicious variations.

If NCI is applied to this scenario, GPS module's ***health*** would reflect such variation with a score of 1 (most critical). Thus, an appropriate action would need to take place. For instance, HAMSTER's SPHERE would isolate GPS module by blocking communications to and from it. Moreover, if no other location module was available on the system, SPHERE would request an Emergency phase from HAMSTER's NP, which in turn would stop the

[1] Although results were obtained using the GPS of a smartphone, a GPS spoofing attack to a UAV would follow a similar approach, thus leading us to conclude that this evidences help towards the case study objective.

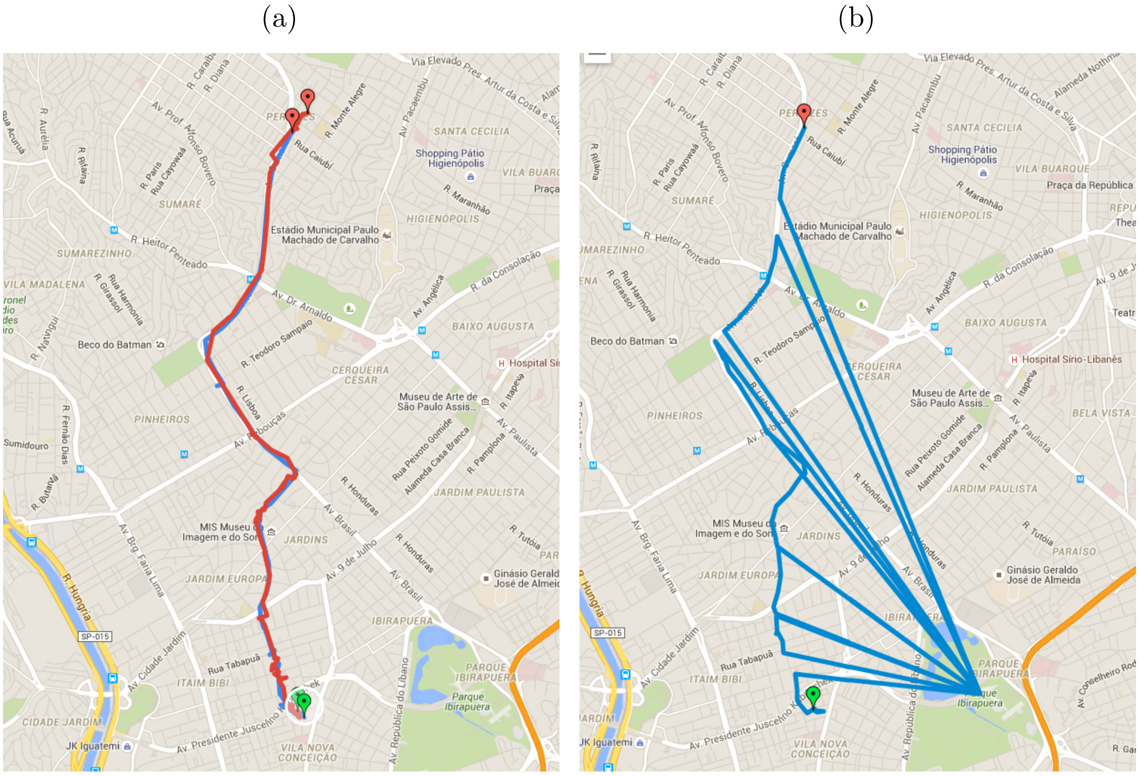
vehicle operation as soon as possible by parking or landing it.

Figure 27 – (a) Desired trajectory between two points in the city of São Paulo; (b) FakeGPS app spoofing Parque Ibirapuera's location to the smartphone.



Source: Elaborated by the author.

Figure 28 – Trajectory performed (a) with authentic GPS signal; (b) under attack.



Source: Elaborated by the author.

5.5 Final remarks

This chapter provided empirical analyses of NCI that validated the criticality index for the case studies presented only. Two scenarios were evaluated regarding how NCI could contribute on tasks delegation, taking into account safety and security at all times. An additional result was conducted to emphasise that a GPS spoofing attack, for instance, would impact NCI index. Although very specific, these analyses are seen in many applications of unmanned vehicles and this validation may be extended to other scenarios.

Furthermore, NCI classification can be used for more than the applications discussed in this chapter, specially due to the fact that it is fully adaptable to heterogeneous scenarios. Modern UV applications should benefit from this approach, leading to a supportive index for the development of more safe and secure vehicles.

Another approach provided by HAMSTER is related to energy efficiency. Next chapter presents case studies that validate Navigation Phases platform.

CHAPTER

6

CASE STUDIES ON NAVIGATION PHASES

6.1 Chapter overview

The substitution of cables by wireless communication has been investigated on UAVs and UGVs under the topics of fly by wireless and drive by wireless. One of the benefits provided by such approaches is the reduction of energy consumption by fully or partially turning off idle nodes when possible. HAMSTER architecture provides Navigation Phases platform, detailed in Section 3.6, which shares this goal of reduced energy consumption, but is not limited to wireless communications only. In this chapter, two case studies will be presented towards the subject: Section 6.2 will compare five communication protocols for wireless communications on internal unmanned vehicles and Section 6.3 will demonstrate experiments on a UAV prototype on how the Navigation Phases concept may contribute for energy efficiency. These results were published in: Pigatto *et al.* (2014), Pigatto *et al.* (2016) and Pigatto *et al.* (2016).

6.2 Case study F: fly by wireless with Flying HAMSTER

This case study aims to provide guidelines for the development of UAS with Flying HAMSTER. Thus, it consists on real experiments regarding efforts to the direction of fly by wireless on IAC.

6.2.1 Background

In the last few years, there has been a huge eagerness by the industry towards fly by wireless (STUDOR, 2007; GOMEZ, 2010; SAMPIGETHAYA; POOVENDRAN, 2013; SáMANO-ROBLES *et al.*, 2016) and drive by wireless (KHAN, 2011; STÄHLE; HUANG; KNOLL, 2014). Due to that, the aerospace industry and technology providers were motivated to establish: i) a new emphasis for system engineering approaches to reduce cables and connectors, ii) provisions for modularity and accessibility in the vehicle architecture and iii) a set of technologies that support alternatives to wired connectivity.

Leipold, Tassetto and Bovelli (2013) investigated Ultra WideBand technology for in-cabin communication with optimised resource allocation in high performance systems. Yedavalli and Belapurkar (2011) presented a survey on the use of wireless sensor networks for aircraft control and health monitoring. Moreover, Dang *et al.* (2012) pointed out the opportunities and challenges on using wireless inter-connect for safety-critical avionics. In summary, the benefits include weight reduction, increased flexibility and decreased costs and maintenance. However, electromagnetic susceptibility and security issues still remain as challenges.

Wireless communications may also be applied for aircraft health monitoring. Sampigethaya and Poovendran (2012) have shown that smart sensors which possess a signal processing unit, memory and a wireless communication unit are deployed on aircraft structures and systems for health monitoring. Such sensors may have heterogeneous capabilities (e.g., node transmission range) and modalities (e.g., vibration, temperature, pressure etc).

Garcia *et al.* (2007) presented a framework of an interface between wireless sensor networks and personal devices like PDAs (Photo-Diode Array), mobile phones and laptops. The framework has a flexible architecture which may be easily adapted to display any kind of data received from the network sensors using content description. The system is applied to display information retrieved from a Bluetooth based wireless sensor network platform which operates on-board on a UAV.

The goal of this case study is to present a performance evaluation of five different communication schemes applied to six sensor nodes and a master node embedded on a UAV.

6.2.2 Material and methods

The experimental scenario is composed by six slave nodes plus a master node placed inside and outside Tiriba UAV prototype (BRANCO *et al.*, 2011) in representation of real sensors and actuators. The communication efficiency of wireless nodes based on IEEE 802.15.4 is evaluated with time division multiple access (TDMA) (ALBA *et al.*, 2007), Flurry and Periodic (WEI *et al.*, 2011) based protocols.

Although the promising applications enabled by wireless sensor networks are very attractive, there are many system challenges to solve. First of all, energy is an essential problem since sensors are usually battery-powered. Second, in some emergency applications, a short time of data collection is also required. TDMA is an efficient choice that meets these requirements (ALBA *et al.*, 2007). It eliminates collisions by avoiding idle listening and entering inactive states. Furthermore, as a collision-free access method, TDMA can bound the delays of packets and guarantee reliable communication. On the other hand, Periodic protocols operate with slave nodes periodically sending messages to the master module. The advantage of using this model is mainly linked to energy saving, once there is the possibility of deactivating nodes when not communicating (WEI *et al.*, 2011). In this evaluation, a third protocol was chosen for investigation, which is based on random operation, named Flurry within this case study context.

Five implementations inspired by above protocols were tested with Arduino Leonardo boards equipped with XBee radio modules, as follows:

1. **TDMA without requests**: the slave node sends a message each 100 ms; then waits for a resynchronisation message each 1200 ms.
2. **TDMA with requests**: if prompted to, the slave node sends a message each 100 ms; then waits for a resynchronisation message each 1200 ms.
3. **Flurry**: each node has 33% chance of being activated; once active, it sends a message each 100 ms until achieving 12 messages; then, it waits for 1200 ms.
4. **Periodic without requests**: all nodes simultaneously send a message each 100 ms; then, for each batch of execution, they wait for a resynchronisation message.
5. **Periodic with requests**: all nodes simultaneously send a message each 100 ms; then, for each batch of execution, they wait for a request message from the master module to start sending messages again.

Implementations identified with requests are dependent on the master node, which was programmed to send a request message each 50 ms.

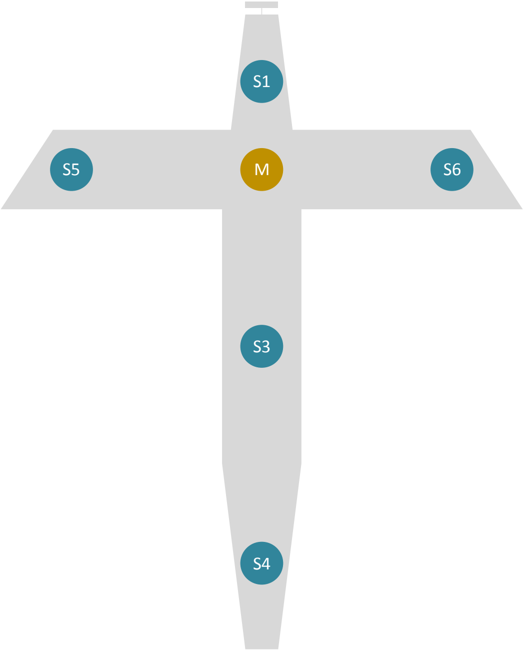
A master node (M) and four slave node ($S1$ to $S4$) were positioned inside the aircraft. Two additional slaves nodes ($S5$ and $S6$) were attached to the wings. Figure 29 presents nodes distribution along Tiriba and figure 30 shows a photo of the setup phase.

Experiments were replicated 10 times, ensuring a statistical validation since there was no large standard deviation among results.

6.2.3 Results and discussion

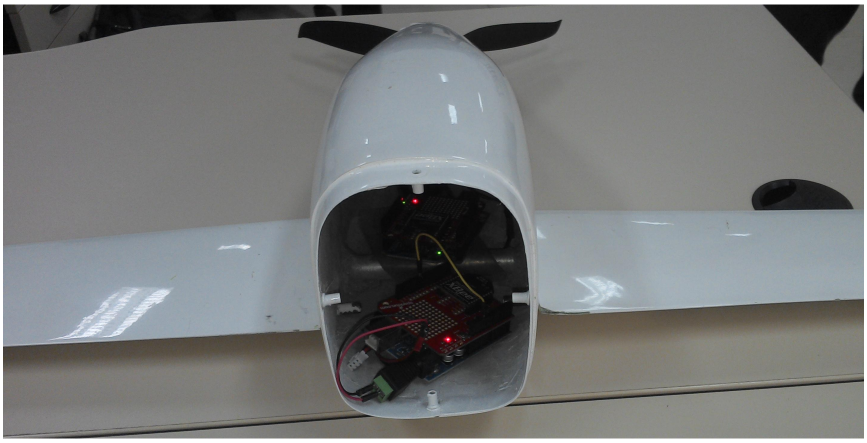
Figure 31 illustrates the frequency of messages sent by each protocol. TDMA implementations (with and without requests) had an inferior performance, which is due to the algorithm dependency on master node requests. A random behaviour by Flurry protocol was also noticed. And the Periodic protocol was the one which transmitted more

Figure 29 – The nodes distribution over the aircraft: slave modules ($S1$-$S4$) and Master module (M) are inside the aircraft, while slave modules $S5$ and $S6$ are above the aircraft wings.



Source: Adapted from Pigatto *et al.* (2016).

Figure 30 – Modules positioned inside the aircraft during the setup stage.

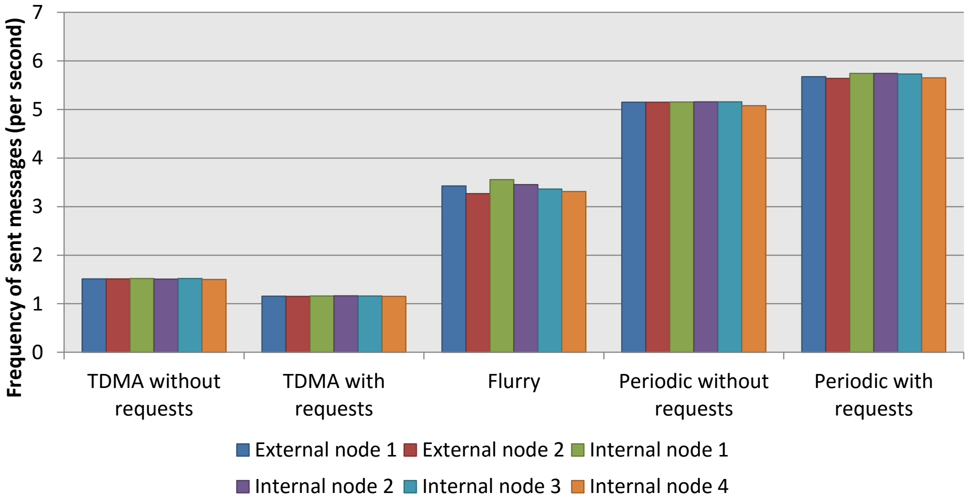


Source: Adapted from Pigatto *et al.* (2016).

messages. In some cases, external modules had a bit less success whether compared to internal ones. It was expected due to the fact that external modules have more obstacles to overpass, once they are positioned outside the UAV.

The communication efficiency was calculated by comparing transmitted and successfully delivered messages. Figure 32 presents the percentage of successfully delivered

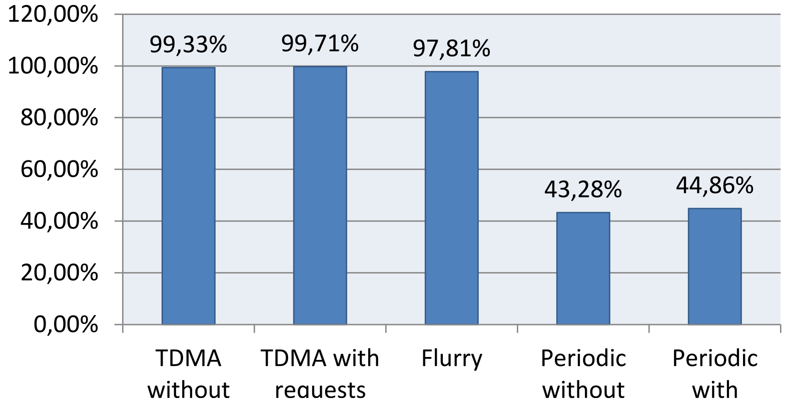
Figure 31 – Transmission frequency.



Source: Adapted from Pigatto *et al.* (2016).

messages. Although an overall bad performance is observed on results by Periodic protocols, they had a slightly superior performance indeed when it comes to successfully delivered messages. On the other hand, Flurry and TDMA protocols had almost all messages successfully delivered. TDMA with and without requests had 99.71% and 99.33% of messages delivered, respectively. Similarly, Flurry protocol had 97.81% of success.

Figure 32 – The communication efficiency of each communication scheme.



Source: Adapted from Pigatto *et al.* (2016).

These results provide experimental results on the applicability of IEEE 802.15.4 to fly by wireless and drive by wireless paradigms. Next section presents a case study on energy saving approaches for internal wireless communications on unmanned vehicles.

6.3 Case study G: reducing energy consumption on internal communications with HAMSTER's Navigation Phases

Navigation Phases is an innovative concept provided by HAMSTER architecture that may potentially bring benefits to internal communications. This section presents both simulated and real experiments on how such concept introduces a new way of saving energy, improves fly by wireless and drive by wireless paradigms and opens new ways of providing security and safety to unmanned systems.

6.3.1 Material and methods

Navigation Phases concept was adapted to a UAS and is presented in Table 20. This table will be used as reference for both simulated and real experiments.

Table 20 – Navigation phases applied to UAS.

Navigation phases	Navigation Sub-phases	ID	Description	Active modules	Identifier
1	Pre-flight	1.1	Modules health, energy and authentication checking	All nodes	ALL
2	Departure and climb	2.1	Taxiing	Main nodes only	MAIN
		2.2	Taking-off	Main nodes only	MAIN
		2.3	Climbing	Main nodes only	MAIN
3	Cruise	3.1	Stabilising from climbing	Main nodes only	MAIN
		3.2	Heading to the destination	All nodes	ALL
		3.3	Performing mission	All nodes	ALL
		3.4	Preparing to descent	Main nodes only	ALL
4	Descent and approach	4.1	Descending	Main nodes only	MAIN
		4.2	Landing	Main nodes only	MAIN
		4.3	Taxiing	Main nodes only	MAIN
5	Post-flight	5.1	Modules health, energy and authentication checking	All nodes	ALL
		5.2	Mission data manipulation	Mission nodes only	MISSION
E	Emergencies	E.1	Returning to the Ground Control Station	Main nodes only	MAIN
		E.2	Landing ASAP	Main nodes only	MAIN
		E.3	Starting self-destruction (wipe data)	Mission nodes only	MISSION
		E.4	Stabilising (after non predicted movements)	Main nodes only	MAIN

Source: Elaborated by the author.

Pre-flight phase is dedicated to several inspections. In this case study, it is composed of only one sub-phase, which performs authentication and monitors "health" and energy. The transmission rate and the size of the exchanged messages are the same for all modules.

Departure and climbing phase occurs when the UAV is moving on the ground, taking off and stabilising in the air. At this phase, the UAV works exclusively with modules classified as Main since it is a critical phase.

Cruise phase is usually the longest flight. The UAV reaches a specific altitude (which can vary during the phase), stabilises, go to the destination (e.g., mission execution site), performs the mission and prepares for the next phase. At this stage, there is a greater variation about which nodes will be active in each sub-phase.

Descent and approach phase is the period when the UAV starts to descent, landing and then moving on the ground. Only modules classified as Main are allowed to exchange messages at this stage.

Post-flight phase the first phase checking is performed again and the acquisition and manipulation of mission data also takes place. The frequency and the amount of data exchanged at this stage varies for each module.

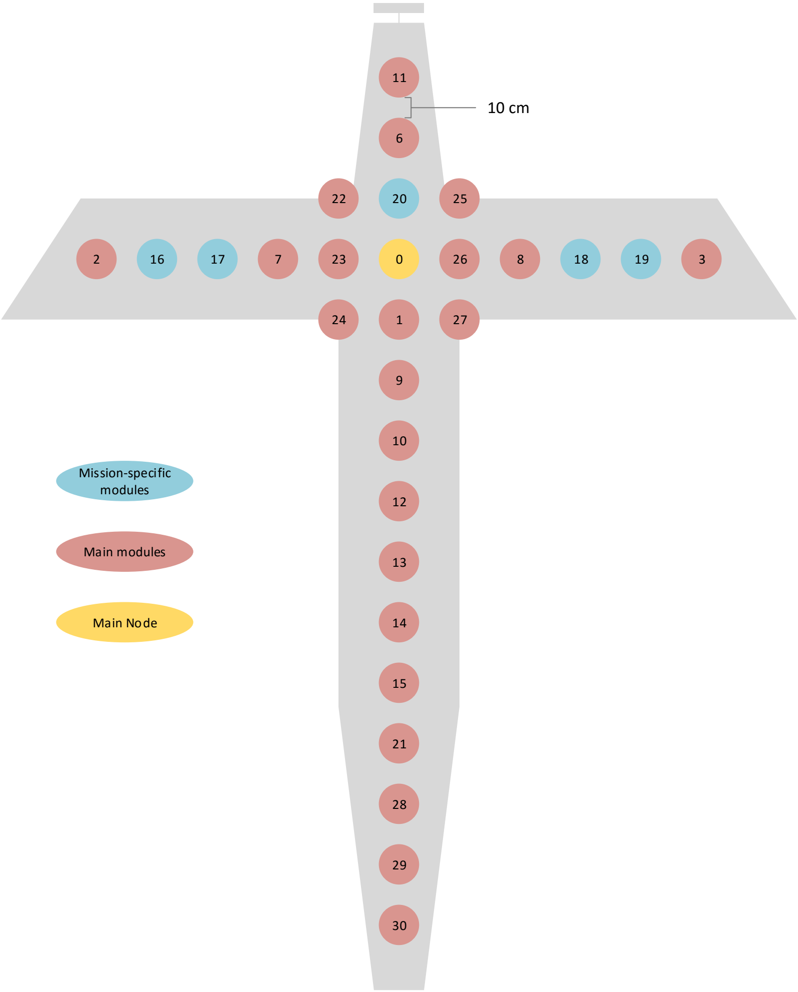
And finally, **Emergencies** phase includes various abnormal situations, such as power outages, flight difficulties, adverse weather conditions, unexpected obstacles, security attacks etc. For each case, a procedure is triggered trying to circumvent the problematic situation, save/delete sensitive data and prevent the UAV to cause some kind of injury.

6.3.2 Simulation of Navigation Phases

First, simulated results were performed for an initial behaviour analysis. A set of 30 modules were classified into Main and Mission-specific categories. Main modules are essential for the aircraft operation and must be working at almost all the navigation phases. Mission-specific modules can be turned off at various stages of operation, reducing energy consumption. Figure 33 demonstrates the modules positioning on simulated experiments.

Considering all the sensors that usually can be found in a UAV, an empiric list was created and is presented in Table 21. It shows details about specific hardware, the energy supply needed by each module, the message size given by each hardware manufacturer and the burst size. For the translation of the following parameters to OMNeT++ simulator, it was considered that XBee demands around 2.1 V to 3.6 V of energy supply.

Figure 33 – Nodes distributed on the inner side of UAV.



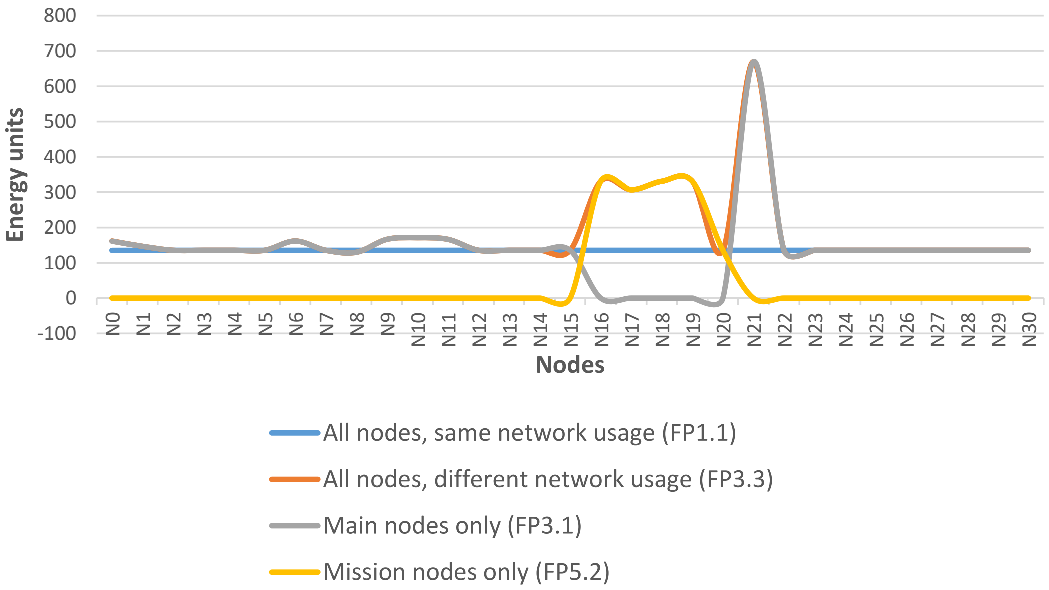
Source: Elaborated by the author.

Table 21 – Modules, respective hardware and parameters considered for simulations.

Groups of Modules	Modules	ID	Product	Supply	XBee Supply	Package size	burstSize
Flight Controller Navigation Controller Subsystem Controller	Central Unit	0	ODROID-U3 (Autopilot)	5V	2.1V to 3.6V	128 bytes	3
HAMSTER Sphere	Security Central Unit	1	Overo® FIRESTORM-Y COM (general purpose)	3.3V to 4.2V	2.1V to 3.6V	32 bytes	1
Actuators (servo)	Aileron (left wing)	2	-	3.6V	2.1V to 3.6V	32 bytes	1
	Aileron (right wing)	3	-	3.6V	2.1V to 3.6V	32 bytes	1
	Elevon (back)	4	-	3.6V	2.1V to 3.6V	32 bytes	1
	Rudder (back)	5	-	3.6V	2.1V to 3.6V	32 bytes	1
	Motor	6	Align RCMBL700MX	5V	2.1V to 3.6V	32 bytes	3
Weather Monitoring/Forecasting	Stormscope	7	-	3.6V	2.1V to 3.6V	32 bytes	3
Flight	GPS	8	GPS Receiver LS20031 5Hz	3.3V	2.1V to 3.6V	75 bytes / 5 Hz	3
	Barometric Altimeter Aerodynamic Speed Magnetic Compass IMU (Inertial Measurement Unit)	9	ADIS16407	4.75V to 5.25V	2.1V to 3.6V	32 bytes / 1.5 kHz	2
	Sonar	10	MB1242 I2CXL-MaxSonar®-EZ4	3V to 5.5V	2.1V to 3.6V	32 bytes	3
	Airspeed Sensor	11	MPXV7002DP Based Differential Airspeed sensor	4.75V to 5.25V	2.1V to 3.6V	1-2 bytes / 1kHz	3
Takeoff Gear	Catapult	12	-	3.6V	2.1V to 3.6V	32 bytes	1
	Wheels	13	-	3.6V	2.1V to 3.6V	32 bytes	1
	Ski	14	-	3.6V	2.1V to 3.6V	32 bytes	1
	Rocket	15	-	3.6V	2.1V to 3.6V	32 bytes	1
Mission	Camera	16	FPV Camera	9V to 14V	2.1V to 3.6V	3 Mbytes / 3 sec	5
	Camera	17	Mini-MCA (Tetracam's Miniature Multiple Camera Array)	9V to 14V	2.1V to 3.6V	3 Mbytes / 3 sec	5
	Camera	18	ADC Lite (Tetracam's Lightweight ADC)	9V to 14V	2.1V to 3.6V	3 Mbytes / 3 sec	5
	Camera	19	Tau 2 LWIR Thermal Imaging Camera Cores	9V to 14V	2.1V to 3.6V	3 Mbytes / 3 sec	5
	Mission controller Unit	20	-	3.6V	2.1V to 3.6V	32 bytes	3
Air Traffic Control	Transponder ADS-B	21	XPS-TR Mode S with ADS-B Out	10V to 32V	2.1V to 3.6V	32 bytes	2
Engine	RPM	22	-	3.6V	2.1V to 3.6V	32 bytes	2
	CHT (Cylinder Head Temperature)	23	-	3.6V	2.1V to 3.6V	32 bytes	2
	EGT (Exhaust Gas Temperature)	24	-	3.6V	2.1V to 3.6V	32 bytes	2
	Fuel	25	-	3.6V	2.1V to 3.6V	32 bytes	2
Reaction	Gas Turbine	26	-	3.6V	2.1V to 3.6V	32 bytes	2
Landing Recovery Gear	Wheels	27	-	3.6V	2.1V to 3.6V	32 bytes	1
	Ski	28	-	3.6V	2.1V to 3.6V	32 bytes	1
	Airbag	29	-	3.6V	2.1V to 3.6V	32 bytes	1
	Parachute	30	-	3.6V	2.1V to 3.6V	32 bytes	1

Source: Elaborated by the author.

Figure 34 – Energy consumption of 31 nodes in fly by wireless communications in four different situations. The Flight Phases identified in each situation can be seen in Table 20.



Source: Elaborated by the author.

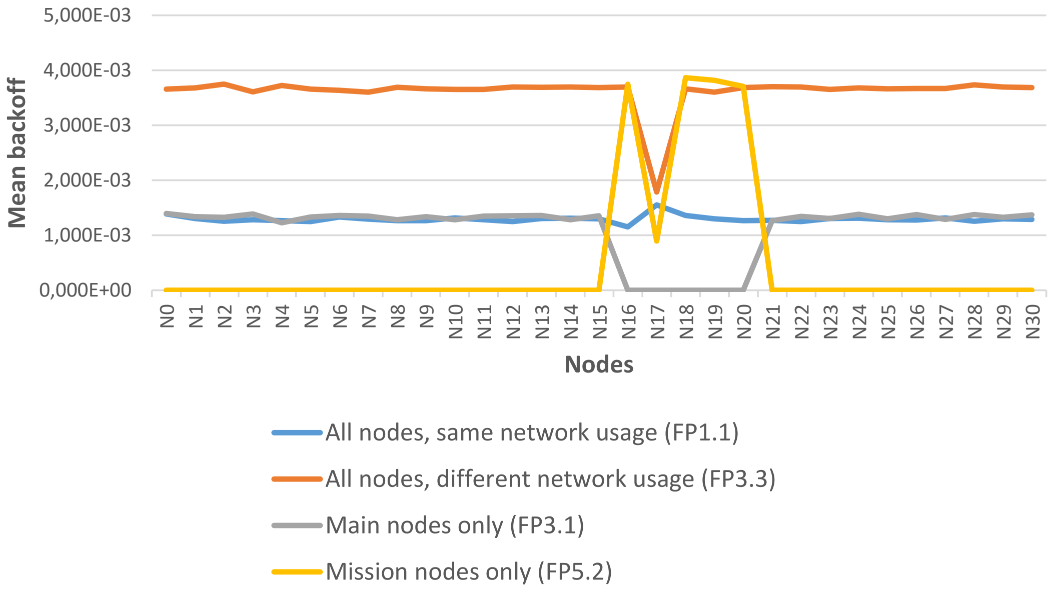
Figure 34 presents some different situations where the energy consumption can be reduced. The blue line represents the energy consumption of all nodes during Navigation Phase 1.1. This phase consists on SPHERE's CSU authentication, which is performed before taking off, thus the energy consumption is pretty much the same for every node. Similarly, the red line shows the consumption when all nodes are active and also illustrates the peak in nodes $N16$ to $N19$, the Mission-specific ones. These nodes were defined as cameras and their energy consumption is bigger as shown in Table 21. The peak on node $N21$ is a specific case where the position of a node is determinant for its performance and energy consumption. Finally, the grey and yellow lines are either showing the energy consumption of Main nodes and Mission-specific nodes only, respectively.

Figure 35 shows the mean back off for the same case aforementioned. A binary exponential back off or truncated binary exponential back off refers to an algorithm used to space out repeated retransmissions of the same block of data, often as part of network congestion avoidance. In these experiments, the technique took place when high amounts of data were transferred.

6.3.3 Real experiments on Navigation Phases

Although simulations may provide a more controlled environment and the possibility of replicating experiments as much as needed, experiments on real prototypes can highlight

Figure 35 – Backoff of 31 nodes in four different situations. The Navigation Phases identified in each situation can be seen in Table 20.



Source: Elaborated by the author.

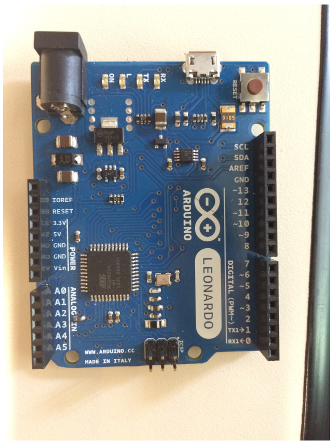
new issues that a simulator might not consider. Thus, a set of experiments was run using Arduino Leonardo and XBee boards.

Arduino board is equipped with an ATmega32u4 micro controller and has 20 digital input/output pins, a 16 MHz crystal oscillator, a micro USB connection, a power jack, one ICSP (In-Circuit Serial Programming) header and a reset button (ARDUINO, 2016) (see figure 36). In this case study, Leonardo was used exclusively to control the XBee regarding data processing and to determine the cycles that XBee would be active or inactive. Concerning serial communication, Leonardo has an advantage when compared to other Arduino boards, since it has 2 serial ports. Thus, for these experiments, a serial port was used to exchange data with the computer and the other to communicate with XBee. Tests were easily monitored and controlled by the computer.

A specific expansion shield was used to connect XBee to Arduino Leonardo (see Figure 37). It enables communication between Arduino Leonardo pins and the XBee ports.

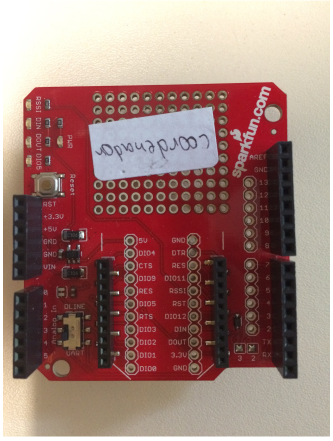
The XBees used in this case study were XBee S1 and XBee-PRO S1, both implementing IEEE 802.15.4 standard. Thus, they are suitable for projects with low-cost and low-power requirements. Differences between these XBee models consist basically on the transmitted signal power and the sensitivity to received signals. A comparison is shown in Table 22 (SPARKFUN, 2016).

Figure 36 – Arduino Leonardo photograph.



Source: Elaborated by the author.

Figure 37 – Expansion shield to connect XBee to Arduino Leonardo photograph.



Source: Elaborated by the author.

Table 22 – Specifications of the modules XBee and XBee-PRO.

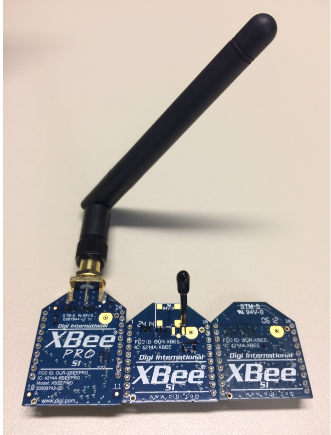
Specification	**XBee**	**XBee-PRO**
Indoor range	up to 30 m	up to 60 m
Outdoor range (line-of-sight)	up to 90 m	up to 750 m
Transmit power	1 mW (0 dBm)	10 mW (10 dBm)
Transmission rate	250 kbps	250 kbps
Receiver sensitivity	-92 dBm	-100 dBm
Supply voltage	2,8-3,4 V	2,8-3,4 V
Transmit current (typical)	45 mA (at 3.3 V)	antenna RPSMA 180 mA
Idle/Receive current (typical)	50 mA (at 3.3 V)	55 mA (at 3.3 V)
Power-down current	$< 10\ \mu$A	$< 10\ \mu$A

Source: Elaborated by the author.

The antennas are another important difference between XBee S1 and XBee-PRO S1. The available antennas are wire, printed circuit board (PCB) and Reverse-Polarity

Sub-Miniature version A (RPSMA), which can be seen in Figure 38. The PCB antenna has a more directional propagation, having worse or better performance depending on the relative position of the transmitter and receiver. On the other hand, the wire and RPSMA antennas have a more multi-directional propagation (DIGI, 2016).

Figure 38 – From left to right: RPSMA, wire and PCB antennas.



Source: Elaborated by the author.

XBee was chosen due to evidence that it is suitable for fly by wireless applications (AMINI; GILL; GAYDADJIEV, 2007; DAWSON *et al.*, 2008; OSSA-GOMEZ; MOARREF; RODRIGUES, 2011; OSSA-GÓMEZ; MOARREF; RODRIGUES, 2011). An important aspect analysed was the power consumption, considering that battery is a limited resource in embedded systems. In Stankunas, Rudinskas and Lasauskas (2011), a comparison between ZigBee, Bluetooth and WLAN protocols was carried out (see Table 23). Stankunas, Rudinskas and Lasauskas (2011) summarised the observed characteristics, such as the superior performance of ZigBee protocol when compared to others regarding energy consumption.

Table 23 – Bluetooth, WLAN and ZigBee specifications.

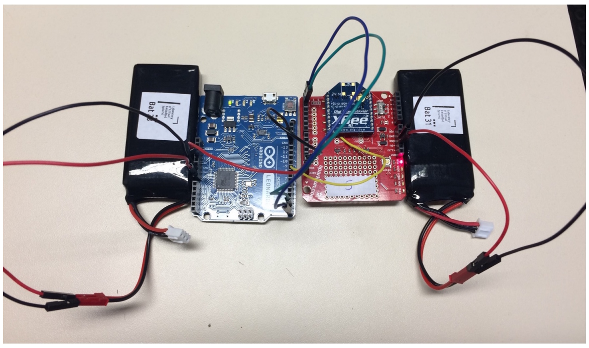
Name	**Range (m)**	**Network Topology**	**Transmission Rate (kbps)**	**Power (mW)**	**Bandwidth (MHz)**	**Module dimension (cm)**	**Estimated battery time**
Bluetooth	1-100	Ad hoc, point-to-point, star	2400	100	2400	31x16x2.2	days-months
WLAN	300	Mesh, ad hoc, star	11000	100	2400	10x10x1	days
ZigBee	up to 400	Mesh, ad hoc, star	250	30	2400, 868, 915	28x18x2	6 months-2 years

Source: Elaborated by the author.

According to Dementyev *et al.* (2013), an important aspect is the cyclic sleep provided by XBee, which adds the possibility to activate or deactivate wireless communication modules as needed. Among ZigBee (which is implemented by XBee), Bluetooth and ANT, only ZigBee has cyclic sleep function implemented. In addition to the facility to use cyclic sleep, the communication between nodes can also be quite simple in XBee.

Regarding power consumption measurement, it was necessary to find a way of decoupling the XBee from the expansion shield to measure the current from XBee ports independently from the Arduino Leonardo board. The decoupled scenario can be seen in Figure 39.

Figure 39 – XBee isolated from Arduino Leonardo for current measurement.



Source: Elaborated by the author.

To monitor Xbee voltage drop during the experiment, the configuration is shown in Figure 40a. A schematic of the circuit used to perform the voltage measurement can be seen in Figure 40b. In this schematic, the black board on the top left is the XBee.

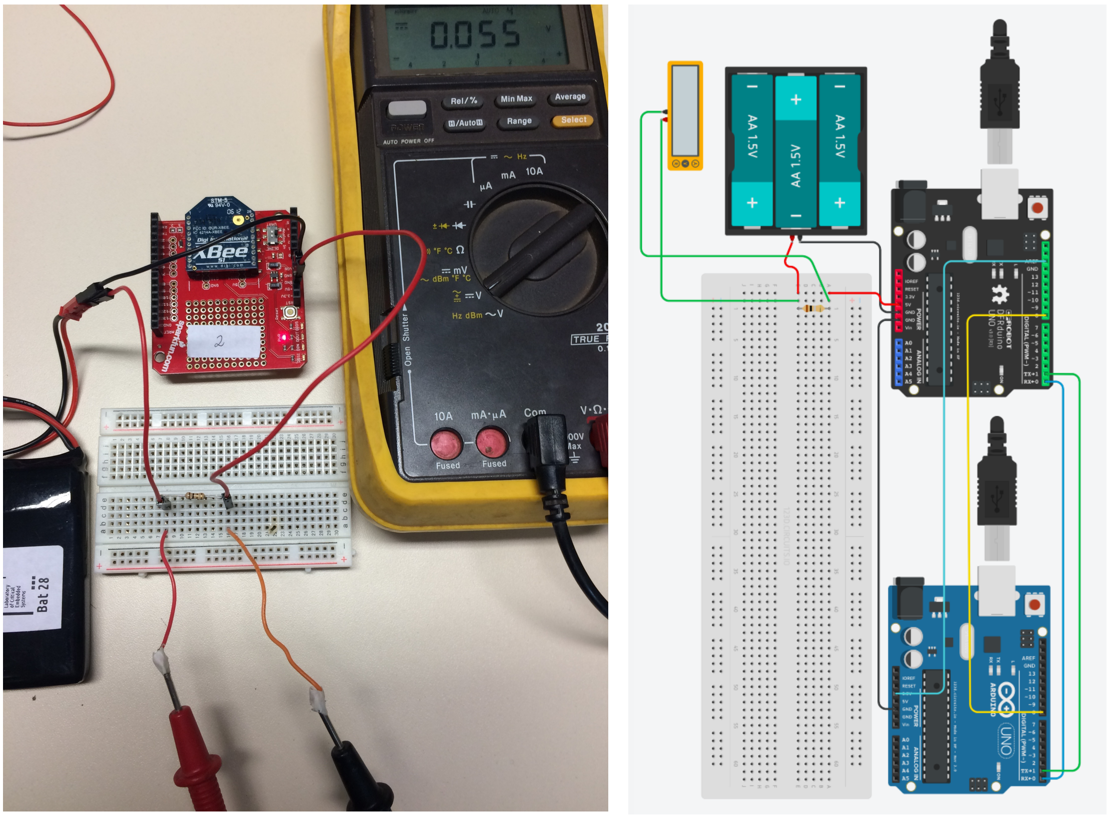
Table 24 presents smaller set of Navigation Phases for real experiments. It describes the message size that each node must send and the time (in seconds) of each phase. Message sizes are shown in Table 25.

Table 24 – Definition of a mission that contemplates different Navigation Phases. An individual packet size was defined for each and different time durations for each phase.

Navigation Phases	**Main nodes**			**Mission Nodes**		**Duration (s)**
	N1	**N2**	**N3**	**N4**	**N5**	
1.1	1	1	1	1	1	40
2.1	2	2	2	0	0	30
2.2	4	4	4	0	0	10
2.3	4	4	4	0	0	6
3.1	4	4	4	0	0	2
3.2	4	4	3	5	5	30
3.3	2	2	2	5	5	60
3.4	4	4	4	0	0	20
4.1	4	4	4	0	0	20
4.2	4	4	4	0	0	6
4.3	2	2	2	0	0	30
5.1	2	2	2	2	2	40
5.2	3	3	3	5	5	30

Source: Elaborated by the author.

Figure 40 – (a) Setup of the circuit used to measure the voltage drop; (b) Schematic of the circuit used to measure the voltage drop.



(a)

(b)

Source: Elaborated by the author.

Table 25 – Messages size in number of characters.

Reference	**Size in characters**
Off (0)	0
Low (1)	4
Medium-low (2)	8
Medium (3)	16
Medium-high (4)	32
High (5)	64

Source: Elaborated by the author.

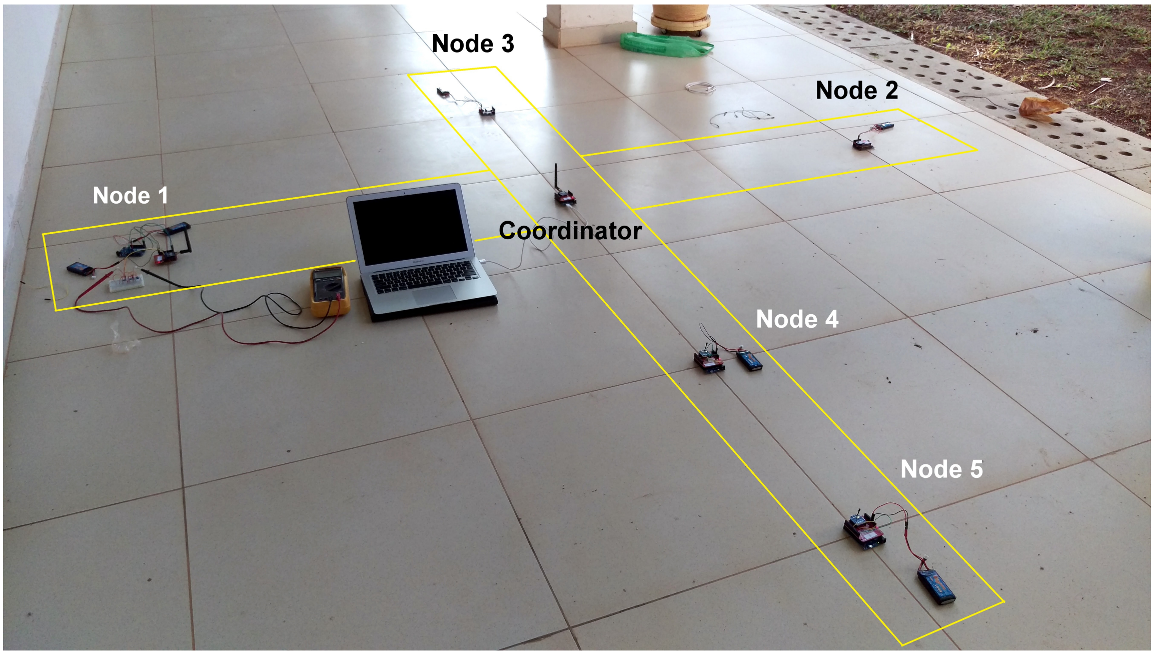
The Arduino program is divided in two stages: the setup that runs only once when the microcontroller is turned on and the loop stage that runs indefinitely afterwards. These specific programs for each node were recorded in the Arduino Leonardo boards, defining the behaviour of each XBee. The communication between Arduino Leonardo and XBee was performed by a serial communication. The setup stage is used to configure XBee, to set the serial communication between Arduino and the computer, to define the interruption parameters and to define Arduino pins behaviour as inputs or outputs.

The coordinator node and all the XBee boards were set with the same network ID. Moreover, the coordinator was defined as the network coordinator and its transmission mode was set as broadcast, because the only message sent by the coordinator is the command to begin the experiment.

The routine performed by the coordinator node is to receive all packets sent by all the other nodes and count how many packets have been received from each one. Main Nodes (1, 2 and 3) were configured similarly, sending unicast messages to the coordinator. In contrast, nodes 4 and 5 remain inactive during some phases. This is achieved thanks to the cyclic sleep feature that is implemented by *TimerOne.h* library for Arduino. All nodes start to operate at the same time as soon as the coordinator commands. The difference lies in the routine that each node performs.

Messages were sent with a frequency of 2 Hz by defining a delay of 500 ms. A total of 6 experiment replications were carried out. In each replication, nodes 1, 2 and 3 sent 648 messages each, while nodes 4 and 5 sent 400 messages.

Figure 41 – Experimental setup following similar dimensions of a UAV.



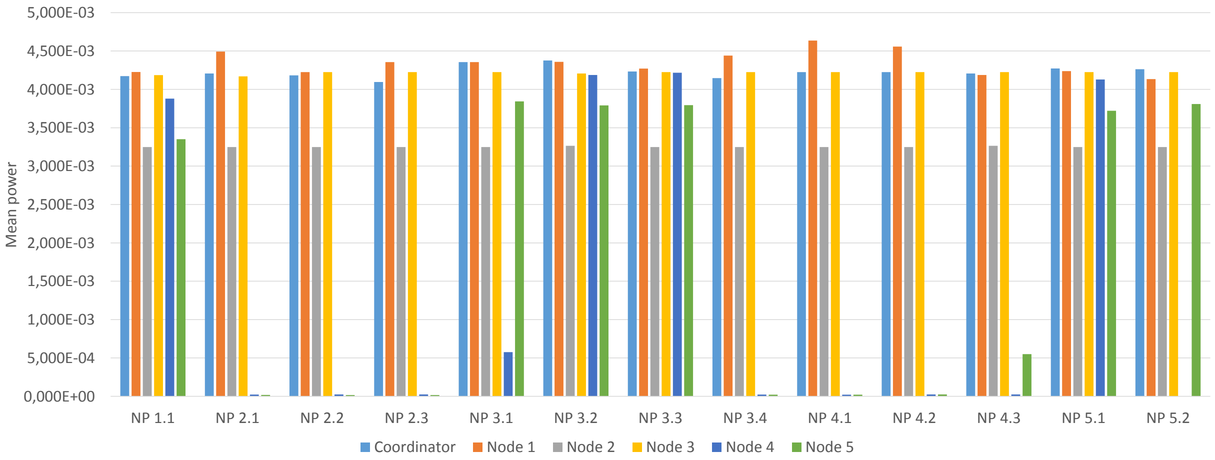
Source: Elaborated by the author.

Figure 41 identifies each node. The coordinator and node 1 were equipped with a XBee-PRO S1 and RPSMA antennas; nodes 2 and 3 with a XBee S1 and PCB antennas; and nodes 4 and 5 with XBee S1 and wire antennas.

Samples were collected every 4 seconds, totalling 81 measures for each node.

The average power consumed by each node is shown in Figure 42. One can say that, in general, the coordinator node and node 1 spend more energy, which is expected because both are equipped with an XBee-Pro S1. In addition, generally node 1 had higher energy consumption than the coordinator. Nodes 4 and 5 performed similarly to node 3 in energy consumption. Node 2 had a smaller consumption than node 3, even with the fact that both nodes were equipped with the same XBee board and antenna.

Figure 42 – Average power in each Navigation Phase.



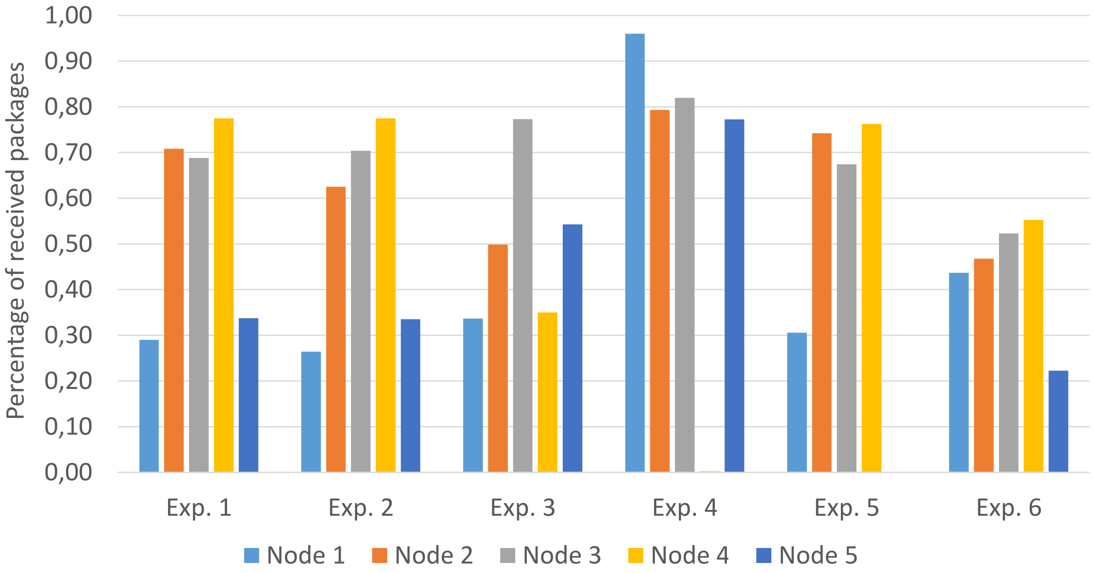
Source: Elaborated by the author.

Messages size had no relevant influence on energy. On the other hand, inactive XBee periods could be clearly seen and measured. Nodes 4 and 5 were inactive during 39% of the total time. Therefore, if nodes had remained active, both would spend approximately 93% more energy.

Figure 43 shows the percentage of messages received by the coordinator in each experiment. It was calculated based on the quantity of messages sent by nodes.

Analysing the data packets received by the coordinator, it can be concluded that about 50% of the packets were lost. Moreover, node 3 had fewer packet loss (30%) due to its position closer to the coordinator.

Figure 43 – Percentage of received packets by the coordinator in each experiment.



Source: Elaborated by the author.

6.4 Final remarks

This chapter presented results related to fly by wireless with Navigation Phases platform provided by HAMSTER. First, simulated experiments were discussed which proved that it is possible to explore scenarios with an elevated quantity of nodes. However, experiments on a real prototype were also carried out to identify how different XBee boards, antennas and the possibility of inactivate idle nodes interferes on energy consumption.

Although Navigation Phases platform provides a small contribution to green solutions, it was created and made independent in HAMSTER architecture aiming at future improvements that are expected to be seen in unmanned vehicles development. This platform must be further explored in future works, considering more detailed operation patterns. Indeed, the Navigation Phases platform will perform better and be more relevant if carefully designed for each application and system.

Lastly, case studies to validate NIMBLE platform were carried out and are presented in next chapter.

CHAPTER 7

CASE STUDIES ON NIMBLE

7.1 Chapter overview

It is essential that communications carefully meet mobility and time requirements, increasing the system overall capabilities and, consequently, allowing unmanned vehicles to be certified and integrated into their operation space. This chapter provides results that validate M2M and M2I communications through two case studies: Section 7.2 provides results on mobility and Section 7.3 carries out a comparative study on communication quality of service to provide safe FANETs. These results were published in: Munhoz *et al.* (2016) and Marconato *et al.* (2016). Part of these results were recently extended as a full paper submitted to the Journal of Communications in Computer and Information Science (as it will be listed in the Conclusions).

7.2 Case study H: performance evaluation of handoff in Mobile IPv6 networks with NIMBLE

The goal of this case study is to compare two handoff algorithms in IPv6 (Internet Protocol version 6) networks, especially investigating the impact of mobility support.

7.2.1 Background

UVs are becoming highly connected for cooperation purposes (e.g. distributed tasks) and also to the Internet for real-time services provision (e.g. IoT and cloud-based applications). Thus, a tendency on these areas is the use of IPv6 protocol that allows an exclusive address on the Internet and provides mobility approaches suitable for UVs applications.

Therefore, the study of handoff process on IPv6 is an important research topic for applications that depend on wireless mobile networks, mainly due to the fact that it is a critical aspect regarding connection quality that is impacted by packet losses during handoff transition process. Thus, it is necessary to analyse how worth the handoff process is considering network conditions (there might exist a better network to choose to connect to) and the criticality of an ongoing operation. Moreover, when it comes to high critical applications, handoff might be a failure point to be considered.

Contributions of this case study extend to embedded systems connected by IPv6 mobile networks, providing data and comparisons that can be used by developers and researchers. We analyse aspects such as run time in each step of the process and factors that influence the decision-making algorithm, such as signal strength and data transmission rate.

The advantage of wireless mesh networks is seen when self-organisation is coupled with seamless handover to provide continuity of service to the users. Handover (sometimes seen as handoff) is common in cellular networks, where mobile stations frequently move out of the coverage area of one cell tower and into that of a neighbouring tower (GUPTA; JAIN; VASZKUN, 2016).

In Mishra, Shin and Arbaugh (2003), authors measured latencies of all handoff process stages. They concluded that: i) the wireless network adaptor used in both the Mobile Unit (MU) and the Access Points (APs) directly influences the handoff latency; ii) different MUs treat sequences of messages slightly different; and iii) the search phase takes about 90% of the handoff process time. However, authors did not explore real situations even with simulated results, which might change the perception if contextualised in some areas.

In Vatn (2003), authors discussed the effects of handoff in data transmission. Similarly to Mishra, Shin and Arbaugh (2003), they concluded that the network interface affects latency and the longest stage of the process is search. They have also noted that a MU can keep receiving data from previously connected AP even during the search phase. Finally, authors concluded that the behaviour of handoff process depends not only on the hardware used (network adaptor), but also on data stream, e.g. if MU is sending or receiving packets.

In Chuang and Lee (2011), authors discussed latency issues in Mobile IPv6 and its derivative Proxy Mobile IPv6, highlighting the difficulty of using such handoff mechanisms in real-time systems. To solve this problem, authors proposed a new handoff scheme for Proxy Mobile IPv6 network that reduces latency and solves the loss problem.

We have identified that there is a lack of comparisons between IPv6 protocol with and without mobility on the context of unmanned aerial vehicles (UAVs). Thus, in this chapter we will carry out a discussion on how handoff process impacts IPv6 networks from the point of view of unmanned aircraft systems (UAS) in general, which includes not only the UAV, but all the supporting systems.

Next subsection will address important concepts review on IPv6.

7.2.2 Internet Protocol version 6

In 1990s, the Internet Engineering Task Force started developing the successor to the Internet Protocol version 4 (IPv4). The motivation for that was the fact that 32-bit IP address was beginning to be used up as more devices were connected to the Internet with unique IP addresses. To respond for the need of a large IP address space, a new IP protocol was developed. The Internet Protocol version 6 (IPv6) has also improved some aspects based on accumulated operational experience with IPv4 (KUROSE; ROSS, 2009). There was considerable debate about when the IPv4 addresses would be completely allocated. Many approaches were developed since then trying to prolong the use of IPv4 as much as possible. However, we have been currently seeing major companies and institutions moving to IPv6, but the transition may still take some time to be fully completed.

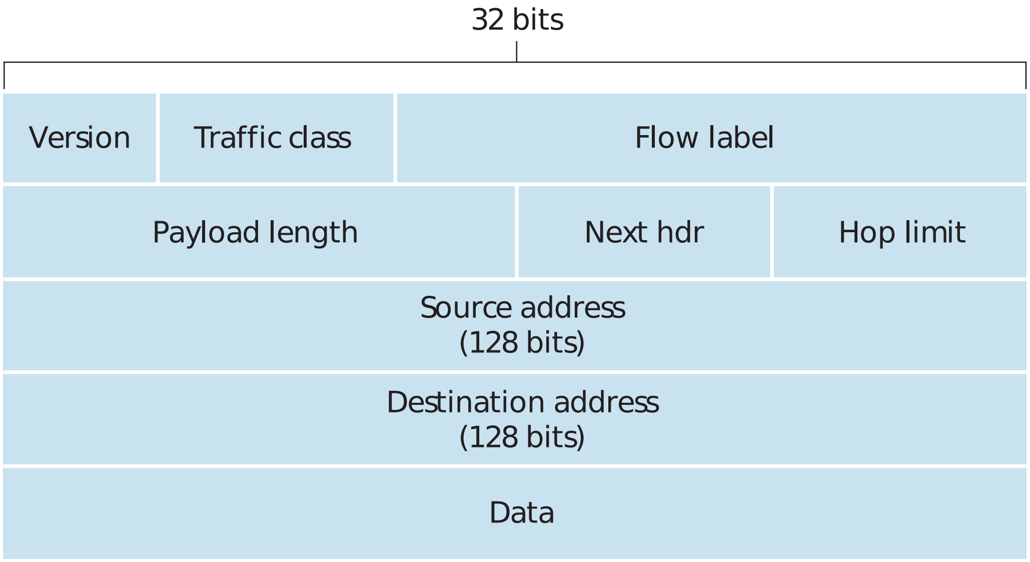
According to Kurose and Ross (2009), the most important changes in IPv6 datagram (Figure 44) format are:

- **Expanded addressing capabilities**. IPv6 increases the size of the IP address from 32 to 128 bits, ensuring that there will be enough IP addresses for nowadays applications. As a comparison matter, every grain of sand on the planet can be IP-addressable.
- **A streamlined 40-byte header**. The 40-byte fixed-length header allows for faster processing of the IP datagram. A new encoding of options allows for more flexible options processing.
- **Flow labelling and priority**. IPv6 has an elusive definition of a flow, e.g. audio and video transmission might likely be treated as a flow. On the contrary, the more traditional applications might not be treated as flows. Thus, the designers of IPv6 foresee the eventual need to be able to differentiate among the flows.

The following fields are defined in IPv6 (KUROSE; ROSS, 2009) (see Figure 44):

- **Version**. This 4-bit field identifies the IP (Internet Protocol) version number.
- **Traffic class**. This 8-bit field is similar in spirit to the Type of Service in IPv4.
- **Flow label**. This 20-bit field is used to identify a flow of datagrams.

Figure 44 – IPv6 datagram.



Source: Adapted from Kurose and Ross (2009).

- **Payload length**. This 16-bit value is treated as an unsigned integer giving the number of bytes in the IPv6 datagram following the fixed-length, 40-byte datagram header.

- **Next header**. This field identifies the protocol to which the contents (data field) of this datagram will be delivered (for example, to Transmission Control Protocol (TCP) or User Datagram Protocol (UDP)).

- **Hop limit**. The contents of this field are decremented by one by each router that forwards the datagram. If the hop limit count reaches zero, the datagram is discarded.

- **Source and destination addresses**. The various formats of the IPv6 128-bit address are described in RFC 4291 (HINDEN; DEERING, 2006).

- **Data**. This is the payload portion of the IPv6 datagram. When the datagram reaches its destination, the payload will be removed from the IP datagram and passed on to the protocol specified in the next header field.

Mobile IPv6 (MIPv6) describes the protocol operations needed to keep a mobile node connected to the Internet during its handover from one access router to another. These operations involve movement detection, IP address configuration and location update (RFC 4068) (KOODLI, 2005). Mobile IPv6 is an Internet Engineering Task Force standard that has added the roaming capabilities of mobile nodes in IPv6 network (RFC 3775) (JOHNSON; PERKINS; ARKKO, 2004).

The major benefit of this standard is that the mobile nodes (as IPv6 nodes) change their point-of-attachment to the IPv6 Internet without changing their IP address (DAS, 2017). Moreover, MIPv6 is an update to Mobile IP (RFC 6275) (PERKINS; JOHNSON; ARKKO, 2011), designed to authenticate mobile devices using IPv6 addresses.

In traditional IP network routing, addresses represent a topology. The routing mechanisms were made under the assumption that each network node has always the same entry point to the Internet and that each IP address identifies the link to which it is connected. MIPv6 allows a mobile node to transparently maintain connections as it moves from a network edge to another.

7.2.3 Methodology

As mentioned in Chapter 3, every HAMSTER version has four elements: NCI, NP, SPHERE and NIMBLE. NIMBLE is the platform for mobility is important for aspects regarding external communications, as detailed in Section 3.7. Handoff is one of the biggest challenges for critical embedded systems and will be investigated with experiments on a well-accepted simulator. This subsection presents the simulator, the criteria and the results collection methods.

7.2.3.1 OMNeT++

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators (OMNeT++ Discrete Event Simulator, 2017). Due to its general structure, it can be used for analysis and study of different problems, including:

1. Modelling wired/wireless communication networks;
2. Protocols;
3. Architecture validation and distributed systems modelling;
4. Performance evaluation of complex systems;
5. Modelling and simulation of any system in which the approach to discrete event is appropriate and entities can be conveniently mapped.

OMNeT++ provides a component architecture for models. Components (modules) are programmed in C++, then assembled into larger components and models using a high-level language. Reusability of models comes for free. OMNeT++ has extensive GUI (Graphical User Interface) support and, due to its modular architecture, the simulation kernel (and models) can be embedded easily into applications (OMNeT++ Discrete Event Simulator, 2017).

In addition to OMNeT++, INET (INET, 2017) is also used in this case study. INET Framework is an open-source model library for the OMNeT++ simulation environment. It provides protocols, agents and other models for researchers and students working with communication networks. INET is especially useful when designing and validating new protocols, or exploring new or exotic scenarios.

INET contains models for the Internet stack (TCP, UDP, IPv4, IPv6, Open Shortest Path First, Border Gateway Protocol, etc.), wired and wireless link layer protocols (Ethernet, Point-to-Point Protocol, IEEE 802.11, etc), support for mobility, MANET protocols, DiffServ, Multiprotocol Label Switching with Label Distribution Protocol and Resource Reservation Protocol - Traffic Engineering signalling, several application models and many other protocols and components. Several other simulation frameworks take INET as a base and extend it into specific directions, such as vehicular networks, overlay/peer-to-peer networks, or Long-Term Evolution (INET, 2017).

OMNeT++ simulator was chosen to run the simulations that will be discussed in Section 7.2.4 because it is openly distributed and well accepted. We have implemented two simulations using INET Framework and OMNeT++ to compare distinct handoff processes: with and without mobility.

Next subsection provides details on criteria and results collection.

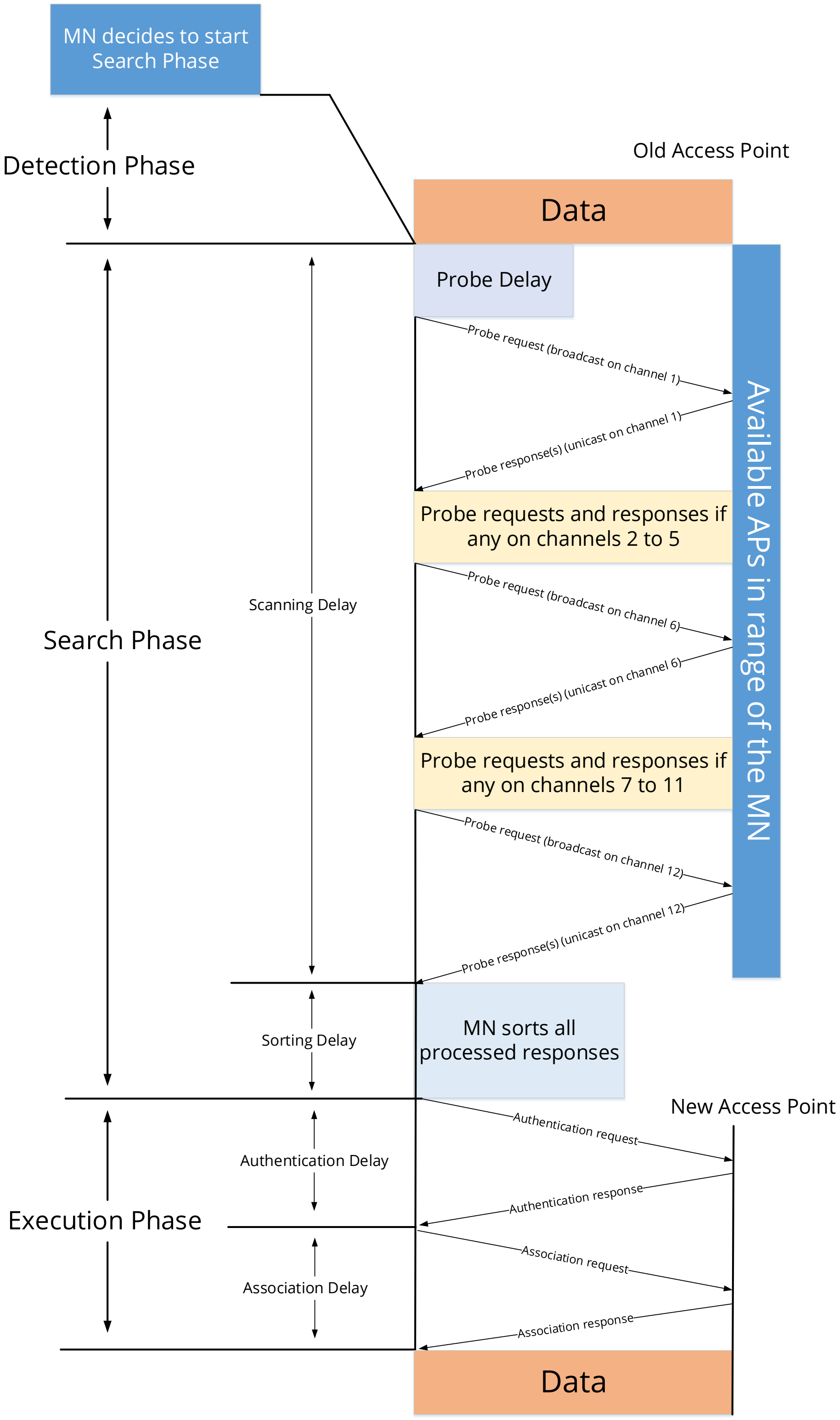
7.2.3.2 Criteria and results collection

First, we have investigated a way of making simulations as similar as possible. For instance, the speed at which the MU would move, plus routers distance and coverage area must be the same in both simulations, in order not to compromise results. Criteria used for comparison between algorithms are: process time and signal strength.

The handoff run time criterion is perhaps the most complex and that best defines the difference between processes. For some algorithms, it is highly important for a handoff process to be performed as fast as possible due to the fact that the MU needs to disconnect from previous AP to start sending requests to the next one. Packet losses may occur due to disconnected time (TANENBAUM; WETHERALL, 2011). For example, if there is a real-time video transmission, packets will likely be lost causing issues.

The handoff process can be divided into two phases (NANKANI, 2005), as shown in Figure 45. For further analysis, the run time is measured for each step separately, allowing a step by step comparison. The first phase is defined as search. The MU scans for nearby available APs. Once found, the AP in better conditions according to the algorithm criteria, is the one chosen by MU for connection establishment.

Figure 45 – Phases of handoff process.


MN decides to start Search Phase
Detection Phase
Old Access Point
Data
Probe Delay
Probe request (broadcast on channel 1)
Probe response(s) (unicast on channel 1)
Probe requests and responses if any on channels 2 to 5
Probe request (broadcast on channel 6)
Probe response(s) (unicast on channel 6)
Probe requests and responses if any on channels 7 to 11
Probe request (broadcast on channel 12)
Probe response(s) (unicast on channel 12)
Available APs in range of the MN
Scanning Delay
Search Phase
Sorting Delay
MN sorts all processed responses
New Access Point
Authentication request
Authentication response
Authentication Delay
Execution Phase
Association request
Association response
Association Delay
Data

Source: Adapted from Nankani (2005).

Search phase can be divided into two steps: scanning delay and classification delay. Still, in scanning delay, it is possible to scan each channel separately. In practice, the initial event of the search phase is Beacon Timeout, i.e., the MU has received at least three Beacon packets that were considered as noise, representing a connection lost state, which leads to a new search for APs. In the next event, the MU disassociates from AP1 and sends an internal command to tune the channel 0, starting a new search. The scanning delay ends in the event when the MU radio can tune channel 0 and then start the effective search.

The next step consists on scanning each channel in order to find an AP. This scan takes place as follows: the MU tunes to a certain channel and listens for incoming messages from any AP in that frequency, should it be a Beacon or a Router Advertisement. Such listening process lasts a preset time. After that, the MU is aware whether there is any AP operating in that frequency that can be include on the list of new AP candidate. If a channel has been scanned and no message was detected, the radio warns the management layer that the time limit is over and no AP was found. A command for channel change is also triggered.

After the minimum scanning time and the positive identification of another device operating on the same frequency, the MU sends a Probe Request message, asking for information about how the AP works, e.g. transmission type, transmission rate. The AP then responds with a Probe Response message. Later, the MU updates the known AP list and proceeds with the scanning process. This process is repeated for all four possible channels. After that, it terminates the scanning delay process. For comparison matters, search times will be measured on each channel, apart from the initial scan phase.

The last stage of the search phase is the classification delay, in which the MU classifies all found APs. A limitation imposed by our simulator is seen in this phase. Although it simulates these events as concomitant processes, they are not so in reality. Thus, the classification delay cannot be measured in details, but its overall time is included in results and does not imply problems for our analyses.

The next phase of handoff process is called Execution Phase. Here, the MU exchanges messages with the AP in order to associate with it. Some of these messages are the authentication request, followed by its response, request and association response.

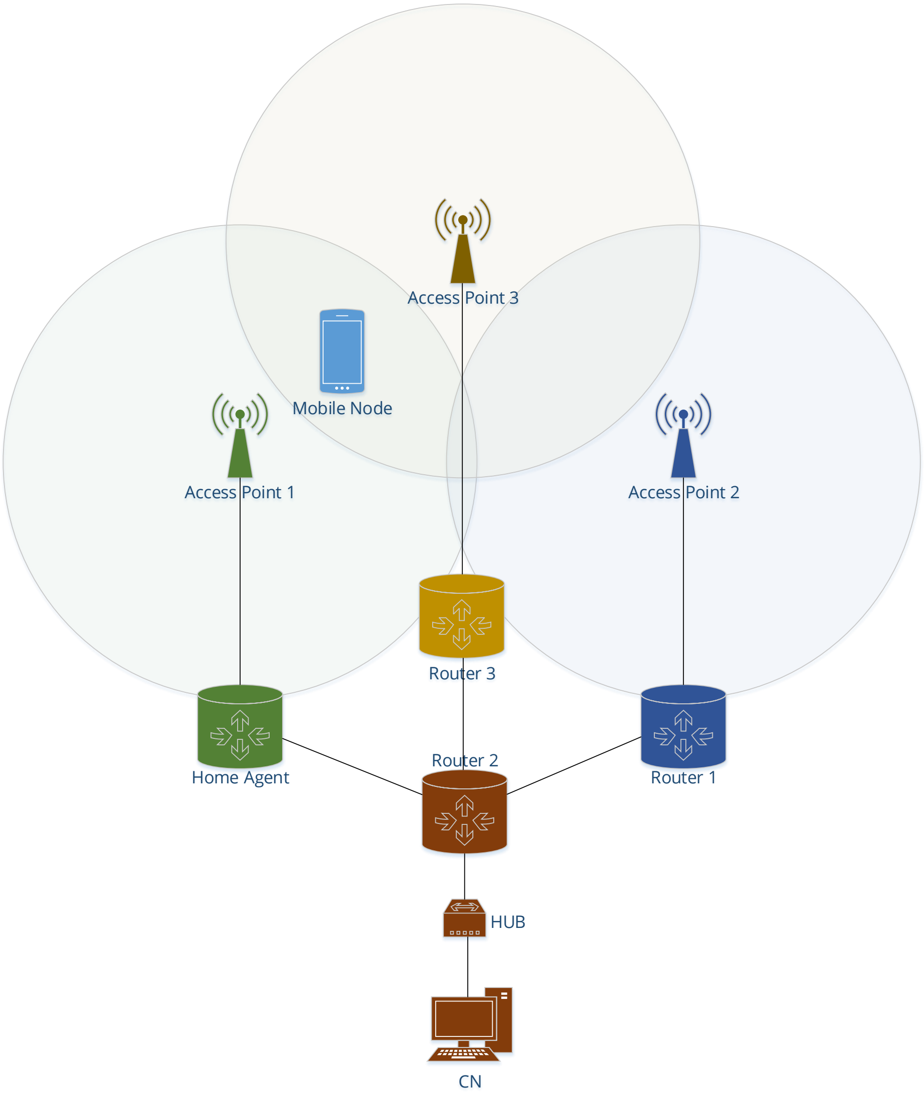
The received signal strength indicator of an AP is one of the decisive factors of a handoff process. When making the decision of which AP the MU will connect to, there must be a consideration about the best signal, which can potentially avoid new handoffs.

Changing the decision-making policy can be a tricky task, since it is not always possible to get access to codes that implement the direct process in real-time operating systems. In our experiments, such change is not possible due to simulator restrictions. However, we will carry out tests that help checking the influence of some factors in the decision-making algorithm used by OMNeT++ simulator in this chapter.

To analyse this by a different perspective, a new scenario has been proposed: the inclusion of a new AP to our simulation that addresses a scenario with no mobility. Thus, an area covered by three different APs was created. By changing parameters of any AP, it would be possible to discuss about which aspect was taken into consideration for the decision making.

The aspects we aim to analyse with this research are signal strength and data transfer rate. Basically, the positioning of all three routers forms an equilateral triangle, as shown in Figure 46. In each test, the parameter of one of the APs were modified in order to decrease one of its features. Afterwards, scenarios in Table 29 were proposed.

Figure 46 – Network scheme adopted for testing handoff decision making algorithm.



Source: Elaborated by the author.

Next subsection presents the experimental setup for experiments.

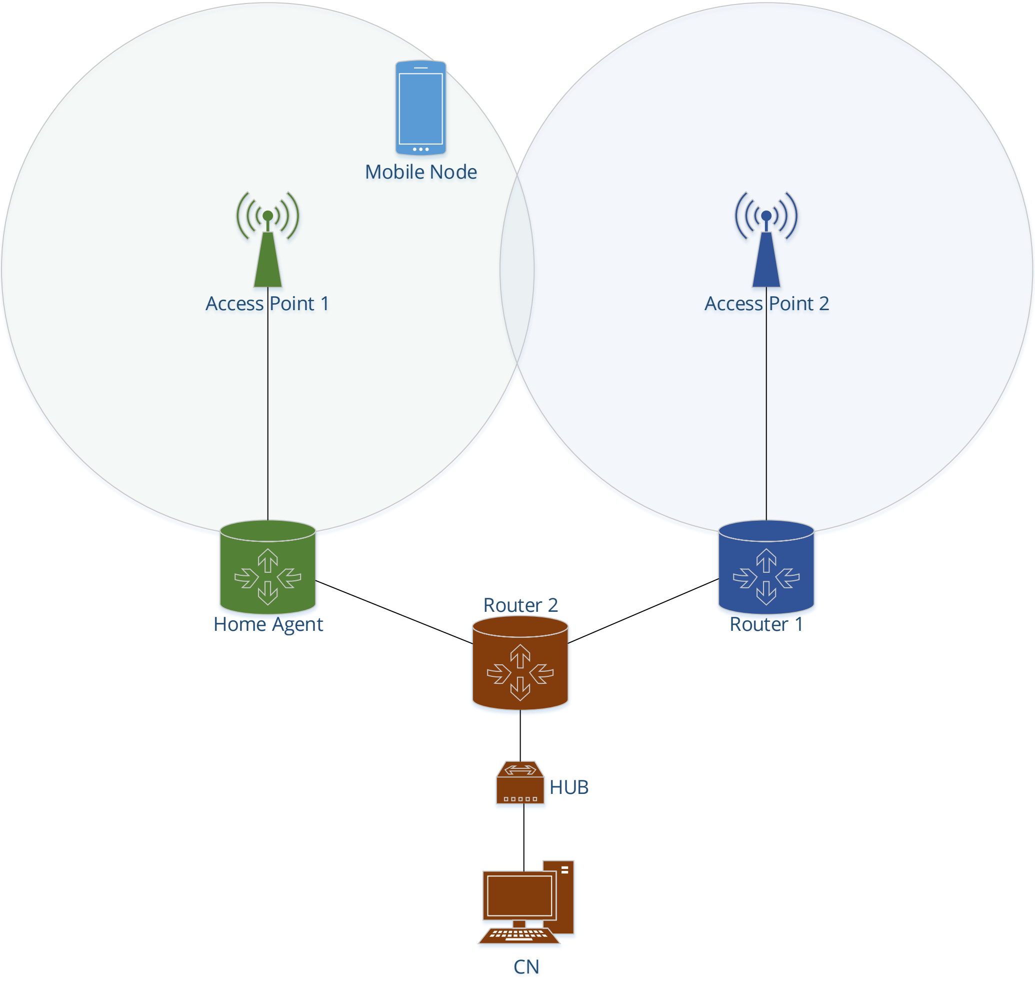
7.2.4 Experimental setup

7.2.4.1 IPv6 with Mobility

The "IPv6 with Mobility" simulation (*MIPv6Network.ned*) is part of the INET simulations package (INET, 2017). This simulation allows the analysis of handoff process on an IPv6 network with mobility.

The MIPv6Network is composed by five basic elements: mobile unit, access points, routers, hub and fixed host. In addition, there are two elements in charge of configuration. Figure 47 illustrates the scenario

Figure 47 – Graphical representation of MIPv6Network.



Source: Elaborated by the author.

The MU is composed by a *WirelessHost6* module. It is the MU, which is always connected to an AP and moves among coverage areas. The APs used in the simulation are both equal. They provide radio for routers i.e. wireless communication between MU and Fixed Host.

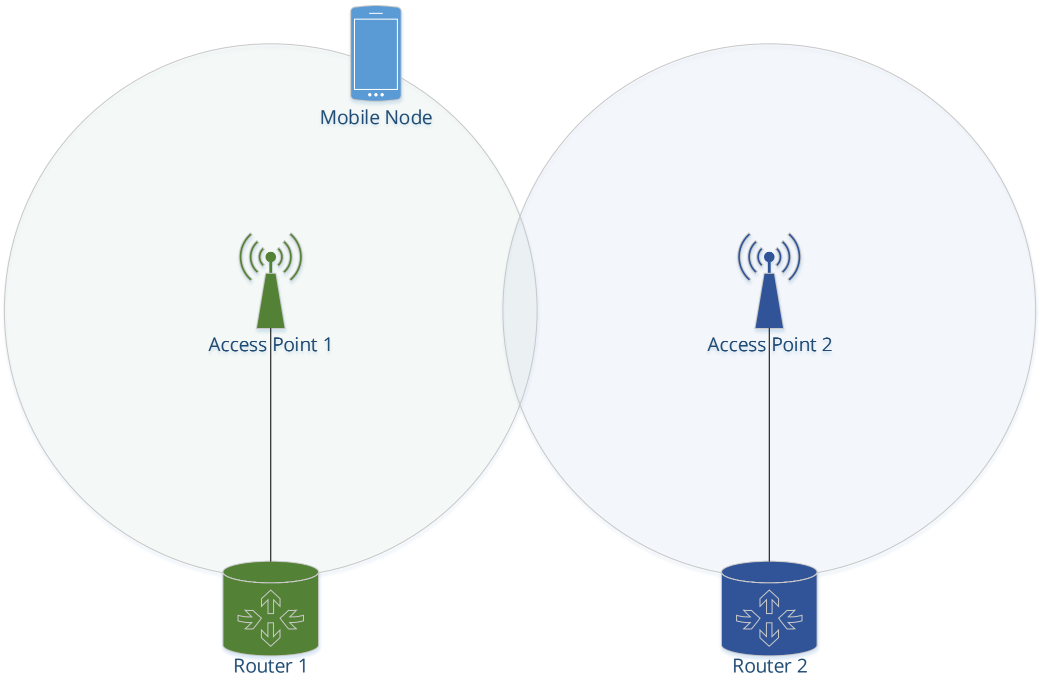
The first router module, identified as ***Router*6**, acts as Home Agent. This router captures packets for the MU should it be connected or not. The Router called *R_1* is the Foreign Agent, i.e. it distributes the packets generated by the MU while outside the original network and acts receiving the tunnelled packets destined to MU. The other Router along with the Hub represents the Internet, providing the tunnel between Routers that provide connectivity.

7.2.4.2 IPv6 without Mobility

We have developed the “IPv6 without Mobility” simulation (*handoff_ipv6.ned*) in order to meet the requirements we are investigating. This IPv6 network performs experiments of handoff without mobility. As the MU moves from one network to another, considering there is no support for mobility, it completely loses its link with the native address, including a subsequent IP address change.

Figure 48 illustrates the simulation. One can point out that the network is formed by a *WirelessHost6* MU that supports MIPv6, but it has such a feature disabled. As seen in “IPv6 with Mobility” simulation, two APs are used for wireless communication between routers. They are both independent routers that provide distinct links. This is a way of guaranteeing that no mobility is used in our experiments.

Figure 48 – Graphical representation of handoff_ipv6 network.



Source: Elaborated by the author.

In addition to the network elements, two configuration modules were used: *Chan-*

nelControl (for managing wireless communications, including distances and possible interferences) and *flatNetworkConfigurator6* (which manages addresses issues and routing tables).

Next subsection presents results and discussions.

7.2.5 Results and discussions

7.2.5.1 Run time results

The time needed for a MU to either interpret three lost *Beacons* and decide to start a new search, cannot be calculated with our simulator. Thus, it will not be considered for comparison purposes, since it does not affect the results integrity.

Table 26 presents the measured run times of total scanning of frequency channels during handoff process in both simulations. The factor which was more decisive on run time was the number of scans that the MU had to perform before finding an available AP. This difference is noticeable in scenarios where there is intersection between the signal coverage areas (Scenarios 1 and 2) than those with no intersection (Scenarios 2 and 3). If there is no intersection, as the MU loses the first AP signal, it initiates a search on all channels. If this time is not enough for the MU to enter the area of another AP, it will have to perform other complete search. Repeating these searches is a very costly process that also compromises the efficiency of the handoff process, since each additional search takes 1.25 s. In a scenario of high transmission rates, e.g. in a video/audio streaming, a new search might fully compromise the service, as it would mean 1.25 s disconnected from the AP, meaning a loss of thousands of packets.

Table 26 – Total scanning time for the tested scenarios.

	Scenarios	**1st exp.**	**2nd exp.**	**3rd exp.**	**Scanning repetitions**
IPv6 with Mobility	1	1 s 400 ms	1 s 400 ms	1 s 400 ms	1
	2	1 s 400 ms	1 s 400 ms	1 s 400 ms	1
	3	2 s 650 ms	2 s 650 ms	2 s 650 ms	2
	4	6 s 400 ms	6 s 400 ms	6 s 400 ms	5
IPv6 without Mobility	1	1 s 400 ms	1 s 400 ms	1 s 400 ms	1
	62	1 s 400 ms	1 s 400 ms	1 s 400 ms	1
	3	2 s 650 ms	2 s 650 ms	2 s 650 ms	2
	4	6 s 400 ms	6 s 400 ms	6 s 400 ms	3

Source: Elaborated by the author.

Still, in Table 26, one can point out that the velocity of a mobile node directly impacts the handoff time on a situation in which there is no intersection of cells. As the

MU moves faster through the area without signal, it needs to perform fewer scans and therefore connect faster to the new AP.

In the second phase of handoff process, it is performed authentication and association of the MU with the AP. The authentication delay times are shown in Table 27 and the association delay times are shown in Table 28.

Table 27 – Authentication delay times for the tested scenarios.

	Scenarios	1st exp.	2nd exp.	Mean time
IPv6 with Mobility	1	2 ms 375 us 943 ns	2 ms 375 us 943 ns	2 ms 375 us 943 ns
	2	2 ms 376 us 59 ns	2 ms 376 us 59 ns	2 ms 376 us 59 ns
	3	2 ms 376 us 188 ns	2 ms 376 us 188 ns	2ms 376 us 188 ns
	4	2 ms 403 us 214 ns	2 ms 403 us 214 ns	2 ms 403 us 214 ns
IPv6 without Mobility	1	2 ms 349 us 88 ns	2 ms 349 us 88 ns	2 ms 349 us 88 ns
	2	2 ms 403 us 226 ns	2 ms 403 us 226 ns	2 ms 403 us 226 ns
	3	2 ms 402 us 582 ns	2 ms 402 us 582 ns	2ms 402 us 582 ns
	4	2 ms 402 us 609 ns	2 ms 402 us 609 ns	2 ms 402 us 609 ns

Source: Elaborated by the author.

The next handoff phase that impacts the total run time is the authentication/association process. In the case of authentication, there is a difference in microseconds among all tested scenarios. In both scenarios, the IPv6 protocol without mobility performed better than MIPv6 (Scenarios 2 and 3). On the other hand, MIPv6 performed better in the other two. In association phase, MIPv6 took advantage in three scenarios, not being the best option only in the case of travel speed and no cells intersection. This result indicates that MIPv6 can facilitate the MU association process.

Table 28 – Association delay times for the tested scenarios.

	Scenarios	1st exp.	2nd exp.	Mean time
IPv6 with Mobility	1	1 ms 523 us 971 ns	1 ms 523 us 971 ns	1 ms 523 us 971 ns
	2	1 ms 506 us 29 ns	1 ms 506 us 29 ns	1 ms 506 us 29 ns
	3	1 ms 497 us 94 ns	1 ms 497 us 94 ns	1 ms 497 us 94 ns
	4	1 ms 507 us 70 ns	1 ms 507 us 70 ns	1 ms 507 us 70 ns
IPv6 without Mobility	1	1 ms 479 us 44 ns	1 ms 479 us 44 ns	1 ms 479 us 44 ns
	2	1 ms 497 us 113 ns	1 ms 497 us 113 ns	1 ms 497 us 113 ns
	3	1 ms 496 us 805 ns	1 ms 496 us 805 ns	1 ms 496 us 805 ns
	4	1 ms 523 us 387 ns	1 ms 523 us 387 ns	1 ms 523 us 387 ns

Source: Elaborated by the author.

Finally, Table 29 shows the total handoff process run time in all eight tested scenarios.

It is important to point out that IPv6 without mobility performed better and faster in three scenarios. That is due to an overhead imposed by the inherent operation of Mobile

Table 29 – Total handoff time for the tested scenarios.

	Scenarios	Total time
IPv6 with Mobility	1	1 s 403 ms 899 us 971 ns
	2	1 s 403 ms 882 us 88 ns
	3	2 s 653 ms 873 us 282 ns
	4	6 s 403 ms 910 us 278 ns
IPv6 without Mobility	1	1 s 403 ms 828 us 132 ns
	2	1 s 403 ms 900 us 339 ns
	3	2 s 653 ms 899 us 207 ns
	4	6 s 403 ms 925 us 996 ns

Source: Elaborated by the author.

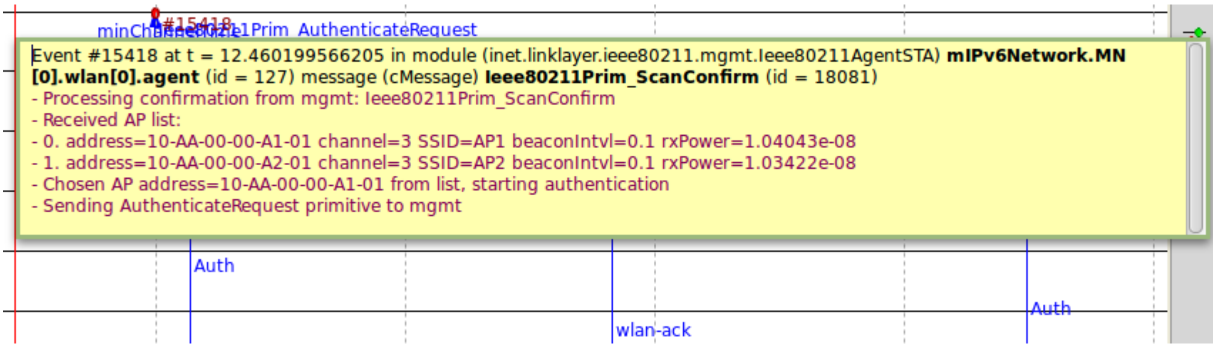
IPv6.

As a final remark, a few microseconds (less than 100 µs in all cases) do not considerably affect the amount of received bits. For example, in a 2 Mbps transmission, around 2 bits per millisecond (ms) are sent — or 0.002 bits per microsecond (µs). Differences of 100 µs result in the loss of 0.2 bit, which is extremely small considering the amount of transmitted information.

7.2.5.2 Decision making

In the first scenario for decision making evaluation, both APs are configured to emit signals with the same power. However, AP1 operated with half the AP2's data transmission rate. At the end of scan phase, the handoff algorithm has chosen to connect to AP1, as shown in Figure 49.

Figure 49 – Handoff classification phase on Scenario 1.

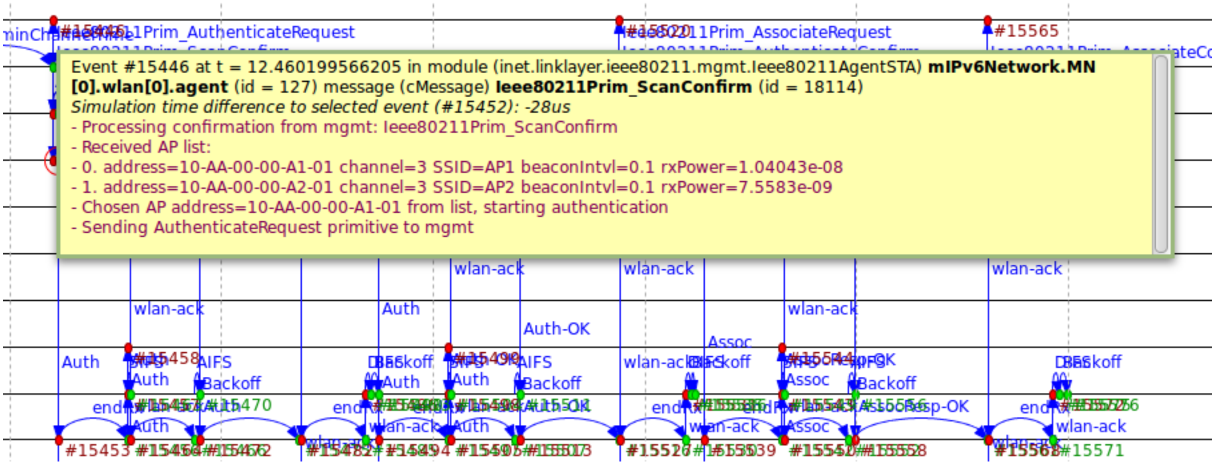


```
Event #15418 at t = 12.460199566205 in module (inet.linklayer.ieee80211.mgmt.Ieee80211AgentSTA) mIPv6Network.MN
[0].wlan[0].agent (id = 127) message (cMessage) Ieee80211Prim_ScanConfirm (id = 18081)
- Processing confirmation from mgmt: Ieee80211Prim_ScanConfirm
- Received AP list:
- 0. address=10-AA-00-00-A1-01 channel=3 SSID=AP1 beaconIntvl=0.1 rxPower=1.04043e-08
- 1. address=10-AA-00-00-A2-01 channel=3 SSID=AP2 beaconIntvl=0.1 rxPower=1.03422e-08
- Chosen AP address=10-AA-00-00-A1-01 from list, starting authentication
- Sending AuthenticateRequest primitive to mgmt
```

Source: Elaborated by the author.

The other analysed scenario contains two APs with the same data transmission rate, however different signal strength. The algorithm has chosen to associate with AP1, as shown in Figure 50.

Figure 50 – Handoff classification phase on Scenario 2.



Event #15446 at t = 12.460199566205 in module (inet.linklayer.ieee80211.mgmt.Ieee80211AgentSTA) **mIPv6Network.MN[0].wlan[0].agent** (id = 127) message (cMessage) **Ieee80211Prim_ScanConfirm** (id = 18114)
Simulation time difference to selected event (#15452): -28us
- Processing confirmation from mgmt: Ieee80211Prim_ScanConfirm
- Received AP list:
- 0. address=10-AA-00-00-A1-01 channel=3 SSID=AP1 beaconIntvl=0.1 rxPower=1.04043e-08
- 1. address=10-AA-00-00-A2-01 channel=3 SSID=AP2 beaconIntvl=0.1 rxPower=7.5583e-09
- Chosen AP address=10-AA-00-00-A1-01 from list, starting authentication
- Sending AuthenticateRequest primitive to mgmt

Source: Elaborated by the author.

Regarding tests on Decision Making, the main conclusion is that the predominant factor in choosing an AP to connect was the signal strength received by MU at the scanning process. In the test performed in Scenario 1, in which both APs operate with equal signal strength and AP1 operates with half the transfer rate of AP2, AP1 was chosen. This can be explained by the difficulty in obtaining an overall symmetry of the problem: as much as the APs are equally distributed, in the meantime while receiving the packet with information about each AP signal strength, the MU has already moved. That leads to slight differences in results, as shown in Figure 49. The predominance of signal quality explains the choice for AP1, despite the low transmission rate.

The experiment in the second scenario only confirms the conclusion aforementioned. When signal strength was far different, the chosen AP was always the one that offered a higher signal quality.

It is important to clarify that tests conducted to observe decision making were carried out in only one of the protocols mentioned in this chapter, the “IPv6 with Mobility” (Section 7.2.4.1). This choice was made due to the fact that the decision making algorithm is linked to the simulator itself, not to the protocols. Thus, for any handoff process in any OMNeT++ simulation, the decision making algorithm was the same. The purpose of these tests were to get more information about this algorithm and complement researches on handoff processes and also to draw conclusions that could help UAVs development and research.

As it could be seen in presented and discussed results, there is a small advantage of “IPv6 without Mobility” when compared to “IPv6 with Mobility”, specifically regarding handoff run time. Considering real-time systems, although seeming irrelevant, such a small difference can be very important. In some applications, such as aviation, critical embedded systems must have low failure rates, such as a serious failure every 10^5 to 10^9 hours of operation at most. Whereas delays in communication can lead to failures, this difference becomes significant.

Taking the example of scenarios where terrestrial or aquatic vehicles operate missions in isolated areas and UAVs are used to fly and collect/provide information for terrestrial and aquatic networks, the flyovers cannot be performed more than once depending on the local access conditions. In some cases, the UAV flight range limit is very short, requiring the flight to be performed with higher speed and only once. In this case, handoff process run time for connection among UAV and terrestrial/aquatic vehicles could significantly impact the overall system.

Another practical example is related to smart cars and roads. On highways with multiple APs, handoff operations happen frequently, which can lead to inherent delays that may affect both safety and entertainment operations. Thus, the mentioned small differences in run time can be important and should be taken into account on these systems design, especially for real-time applications.

7.3 Case study I: a comparison between IEEE 802.11n and IEEE 802.15.4 in regards of M2M and M2I communications to provide safe FANETs

This case study provides a comparison between two protocols regarding external communications managed by HAMSTER's NIMBLE. The analysis goes towards the provision of safe FANETs, which leads to intersections with HAMSTER's SPHERE.

7.3.1 Background

Advances in UAV technologies are allowing FANETs to become a reality. However, in order to achieve an effective cooperation among multiple UAVs, it is necessary to model distinct communication protocols. Basically, a FANET can be considered a robot ad hoc network (when no infrastructure is used) or a robot sensor network (when a CAGE is considered).

The main idea of FANET is to perform cooperative sensing, using multiple UAVs to cover an area that is not possible or viable with a single UAV. Thus, it is necessary to have a reliable communication and guarantee quality of service. There are several research efforts in robot sensor networks and several challenges are faced, such as: robot control, robot localisation and communication Quality of Service (QoS). Therefore, these challenges are very similar in FANETs.

In this case study, a FANET composed by multiple UAVs and a single CAGE will be analysed from HAMSTER's NIMBLE view point. The objective is to find out which FANET topology is more QoS effective. Thus, simulated FANET scenarios based on star network topology (all UAVs directly communicating with the CAGE) and mesh topology (a dynamic routing is necessary) were tested. The simulations were based on real scenarios traces where parameters like speed and mobility pattern were extracted from real-world experiments.

Communication is a crucial element for safety, which is considered part of dependability (HANMER; MCBRIDE; MENDIRATTA, 2007). The main reason is that UAVs must maintain messages exchange rates in order to coordinate a mission. Therefore, it is necessary to define a suitable UAV network topology that achieves the QoS level necessary to keep connectivity. Thus, a FANET composed of n UAVs and a CAGE was considered.

Routing algorithms play an important role in connectivity since broadcasting messages can generate unnecessary traffic on the network and traffic congestion. Thus, the efficient application of routing algorithms becomes a need to ensure connectivity and hence increasing the safety of the UAV and all the elements that compose an unmanned aircraft system, emphasising that FANET mobility patterns are very relevant.

7.3.2 Material and methods

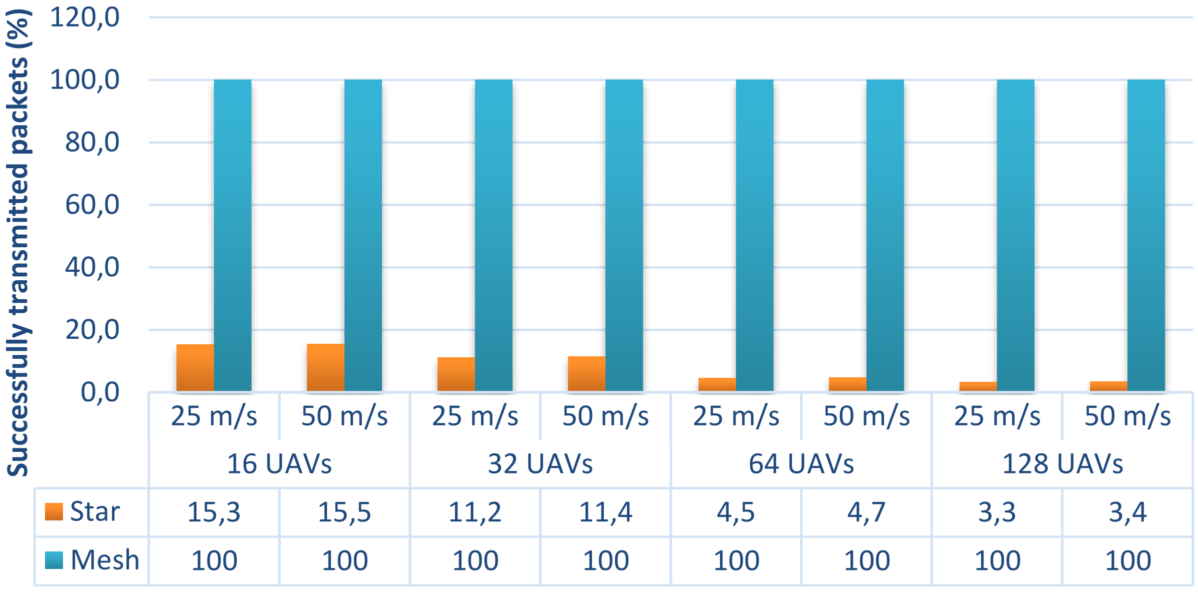
A FANET simulation was set up in OMNeT++ Simulator. The parameters chosen for investigation are described as follows: i) communication protocol: (a) IEEE 802.11n was chosen as a protocol due to its high use in UAVs data exchanges; (b) IEEE 802.15.4 was also chosen for the experiments and comparison due to the low cost, low power consumption and high connectivity. ii) network topology: (a) Star was chosen once it is one of the most common topology in ad hoc networks (broadcast); (b) Mesh was chosen because of the mobility inherent in ad hoc networks, mainly in FANETs (Ad hoc On-Demand Distance Vector Routing Protocol (AODV)). iii) amount of UAVs: we exponentially increase the amount of UAVs keeping just one CAGE in each case (16, 32,64 and 128 UAVs); iv) UAV speed: two different speed were chosen (low: 25m/s and high: 50m/s).

A previous experiment was carried out to compare results with real UAVs and simulated ones (MARCONATO *et al.*, 2016). A few UAVs were available for use in such comparison to validate the similarities between real and simulated experiments. After that, four scenarios were created on ONMeT++ simulator varying the number of hosts in order to provide results on big and small networks, similarly to a publication by Singh (2015). Experiments were run with 16, 32, 64 and 128 hosts distributed in $n \times m$ matrices, respectively: 4×4, 5×5, 8×8 and 11×11. The distance among each node was fixed to 160 m vertically and horizontally.

7.3.3 Results and discussions

In all experiments, the bigger the number of hosts, the bigger the impact on network performance degradation. The only exception was observed on IEEE 802.11n with AODV routing protocol, which had a higher rate of successfully transmitted packets. Regarding simulation time, it time was set to 1000 seconds, providing a simulation time similar to the real flight times observed in small UAVs. Thus, experiments were run and the rate of successfully transmitted packets compared. Figs 51 and 52 show results for IEEE 802.11n and IEEE 802.15.4, respectively.

Figure 51 – Comparison of successfully transmitted packets by IEEE 802.11n.



	16 UAVs		32 UAVs		64 UAVs		128 UAVs	
	25 m/s	50 m/s	25 m/s	50 m/s	25 m/s	50 m/s	25 m/s	50 m/s
🟧 Star	15,3	15,5	11,2	11,4	4,5	4,7	3,3	3,4
🟦 Mesh	100	100	100	100	100	100	100	100

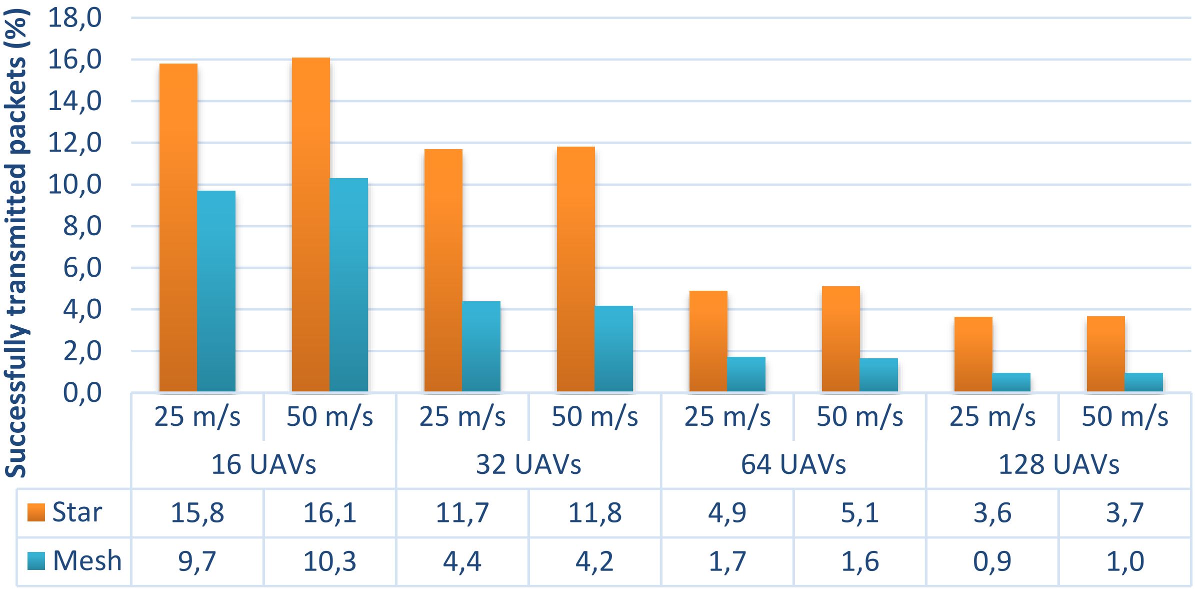
Source: Elaborated by the author.

The general behaviour of IEEE 802.15.4 simulation can be described as a reduced fraction of successfully transmitted packets. The application of AODV routing protocol and the increasing of network hosts caused even worse results.

On the other hand, the general behaviour of IEEE 802.11n changed considerably with AODV routing protocol. Without a routing protocol, the successfully transmitted packets rate was similar to the results observed for IEEE 802.15.4. However, with AODV routing protocol on IEEE 802.11n, almost 100% of packets have successfully reached their destination. The number of hosts did not affect the performance of IEEE 802.11n with AODV routing protocol.

Figure 53 presents the star configuration (without a routing protocol) and Figure 54 presents the mesh configuration (with AODV routing protocol).

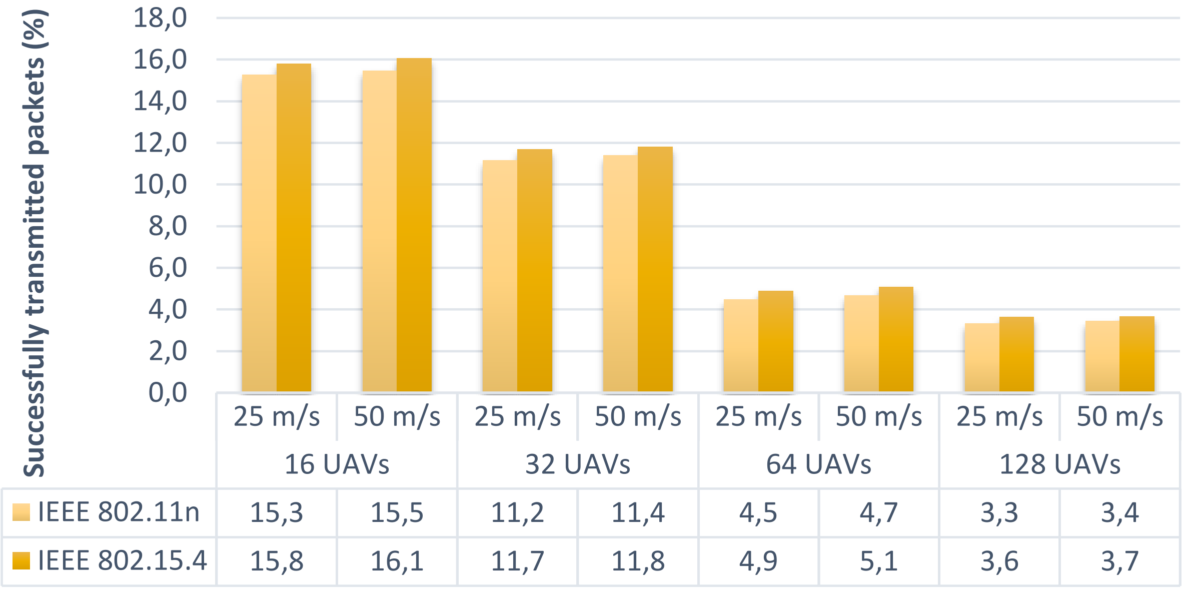
Figure 52 – Comparison of successfully transmitted packets by IEEE 802.15.4.



	16 UAVs		32 UAVs		64 UAVs		128 UAVs	
	25 m/s	50 m/s	25 m/s	50 m/s	25 m/s	50 m/s	25 m/s	50 m/s
🟧 Star	15,8	16,1	11,7	11,8	4,9	5,1	3,6	3,7
🟦 Mesh	9,7	10,3	4,4	4,2	1,7	1,6	0,9	1,0

Source: Elaborated by the author.

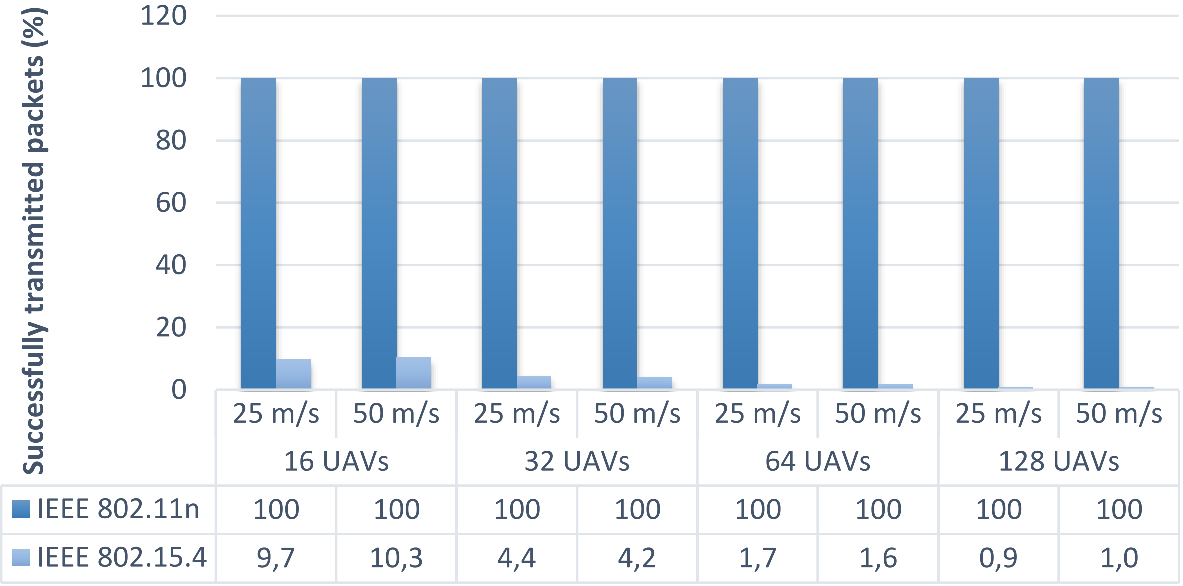
Figure 53 – Comparison of successfully transmitted packets in star configuration.



	16 UAVs		32 UAVs		64 UAVs		128 UAVs	
	25 m/s	50 m/s	25 m/s	50 m/s	25 m/s	50 m/s	25 m/s	50 m/s
🟨 IEEE 802.11n	15,3	15,5	11,2	11,4	4,5	4,7	3,3	3,4
🟨 IEEE 802.15.4	15,8	16,1	11,7	11,8	4,9	5,1	3,6	3,7

Source: Elaborated by the author.

Figure 54 – Comparison of successfully transmitted packets in mesh configuration.



	25 m/s	50 m/s	25 m/s	50 m/s	25 m/s	50 m/s	25 m/s	50 m/s
	16 UAVs		32 UAVs		64 UAVs		128 UAVs	
■ IEEE 802.11n	100	100	100	100	100	100	100	100
■ IEEE 802.15.4	9,7	10,3	4,4	4,2	1,7	1,6	0,9	1,0

Source: Elaborated by the author.

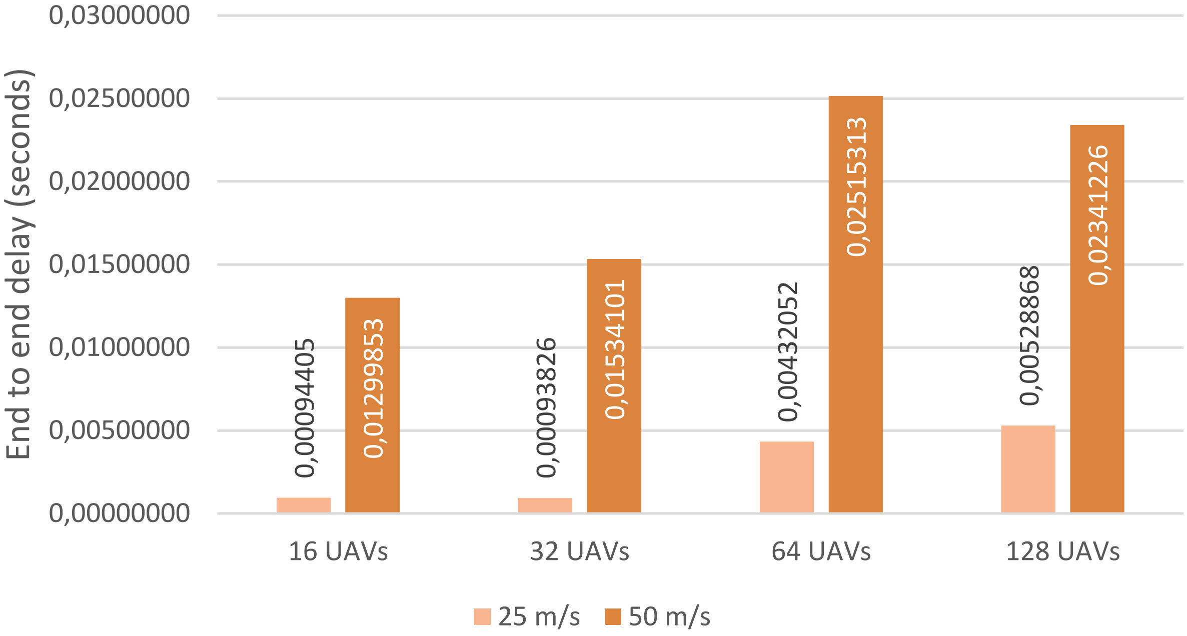
Furthermore, the end-to-end delay presented by each protocol with star and mesh configurations was also analysed. The more UAVs join the FANET, the bigger is the delay in both cases for IEEE 802.11n protocol. However, on star configuration the speed is even more important on delay increasing, as it can be seen in Figure 55. On mesh configuration there is no difference due to the speed change, as shown in Figure 56.

The end-to-end delay analysis for IEEE 802.15.4 presented a different behaviour. On star configuration, the main factor that increased the delay was the number of UAVs (Figure 57). On the contrary, on mesh configuration the delay numbers are very high and did not change much due to the number of UAVs, as shown in Figure 58.

Indeed, AODV protocol reaches high rate of successfully transmitted packets. Based on these results, one can assume that the simulation behaviour is similar as the number of UAVs is increased. Thus, simulations were carried out changing the amount of UAVs to assess the impact of topologies in safe FANETs.

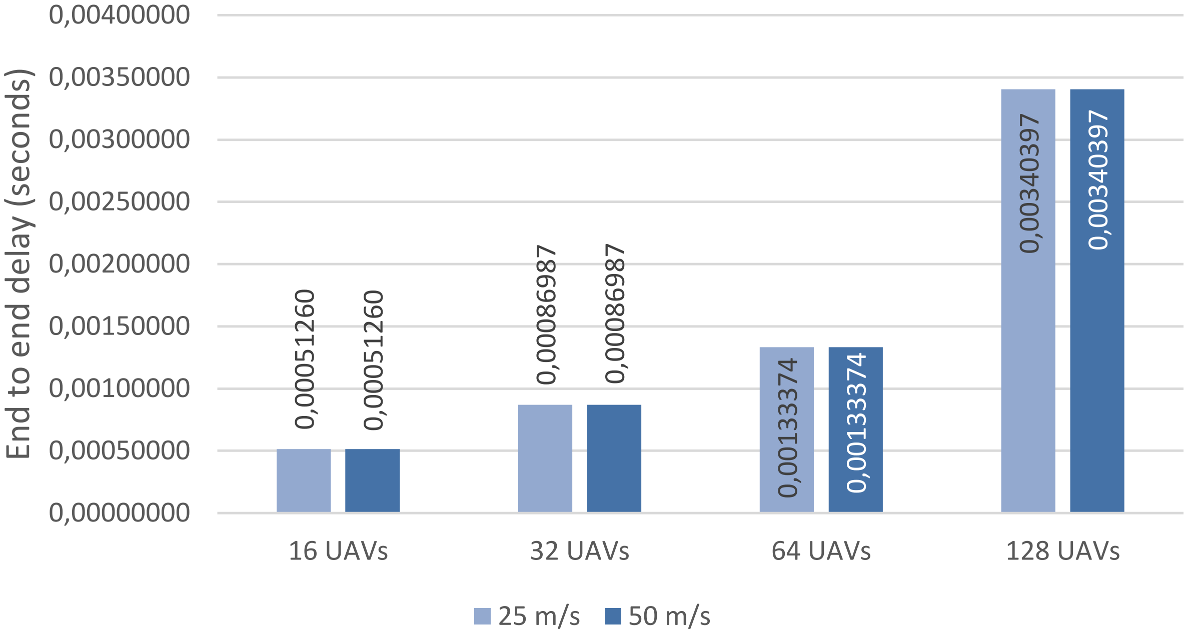
The problem with IEEE 802.15.4 protocol is seen in mesh topology due to the broadcast storm caused by the high amount of "HELLO" messages. Delays in the reconnection outcomes from the loss of "HELLO" messages when there are route losses or UAVs disconnections from the FANET. This can be solved using IEEE 802.11n protocol, which provides high delivery rates in mesh topology even with massive numbers of UAVs.

Figure 55 – Analysis of end to end delay on IEEE 802.11n with star configuration.



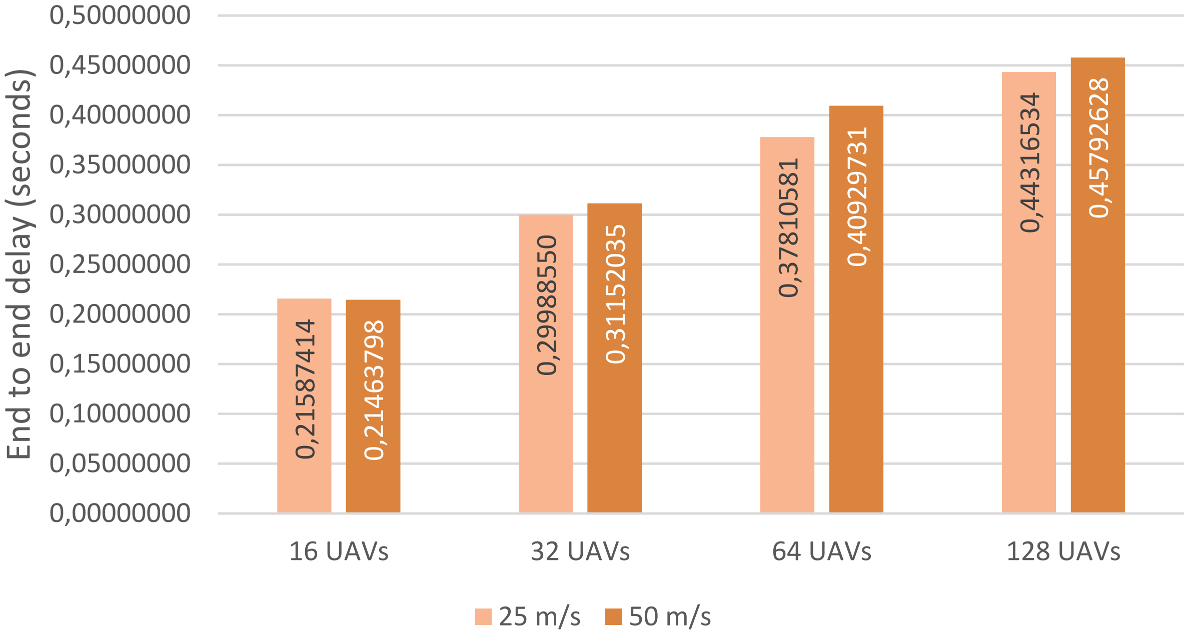
Source: Elaborated by the author.

Figure 56 – Analysis of end to end delay on IEEE 802.11n with mesh configuration.



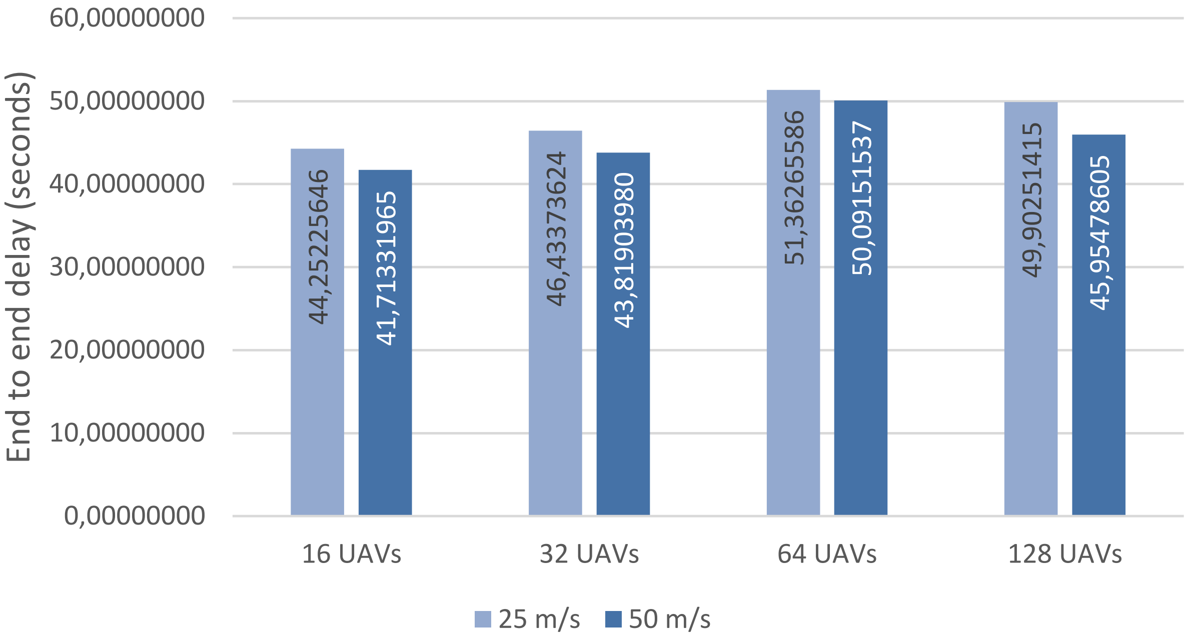
Source: Elaborated by the author.

Figure 57 – Analysis of end to end delay on IEEE 802.15.4 with Star configuration.



Source: Elaborated by the author.

Figure 58 – Analysis of end to end delay on IEEE 802.15.4 with Mesh configuration.



Source: Elaborated by the author.

Another way to solve this problem is to mitigate “HELLO” messages since there is a storm broadcast problem taking place. It can be seen as a threat, once it might cause a non-intentional DoS attack. Solving this problem, both IEEE 802.15.4 and IEEE 802.11n protocols can ensure a low end-to-end delay and high deliverable rates.

Reasons for considering IEEE 802.15.4 protocol are related to cost, power consumption and inferior complexity (FURLONG; ERICKSON, 2011). IEEE 802.11n usually requires a higher-end microcontroller or microprocessor to avoid a bottleneck of messages in its traffic, increasing overall cost. Another problem with IEEE 802.11n is the constant connection that consumes considerably more energy.

Once an IEEE 802.11n connection is a constant wireless link, more complex softwares are required to handle cases in which the connection is dropped. With IEEE 802.15.4 there is no connection that needs to be kept (the end device can simply be activated, transmit, wait for an acknowledgement and then go back to the inactive mode), allowing the device to transmit at higher power levels (longer range) and save more power by spending less time with an active radio frequency connection.

7.4 Final remarks

This chapter carried out two comparative evaluations. The first evaluation discussed was between two IPv6 network based handoff processes, with and without mobility. Some parameters, such as run time and decision making, were taken into account for performance analysis. In fact, the simulated results presented a slight advantage of handoff process without mobility. As for real-time systems, such a small difference may be very representative.

Although tests were conducted in a completely deterministic simulator and the decision making algorithm was the same in both models, results should help researchers and developers to have insights on embedded systems behaviour using IPv6 protocol facing different handoff situations.

The second evaluation was an analysis of FANETs towards the provision of safe ad hoc networks. Experiments showing the behaviour of IEEE 802.11n and IEEE 802.15.4 operating in star and mesh topologies were carried out and discussed. The simulation results showed that star network topology is affected by high UAV density and speed, which impact negatively in packet delivery rates and the end-to-end delays.

These results demonstrated that within star topology more network resources are used, due to collisions. Also due to the high speed of UAVs, the dedicated link between each UAV and CAGE fluctuates and affects data exchanges. In conclusion, FANETs using mesh topology with IEEE 802.11n are safer than using star topology with the same protocol.

Although the performance of IEEE 802.15.4 was not as good as IEEE 802.11n in mesh topology, it should still be considered with mesh topologies a low-cost, low-power, high connectivity solution.

These experiments are part of the implementation of NIMBLE platform for mobility on unmanned vehicles. Although modularised, platforms in HAMSTER architecture may still interfere each other. For instance, results presented in this chapter show that communication is also a relevant aspect for safety.

Next chapter presents the conclusions pointing out the main contributions of this thesis, limitations and difficulties, publications resulted from this thesis and future work.

CHAPTER 8

CONCLUSIONS

Advances in communications have been unarguably essential to enable modern systems and applications as we know them. Ubiquity has turned into reality, allowing specialised embedded systems to eminently grow and spread. That is notably the case of unmanned vehicles which have been creatively explored on applications that were not as efficient as they currently are, neither as innovative as recently accomplished. Therefore, towards the efficient operation of either unmanned vehicles and systems they integrate, in addition to communication improvements, it is highly desired that we carefully observe relevant, co-related necessities that may lead to the full insertion of UVs to our everyday lives. Moreover, by addressing these demands on integrated solutions, better results will likely be produced.

The initial motivation for this thesis' research investigation was mainly centred on the definition of a data communication architecture to provide safety and security to UAVs. However, throughout the development, it was identified that other co-related elements directly or indirectly affect the aforementioned requirements, thus approaches that individually aim to solve their inherent issues are not usually the most effective ones. That does not imply that complex architectures must be developed, but multi-objective and modularised ones tend to be more assertive. In actual fact, decisions were always influenced by constantly analysing the state of the art and by joining relevant discussions, reaching a more organised and focused architecture in regards of UV communications demands. Indeed, the architecture had a very important core change to encompass not only UAVs, but also similarly relevant unmanned vehicles.

Now, HAMSTER architecture contributes on the interconnection of UVs that operate in diverse environments without undertaking the necessities for safety and security. It also addresses pertinent aspects, such as the provision of context-aware formal analyses to measure nodes criticality, new ways of saving energy and individual approaches for external communications which allow the exploration of more targeted, precise solutions.

Beyond that, the case studies provided helped validate HAMSTER in both isolated and joint evaluations, proving that the architecture is applicable to the fields it proposed to improve. Conclusively, the contributions provided by HAMSTER architecture satisfy the multidisciplinary demands of modern applications. The architecture is open for further developments by potentially interested researchers and developers.

8.1 Contributions

The originality of this thesis is the definition of HAMSTER, a data communication architecture that provides an integrated reference model to address one of the main issues faced by unmanned vehicles. However, its design still took into account modularisation, which resulted on the definition of independent platforms to manage the main aspects.

Moreover, this thesis' contributions go towards the requirements of modern unmanned vehicles applications. There are natural limitations in missions performed by a single vehicle either by the restrict set of functions it can execute and the necessity of flexible approaches that imitate and go beyond human capacities. The data communication architecture specified in this thesis helps integrating heterogeneous vehicles into a unique system, keeping high levels of security and safety.

Alongside with the architecture, other relevant contributions and results were provided:

- **HAMSTER architecture**. HAMSTER is the main contribution provided by this thesis. It specifies well defined ways of achieving communication goals through a reference model to assist the development of safety, security, mobility-based, energy-efficient unmanned systems in UML, openly available for further research and development. (PIGATTO, 2013; PIGATTO *et al.*, 2014; PIGATTO *et al.*, 2016).

- **HAMSTER unit**. This unit turns modules into HAMSTER-ready elements, implementing all the platforms and features provided with the architecture on a well-defined way. The main contribution is the abstraction of physical objects when it comes to communications.

- **Security and safety Platform for HEteRogeneous systEms (SPHERE)**. The need for vehicles aligned to certification requirements has recently increased with the introduction of applications demanding their inherent flexibility. SPHERE provides specialised modules to deal with safety and security requirements both on integrated and independent approaches. This is one of the main contributions which directly meets communication-related requirements of certification processes (PIGATTO *et al.*, 2015; SILVA *et al.*, 2015).

- **Node Criticality Index (NCI)**. This contribution is the specification of a formal criticality classification for network nodes in various levels. The estimated score takes into account modules health, UV cost, manipulated and stored data, mission requirements, field of operation and the importance of fully accomplishing a mission. This set of information contributes with the provision of relevant data for the development of communication protocols, tasks delegation management units and improved system safety and information security (PIGATTO *et al.*, 2016).
- **Navigation Phases (NP)**. This approach contributes towards energy efficiency by using the knowledge on unmanned vehicles' operation phases. NP classifies known operation stages and attributes very specific behaviours that may reduce energy consumption (PIGATTO *et al.*, 2015; PIGATTO *et al.*, 2016).
- **NatIve MoBiLity platform for unmanned systEms (NIMBLE)**. External communications may include different requirements regarding mobility and operation modes. Aiming at individually addressing issues, NIMBLE manages external communications with individual modules permitting requirements-oriented developments towards ad hoc and infrastructured networks improvements (MUNHOZ *et al.*, 2016; MARCONATO *et al.*, 2016; MARCONATO *et al.*, 2017).

8.2 Limitations and difficulties

As previously mentioned, the original motivation of this project included unmanned aerial vehicles only. However, the preliminary research phase on data communication architectures for UAVs did not return enough studies to be considered to identify the state of the art. Moreover, it was identified that modern applications started to demand heterogeneous unmanned systems, which led to the necessity of a wider search. Following that, the initial architecture requirements had to be updated to meet general requirements and also domain-specific ones.

During the project development, SPHERE platform had considerably grown, introducing difficulties on management and leading to the possibility of a risky dependency on a single module. That demanded the split of SPHERE into two main modules (safety and security-specific) and the separation of NP and NCI proposals as independent platforms. Although split, safety and security segments could not be completely set apart due to eventual demands for joint approaches. The complexity with which SPHERE has to deal is a limitation on the architecture modularisation that was partially solved with CSU and SMU creation.

Validation of NCI and NP concepts were challenging. The available unmanned vehicles for the development of this research were not completely open for studies. For example, even by the fact that it is a reference for precision agriculture, the SenseFly eBee UAV does not have an open architecture. This limitation affects the completeness of a study and may lead to difficulties on validation processes. However, presented results on NCI and NP were modelled based on very well stated assumptions that present their applicability on given conditions. These conditions are seen in a majority of applications, letting us conclude that the concepts can be widely applied.

A generic reference model was provided in UML. Given the inherent complexity of UVs applications specially regarding communications, the architecture model cannot address specific low-level details of every vehicle. Notwithstanding, this thesis provided chosen case studies on specific technologies and aspects according to available resources. Considering the proportions of this investigation, a clear difficulty was on the decision of which validation aspects should be address first and at which level.

Indeed, HAMSTER was not validated in regards of 3G/4G/5G and Internet of Things connectivity. These requirements introduce an extensive new universe of issues, specially regarding security and safety. Such complexity would demand new long-term research projects to be fully addressed, not being viable within this research period.

8.3 Future works

Due to the complexity of unmanned vehicles and their recent applications, there are an abundance of future works on this field. The contributions of HAMSTER architecture emphasise its modularisation, which leads to several advantages, including the identification of either general and specific future works. The following topics summarise future efforts that should be addressed as a continuity of HAMSTER architecture:

- **General**:
  - Introduce new versions of HAMSTER architecture to deal with alternative unmanned vehicles, such as walking robots, cave/mine explorers, trains, trucks, buses, bikes, stationary balloons, hybrid vehicles and even connected humans (as seen in Body Area Networks);
  - Evaluate efficient ways of implementing HAMSTER units considering hardware and software. Thus, measure overhead and analyse the applicability of cross-layer design to reduce latency;
  - Extract relevant information from unmanned vehicles' databases through data mining techniques and apply knowledge to improve tasks performing.

- **SPHERE-related**:
  - Expand SPHERE for other embedded systems due to the current maturity of the platform and the recent heterogeneous UV applications that may include non-UV elements;
  - Constantly investigate improvements to security and safety according to evidences and official recommendations;
  - Investigate advanced security and safety techniques to allow the connection of unmanned vehicles to the Internet with high levels of confidence;
  - Development of automatic health checking mechanisms for safer unmanned vehicles provision;
  - Assess new cryptographic algorithms that meet requirements of small and complex unmanned vehicles.
- **NCI-related**:
  - Provide intelligent decision-making mechanisms that respond to NCI criticality evaluation;
  - Evaluate NCI on more complex, critical scenarios with real-world experiments;
  - Identify new variables that could be considered on calculations to make NCI more accurate;
  - Develop an attack-oriented NCI alert mechanism to trigger appropriate countermeasures;
  - Expand NCI to other systems for load balancing and resources usage improvements.
- **NP-related**:
  - Implement more accurate and automatic ways of navigation phases identification;
  - Investigate the application of adaptive control techniques to improve NP precision;
  - Apply intelligent approaches for navigation phases transitions, such as fuzzy logic, aiming at softer changes among phases;
  - Carefully analyse NP approach in fly by wireless/drive by wireless scenarios, measuring how energy-efficient the joint approaches can be and how they impact on safety, security and communications.

- **NIMBLE-related**:
  - Investigate accurate ways of connecting unmanned vehicles to the Internet, specially the Internet of Things;
  - Improve mobility considering highly connected environments;
  - Optimise connectivity in remote areas towards an efficient approach to keep connected as much as possible.

8.4 Declaration of original authorship

I confirm that this doctoral thesis has not been submitted in support of an application for another degree at this or any other teaching or research institution. It is the result of my own work and the use of all material from other sources has been properly and fully acknowledged. Research done in collaboration is also clearly indicated. Excerpts of this thesis have been either published or submitted for the appreciation of editorial boards of journals, conferences and workshops, according to the list of publications presented below. My contributions to each publication are listed as well.

8.5 Publication list

8.5.1 Published papers

- **PIGATTO, D. F.**; GONÇALVES, L; ROBERTO, G. F.; RODRIGUES FILHO, J. F.; SILVA, N. B. F.; PINTO, A. R.; BRANCO, K. R. L. J. C. **The HAMSTER Data Communication Architecture for Unmanned Aerial, Ground and Aquatic Systems**. Journal of Intelligent & Robotic Systems, v. 1, p. 1-19, 2016.

  Contribution level: High – the author was the main investigator of this paper.

- SILVA, N.B.F.; **PIGATTO, D. F.**; MARTINS, P. S.; BRANCO, K. R. L. J. C. **Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general purpose computer**. Journal of Network and Computer Applications, v. 58, p. 143, 2015.

  Contribution level: High – the author was one of the main investigator of this paper.

- MUZZI, F. A. G.; CARDOSO, P. R. M.; **PIGATTO, D. F.**; BRANCO, K. R. L. J. C. **Using Botnets to provide security for safety critical embedded systems - a case study focused on UAVs**. Journal of Physics. Conference Series (Print), v. 633, p. 53, 2015.

Contribution level: Medium – the author contributed with the case study development and analysis.

- FERNANDES, L. C.; SOUZA, J. R.; PESSIN, G.; SHINZATO, P. Y.; SALES, D.; MENDES, C.; PRADO, M.; KLASER, R.; MAGALHÃES, A. C.; HATA, A.; **PIGATTO, D. F.**; BRANCO, K. R. L. J. C.; GRASSI, V.; OSORIO, F. S.; WOLF, D. F. **CaRINA Intelligent Robotic Car: Architectural Design and Applications**. Journal of Systems Architecture, v. 60, p. 1-25, 2014.

  Contribution level: Medium – the author contributed with a case study scenario to provide security for intelligent vehicles.

- ROBERTO, G. F.; MASCHI, L. F. C.; **PIGATTO, D. F.**; BRANCO, K. R. L. J. C.; NEVES, L. A.; MONTEZ, C.; PINTO, A. S. R. **Organization model for Mobile Wireless Sensor Networks inspired in Artificial Bee Colony**. Journal of Physics. Conference Series (Print), v. 574, p. 012142, 2015.

  Contribution level: Medium – the author contributed with methodology decisions, experimental setup and results analysis.

- RIBEIRO, A. C.; PINTO, A. S. R.; ZAFALON, G. F. D.; **PIGATTO, D. F.**; BRANCO, K. R. L. J. C.; CANSIAN, A. M. **An approach to mitigate denial of service attacks in IEEE 802.11 networks**. Journal of Computer Sciences, v. 10, p. 128-137, 2014.

  Contribution level: Medium – the author contributed with experiments execution and results analysis.

- **PIGATTO, D. F.**; GONCALVES, L.; PINTO, A. S. R.; ROBERTO, G. F.; FILHO, J. F. R.; BRANCO, K. R. L. J. C. **HAMSTER – Healthy, mobility and security-based data communication architecture for Unmanned Aircraft Systems**. In: 2014 International Conference on Unmanned Aircraft Systems (ICUAS), 2014, Orlando. 2014 International Conference on Unmanned Aircraft Systems (ICUAS). v. 1. p. 52-12.

  Contribution level: High – the author was the main investigator of this research paper.

- **PIGATTO, D. F.**; SMITH, J.; BRANCO, K. R. L. J. C. **Sphere: A novel platform for increasing safety & security on Unmanned Systems** In: 2015 International Conference on Unmanned Aircraft Systems (ICUAS), 2015, Denver. 2015 International Conference on Unmanned Aircraft Systems (ICUAS). v. 1. p. 1059-1066.

  Contribution level: High – the author was the main investigator of this research paper.

- **PIGATTO, D. F.**; BRANCO, K. R. L. J. C. **Ampliando os sistemas de aeronaves não tripuladas: especificação de uma arquitetura de comunicação de dados segura e com vista à mobilidade**. In: 31º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2013, Brasília, DF. Workshop of Communication in Critical Embedded Systems (WoCCES), 2013. v. 1. p. 58-67.

  Contribution level: High – the author was the main investigator of this research paper.

- **PIGATTO, D. F.**; CASTRO, A. F. ; BRANCO, K. R. L. J. C. ; MARTIN, T. **Aplicação de Fuzzy para a redução do consumo de energia de módulos internos em veículos aéreos não tripulados**. In: 2016 8th Euro American Conference on Telematics and Information Systems (EATIS), 2016, Cartagena. 2016 8th Euro American Conference on Telematics and Information Systems (EATIS). v. 1. p. 1-7.

  Contribution level: High – the author was the main investigator of this research paper.

- MUNHOZ, L. T.; OLIVEIRA, G. C.; **PIGATTO, D. F.**; BRANCO, K. R. L. J. C. **Avaliação de desempenho de procedimentos de handoff em redes IPv6 e uma discussão sobre a viabilidade de aplicação em sistemas críticos**. In: IV Workshop of Communication in Critical Embedded Systems (WoCCES), 2016, Salvador, BA. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2016), 2016. v. 1. p. 1-12.

  Contribution level: Medium – the author participated on the research definition, experimental setup and results analysis.

- MARCONATO, E. A.; MAXA, J. A.; **PIGATTO, D. F.**; PINTO, A. S. R.; LARRIEU, N.; BRANCO, K. R. L. J. C. **IEEE 802.11n vs. IEEE 802.15.4: A Study on Communication QoS to Provide Safe FANETs**. In: 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSNW), 2016, Toulouse. 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W). v. 1. p. 184-8.

  Contribution level: Medium – the author participated on the research definition, experimental setup and results analysis.

- MARCONATO, E. A.; RODRIGUES, M.; PIRES, R. M.; **PIGATTO, D. F.**; QUERINO FILHO, L. C.; PINTO, A. R.; BRANCO, K. R. L. J. C. **AVENS - A Novel Flying Ad Hoc Network Simulator with Automatic Code Generation for Unmanned Aircraft System**. In: 50th Hawaii International Conference on System Sciences (HICSS), 2017, Waikoloa Village. Proceedings of the 50th Hawaii International Conference on System Sciences, 2017. v. 1. p. 6275-6284.

Contribution level: Medium – the author participated on the research definition, experimental setup and results analysis.

- CASTRO, A. F.; **PIGATTO, D. F.**; MAIA, R. F. **Lógica Fuzzy aplicada em sistemas de transporte**. In: Escola Potiguar de Computação e suas Aplicações (EPOCA 2015), 2015, Caicó, RN. Escola Potiguar de Computação e suas Aplicações (EPOCA 2015), 2015. v. 1. p. 94-103.

  Contribution level: Medium – the author participated on the research definition, experimental setup and results analysis.

- SILVA, N. B. F.; **PIGATTO, D. F.**; PIRES, R. M.; MARTINS, PAULO S.; BRANCO, K. R. L. J. C. **Case Studies of Performance Evaluation of Asymmetric and Symmetric Cryptographic Algorithms for Embedded Systems**. In: XII Simp ósio Brasileiro de Automação Inteligente (SBAI), 2015, Natal, RN. XII Simp ósio Brasileiro de Automa cão Inteligente (SBAI), 2015. v. 1. p. 319-324.

  Contribution level: High – the author participated on the research definition, experimental setup and results analysis.

- RODRIGUES FILHO, J. F.; **PIGATTO, D. F.**; PINTO, A. R.; BRANCO, K. R. L. J. C. **Implementação de um protótipo de unidade central de autenticação de módulos para VANTs e estabelecimento de comunicação segura com IPSec**. In: III Workshop de Comunicação em Sistemas Embarcados Críticos (WoC-CES), 2015, Vitória, ES. XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2015), 2015. v. 1. p. 1-10.

  Contribution level: High – the author participated on the research definition, experimental setup and results analysis.

- MARCONATO, E. A.; **PIGATTO, D. F.**; BRANCO, K. R. L. J. C.; BRANCO, L. H. C. **LARISSA: Layered architecture model for interconnection of systems in UAS**. In: 2014 International Conference on Unmanned Aircraft Systems (ICUAS), 2014, Orlando. 2014 International Conference on Unmanned Aircraft Systems (ICUAS). v. 1. p. 20-12.

  Contribution level: Medium – the author in the review planning and the paper writing.

- ROBERTO, G. F.; MASCHI, L. F. C.; **PIGATTO, D. F.**; BRANCO, K. R. L. J. C.; NEVES, L. A.; MONTEZ, C.; PINTO, A. S. R. **Modelo de Organização para Redes de Sensores Sem fio Móveis inspirada em Colônia de Abelhas**. In:

IV Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC), 2014, Manaus, AM. IV Simpósio Brasileiro de Engenharia de Sistemas Computacionais, 2014. v. 1. p. 1-6.

Contribution level: Medium – the author participated on the research definition and results analysis.

8.6 Accepted for publication

- **PIGATTO, D. F.**; FONTES, J. V. C.; RODRIGUES, M.; PINTO, A. R. S.; SMITH, J. BRANCO, K. R. L. J. C. **The Internet of Flying Things**. (Book Chapter). In: Internet of Things – Applications and Implementations, CRC Press.

  Estimated date for publication: end of 2017.

  Contribution level: High – the author was the main investigator of this book chapter, which includes innovative proposals.

- **PIGATTO, D. F.**; FONTES, J. V. C.; RODRIGUES, M.; PINTO, A. R. S.; DIGUET, J.; BRANCO, K. R. L. J. C. **UAV Integration Into IoIT: Opportunities and Challenges**. In: The Thirteenth International Conference on Autonomic and Autonomous Systems (ICAS 2017).

  Estimated date for publication: middle of 2017.

  Contribution level: High – the author was the main investigator of this book chapter, which includes innovative proposals.

8.7 Submitted papers

- MUNHOZ, L. T.; **PIGATTO, D. F.**; BRANCO, K. R. L. J. C. **Performance Evaluation of Handoff in Mobile IPv6 Networks: the Case of Safety-critical Systems with NIMBLE Platform for Mobility**. In: Communications in Computer and Information Science.

  Contribution level: High – the author was one of the main investigators of this paper, which is highly related to HAMSTER's NIMBLE.

BIBLIOGRAPHY

3DR. *3DR Solo*. 2017. Available: <https://3dr.com/solo-drone/>. Citation on page 107.

ABBOTT-MCCUNE, S. S.; KOBEZAK, P. P.; TRONT, J. J.; MARCHANY, R. R.; WICKS, A. A. UGV: security analysis of subsystem control network. In: KARLSEN, R. E.; GAGE, D. W.; SHOEMAKER, C. M.; GERHART, G. R. (Ed.). **Proceedings of SPIE - The International Society for Optical Engineering**. [s.n.], 2013. v. 8741, p. 87410Z. ISBN 9780819495327. ISSN 0277786X. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84881179395{&}partnerID=40{&}md5=44ebd9855c951ede2a29c3d96676efe9>. Citations on pages 36, 45 e 51.

AKRAM, R. N.; MARKANTONAKIS, K.; KARIYAWASAM, S.; AYUB, S.; SEEAM, A.; ATKINSON, R.; HOLLOWAY, R.; KARIYAWASAM, S.; AYUB, S.; SEEAM, A.; ATKINSON, R. Challenges of security and trust in Avionics Wireless Networks. In: IEEE. **2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)**. IEEE, 2015. p. 4B1–1–4B1–12. ISBN 978-1-4799-8940-9. Available: <http://ieeexplore.ieee.org/document/7311416/>. Citation on page 90.

ALBA, E.; TALBI, E.-g.; ZOMAYA, A. Y.; MAO, J.; WU, Z.; WU, X. A TDMA scheduling scheme for many-to-one communications in wireless sensor networks. **Computer Communications**, v. 30, n. 4, p. 863–872, 2007. Available: <http://www.sciencedirect.com/science/article/pii/S0140366406004002>. Citations on pages 114 e 115.

ALSHBATAT, A. I.; DONG, L. Cross layer design for mobile ad-hoc unmanned aerial vehicle communication networks. In: **2010 International Conference on Networking, Sensing and Control, ICNSC 2010**. Chicago, IL: [s.n.], 2010. p. 331–336. ISBN 9781424464531. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-77953108653{&}partnerID=40{&}md5=4dc6cd929b250b213c4be7e8b5e2fdf2>. Citation on page 40.

AMINI, R.; GILL, E.; GAYDADJIEV, G. The challenges of intra-spacecraft wireless data interfacing. **International Astronautical Federation - 58th International Astronautical Congress 2007**, v. 6, n. January, p. 4020–4025, 2007. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-54949127541{&}partnerID=tZOtx3y1>. Citation on page 125.

ARANHA, D. F.; GOUVÊA, C. P. L. **RELIC is an Efficient LIbrary for Cryptography**. 2011. Available: <http://code.google.com/p/relic-toolkit/>. Citation on page 84.

ARDUINO. **Arduino Board Leonardo**. 2016. Url{https://www.arduino.cc/en/Main/ArduinoBoard
Citation on page 123.

ASMAT, J.; RHODES, B.; UMANSKY, J.; VILLAVICENCIO, C.; YUNAS, A.; DONOHUE, G.; LACHER, A. UAS Safety: Unmanned Aerial Collision Avoidance System (UCAS).

In: **Systems and Information Engineering Design Symposium, 2006 IEEE**. [S.l.: s.n.], 2006. p. 43–49. Citation on page 30.

BAKAR, A. A.; ISMAIL, R.; AHMAD, A. R.; ABDUL, J. L.; JAIS, J. Group based access control scheme (gbac): Keeping information sharing secure in mobile ad- hoc environment. In: **2009 Fourth International Conference on Digital Information Management**. [S.l.: s.n.], 2009. p. 1–6. Citation on page 44.

BAKER, E. **Suite B Cryptography**. 2006. Available: <http://csrc.nist.gov/groups/SMA/ispab/documents/minutes/2006-03/E{_}Barker-March2006-ISPAB.pdf>. Citation on page 86.

Baraldi Sesso, D.; VISMARI, L. F.; Vieira Da Silva Neto, A.; CUGNASCA, P. S.; Camargo Jr., J. B. An Approach to Assess the Safety of ADS-B-based Unmanned Aerial Systems: Data Integrity As a Safety Issue. **J. Intell. Robotics Syst.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 84, n. 1-4, p. 621–638, dec 2016. ISSN 0921-0296. Available: <https://doi.org/10.1007/s10846-015-0321-0>. Citation on page 29.

BARR, M. **Programming embedded systems in C and C++**. [S.l.]: O'Reilly, 1999. (O'Reilly Series). ISBN 9781565923546. Citation on page 35.

BEKMEZCI, I.; SAHINGOZ, O. K.; TEMEL, S. Flying Ad-Hoc Networks (FANETs): A survey. **Ad Hoc Networks**, v. 11, n. 3, p. 1254–1270, may 2013. ISSN 15708705. Citation on page 67.

BETTSTETTER, C.; HARTMANN, C.; MOSER, C. How does randomized beamforming improve the connectivity of ad hoc networks? In: **IEEE International Conference on Communications, 2005. ICC 2005. 2005**. [S.l.: s.n.], 2005. v. 5, p. 3380–3385 Vol. 5. ISSN 1550-3607. Citation on page 40.

BILAL, K.; MALIK, S. U. R.; KHALID, O.; HAMEED, A.; ALVAREZ, E.; WIJAYSEKARA, V.; IRFAN, R.; SHRESTHA, S.; DWIVEDY, D.; ALI, M.; Shahid Khan, U.; ABBAS, A.; JALIL, N.; KHAN, S. U. A taxonomy and survey on Green Data Center Networks. **Future Generation Computer Systems**, Elsevier B.V., v. 36, p. 189–208, jul 2014. ISSN 0167739X. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0167739X13001519>. Citation on page 93.

BONAB, T. H.; MASDARI, M. Security attacks in wireless body area networks: challenges and issues. **Acad{é}mie Royale des Sciences d'Outre-Mer - Bulletin des Seances**, v. 4, n. 4, p. 100–107, 2015. Citations on pages 93 e 94.

BONADIES, S.; LEFCOURT, A.; GADSDEN, S. A. A survey of unmanned ground vehicles with applications to agricultural and environmental sensing. In: . [s.n.], 2016. v. 9866, p. 98660Q–98660Q–14. Available: <http://dx.doi.org/10.1117/12.2224248>. Citation on page 106.

BOUACHIR, O.; ABRASSART, A.; GARCIA, F.; LARRIEU, N. A mobility model for UAV ad hoc network. In: IEEE. **Unmanned Aircraft Systems (ICUAS), 2014 International Conference on**. [S.l.], 2014. p. 383–388. Citations on pages 36, 42, 49 e 67.

BOVEE, B.; NEKOUI, M.; PISHRO-NIK, H.; TESSIER, R. Evaluation of the Universal Geocast Scheme for VANETs. In: **Vehicular Technology Conference (VTC Fall), 2011 IEEE**. [S.l.: s.n.], 2011. p. 1–5. ISSN 1090-3038. Citation on page 37.

BRANCO, K. R. L. J. C.; PELIZZONI, J. M.; NERIS, L. O.; TRINDADE, O.; OSóRIO, F. S.; WOLF, D. F. Tiriba - a new approach of uav based on model driven development and multiprocessors. In: **2011 IEEE International Conference on Robotics and Automation**. [S.l.: s.n.], 2011. p. 1–4. ISSN 1050-4729. Citation on page 114.

CEMIN, D.; GOTZ, M.; PEREIRA, C. E. Reconfigurable Agents for Heterogeneous Wireless Sensor Networks. In: **Computing System Engineering (SBESC), 2012 Brazilian Symposium on**. [S.l.: s.n.], 2012. p. 1–6. ISSN 2324-7886. Citation on page 43.

CHIEN, H.-Y.; LIN, R.-Y. Identity-based key agreement protocol for mobile ad-hoc networks using bilinear pairing. In: **Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on**. [S.l.: s.n.], 2006. v. 1, p. 8 pp.–. Citation on page 44.

CHUANG, M.-C.; LEE, J.-F. FH-PMIPv6: A fast handoff scheme in Proxy Mobile IPv6 networks. In: IEEE. **2011 International Conference on Consumer Electronics, Communications and Networks (CECNet)**. [S.l.]: IEEE, 2011. p. 1297–1300. ISBN 978-1-61284-458-9. Citation on page 132.

CHUNG, H.; OH, S.; SHIM, D. H.; SASTRY, S. S. Toward Robotic Sensor Webs: Algorithms, Systems, and Experiments. **Proceedings of the IEEE**, IEEE, v. 99, n. 9, p. 1562–1586, sep 2011. ISSN 0018-9219. Citation on page 36.

______. Toward robotic sensor webs: Algorithms, systems, and experiments. **Proceedings of the IEEE**, IEEE, v. 99, n. 9, p. 1562–1586, 2011. Citation on page 67.

CLAPPER, J.; YOUNG, J.; CARTWRIGHT, J.; GRIMES, J. Unmanned systems roadmap 2007-2032. **Office of the Secretary of Defense**, p. 188, 2007. Citation on page 68.

COLOMINA, I.; MOLINA, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. **{ISPRS} Journal of Photogrammetry and Remote Sensing**, v. 92, p. 79 – 97, 2014. ISSN 0924-2716. Available: <http://www.sciencedirect.com/science/article/pii/S0924271614000501>. Citation on page 35.

COVENEY, S.; ROBERTS, K. Lightweight uav digital elevation models and orthoimagery for environmental applications: data accuracy evaluation and potential for river flood risk modelling. **International Journal of Remote Sensing**, v. 0, n. 0, p. 1–22, 2017. Available: <http://dx.doi.org/10.1080/01431161.2017.1292074>. Citation on page 106.

DANG, D.-k.; MIFDAOUI, A.; GAYRAUD, T.; Dinh-Khanh Dang; MIFDAOUI, A.; GAYRAUD, T. Fly-By-Wireless for next generation aircraft: Challenges and potential solutions. In: **2012 IFIP Wireless Days**. IEEE, 2012. p. 1–8. ISBN 978-1-4673-4404-3. ISSN 2156-9711. Available: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6402820http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6402820http://ieeexplore.ieee.org/document/6402820/>. Citations on pages 38, 39, 92 e 114.

DAS, K. **Mobile IPv6: What is Mobile IPv6?** 2017. Available: <http://ipv6.com/articles/mobile/Mobile-IPv6.htm>. Citation on page 135.

DAWSON, J. F.; HOPE, D. C.; PANITZ, M.; CHRISTOPOULOS, C. WIRELESS NETWORKS IN VEHICLES. In: **Electromagnetic Propagation in Structures and Buildings, 2008 IET Seminar**. [S.l.]: IEE, 2008. Citation on page 125.

DEFENSE, U. D. of. **CERT C Programming Language Secure Coding Standard**. [S.l.], 2007. Available: <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1255.pdf>. Citation on page 90.

DEMENTYEV, A.; HODGES, S.; TAYLOR, S.; SMITH, J. Power consumption analysis of Bluetooth Low Energy, ZigBee and ANT sensor nodes in a cyclic sleep scenario. In: **2013 IEEE International Wireless Symposium (IWS)**. IEEE, 2013. p. 1–4. ISBN 978-1-4673-2141-9. Available: <http://ieeexplore.ieee.org/document/6616827/>. Citation on page 125.

DENER, M. Security Analysis in Wireless Sensor Networks. **International Journal of Distributed Sensor Networks**, Hindawi Publishing Corporation, v. 2014, p. 1–9, 2014. ISSN 1550-1329. Available: <http://www.hindawi.com/journals/ijdsn/2014/303501/>. Citations on pages 92, 93, 94 e 95.

DIGI. **Wireless Mesh Networking RF Module**. 2016. Url{http://ftp1.digi.com/support/images/XST-AN019a_XBeeAntennas.pdf}. Citation on page 125.

DOD, U. S. Unmanned systems integrated roadmap: 2013-2038. **Washington, DC, USA**, 2013. Citation on page 29.

DURHAM, C. M.; ANDEL, T. R.; HOPKINSON, K. M.; KURKOWSKI, S. H. Evaluation of an OPNET model for unmanned aerial vehicle (UAV) networks. In: **Proceedings of the 2009 Spring Simulation Multiconference**. San Diego, CA, USA: Society for Computer Simulation International, 2009. (SpringSim '09), p. 66:1—-66:8. Available: <http://dl.acm.org/citation.cfm?id=1639809.1639878>. Citation on page 43.

EISSA, T.; RAZAL, S. A.; ASRINGADI, M. D. Enhancing manet security using secret public keys. In: **2009 International Conference on Future Networks**. [S.l.: s.n.], 2009. p. 130–134. Citation on page 82.

Elaine Shi; PERRIG, A. Designing Secure Sensor Networks. **IEEE Wireless Communications**, IEEE, v. 11, n. 6, p. 38–43, dec 2004. ISSN 1536-1284. Available: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1368895http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1368895>. Citation on page 92.

FAUGHNAN, M. S.; HOURICAN, B. J.; MACDONALD, G. C.; SRIVASTAVA, M.; WRIGHT, J. P. A.; HAIMES, Y. Y.; ANDRIJCIC, E.; GUO, Z.; WHITE, J. C. Risk analysis of unmanned aerial vehicle hijacking and methods of its detection. In: **2013 IEEE Systems and Information Engineering Design Symposium**. [S.l.: s.n.], 2013. p. 145–150. Citations on pages 44 e 82.

FERNANDES, L. L. C.; SOUZA, J. J. R.; PESSIN, G.; SHINZATO, P. P. Y.; SALES, D.; MENDES, C.; PRADO, M.; KLASER, R.; MAGALHÃES, A. A. C.; HATA, A.; PIGATTO, D.; Castelo Branco, K.; GRASSI, V.; OSORIO, F. F. S.; WOLF, D. F. D. CaRINA Intelligent Robotic Car: Architectural design and applications. **Journal of Systems Architecture**, v. 60, n. 4, p. 372–392, apr 2014. ISSN 13837621. Available:

<http://www.sciencedirect.com/science/article/pii/S1383762113002841>. Citations on pages 36 e 106.

FREW, E. E. W. E. E. W.; BROWN, T. T. X. Networking Issues for Small Unmanned Aircraft Systems. **Journal of Intelligent and Robotic Systems**, v. 54, n. 1-3, p. 21–37, jul 2008. ISSN 0921-0296. Available: <http://link.springer.com/10.1007/s10846-008-9253-2http://www.scopus.com/inward/record.url?eid=2-s2.0-56649105241{&}partnerID=40{&}md5=d6b39115abd2f7439f3ddec3335782af>. Citations on pages 37 e 38.

FURLONG, K.; ERICKSON, R. **The Power of 802.15.4 and Ethernet**. [S.l.], 2011. Available: <https://www.lsr.com/white-papers/the-power-of-802-15-4-and-ethernet>. Citation on page 153.

GARCIA, R. M.; CARVALHAL, P.; FERREIRA, M. J.; SILVA, L. F.; ALMEIDA, H.; SANTOS, C.; AFONSO, J. A. A flexible framework for data exchange and presentation between wireless sensor networks and personal devices. In: **EUROCON 2007 - The International Conference on "Computer as a Tool"**. [S.l.: s.n.], 2007. p. 1101–1105. Citation on page 114.

GASHI, I.; POVYAKALO, A.; STRIGINI, L.; MATSCHNIG, M.; HINTERSTOISSER, T.; FISCHER, B. Diversity for Safety and Security in Embedded Systems. In: **IEEE International Conference on Dependable Systems and Networks**. Atlanta, GA, USA: [s.n.], 2014. Citation on page 36.

GIMENES, R. A. V.; VISMARI, L. F.; AVELINO, V. F.; CAMARGO, J. B.; ALMEIDA, J. R.; CUGNASCA, P. S. Guidelines for the Integration of Autonomous UAS into the Global ATM. **Journal of Intelligent & Robotic Systems**, Kluwer Academic Publishers, v. 74, n. 1-2, p. 465–478, sep 2013. ISSN 0921-0296. Available: <http://dl.acm.org/citation.cfm?id=2589980.2590030>. Citation on page 45.

GODDEMEIER, N.; ROHDE, S.; POJDA, J.; WIETFELD, C. Evaluation of Potential Fields mobility strategies for aerial network provisioning. In: **GLOBECOM Workshops (GC Wkshps), 2011 IEEE**. [S.l.: s.n.], 2011. p. 1291–1296. Citation on page 41.

GOMEZ, O. E. Fly-by-wireless: Benefits, risks and technical challenges. In: **CANEUS Fly by Wireless Workshop 2010**. IEEE, 2010. p. 14–15. ISBN 978-1-4244-9255-8. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5613788>. Citations on pages 39 e 113.

GREEN, S.; ÇIÇEK, Í.; KOÇ, Ç. K. Continuous-time computational aspects of cyber-physical security. In: **2016 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)**. [S.l.: s.n.], 2016. p. 59–62. Citation on page 36.

GUPTA, L.; JAIN, R.; VASZKUN, G. Survey of Important Issues in UAV Communication Networks. **IEEE Communications Surveys & Tutorials**, v. 18, n. 2, p. 1123–1152, 2016. ISSN 1553-877X. Citations on pages 42 e 132.

HANMER, R. S.; MCBRIDE, D. T.; MENDIRATTA, V. B. Comparing reliability and security: Concepts, requirements, and techniques. **Bell Labs Technical Journal**, Alcatel-Lucent, v. 12, n. 3, p. 65–78, 2007. Citation on page 147.

HATZIEFREMIDIS, A.; ZARGANIS, K. E.; LELIGOU, H. C.; PLEROS, N. Bit error rate analysis along a slanted path link between UAVs and Ground Stations. In: **Transparent Optical Networks (ICTON), 2013 15th International Conference on**. [S.l.: s.n.], 2013. p. 1–4. ISSN 2161-2056. Citation on page 43.

HENNIGER, O.; APVRILLE, L.; FUCHS, A.; ROUDIER, Y.; RUDDLE, A.; WEYL, B. Security requirements for automotive on-board networks. In: **2009 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST)**. IEEE, 2009. p. 641–646. ISBN 978-1-4244-5346-7. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5399279>. Citations on pages 90, 97, 98, 99 e 100.

HINDEN, R.; DEERING, S. **RFC 4291 - IP Version 6 Addressing Architecture**. [S.l.], 2006. 25 p. Available: <https://tools.ietf.org/html/rfc4291.html>. Citation on page 134.

HONKAVAARA, E.; SAARI, H.; KAIVOSOJA, J.; PöLöNEN, I.; HAKALA, T.; LITKEY, P.; MäKYNEN, J.; PESONEN, L. Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight uav spectral camera for precision agriculture. **Remote Sensing**, v. 5, n. 10, p. 5006–5039, 2013. ISSN 2072-4292. Available: <http://www.mdpi.com/2072-4292/5/10/5006>. Citations on pages 101 e 102.

IANNICCA, D. C.; YOUNG, D. P.; THADHANI, S. K.; WINTER, G. A. Security risk assessment process for UAS in the NAS CNPC architecture. In: **Integrated Communications, Navigation and Surveillance Conference (ICNS), 2013**. [S.l.: s.n.], 2013. p. 1–9. ISSN 2155-4943. Citation on page 45.

INET. **Network MIPv6 Documentation**. 2017. Available at https://github.com/inet-framework/inet/blob/master/examples/mobileipv6/MIPv6Network.ned. Citations on pages 136 e 140.

Inside GNSS. **UAVs Vulnerable to Civil GPS Spoofing | Inside GNSS**. 2013. Available: <http://www.insidegnss.com/node/3131>. Citation on page 110.

ISO14508. **ISO 14508:2006 Road vehicles - Spark-plugs - Terminals**. [S.l.], 2006. Available: <https://www.iso.org/standard/39779.html>. Citation on page 90.

ISO26262. **ISO 26262-1:2011 - Road Vehicles - Functional safety**. [S.l.], 2011. Available: <https://www.iso.org/standard/43464.html>. Citation on page 95.

JAIN, A.; KANT, K.; TRIPATHY, M. R. Security solutions for wireless sensor networks. In: IEEE. **Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on**. [S.l.], 2012. p. 430–433. Citation on page 93.

JAIN, R. **The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling**. [S.l.]: Wiley, 1991. 685 p. (Wiley professional computing). ISBN 9780471503361. Citation on page 85.

JANUZAJ, V.; KUGELE, S.; LANGER, B.; SCHALLHART, C.; VEITH, H. New Challenges in the Development of Critical Embedded Systems—An "aeromotive" Perspective. In: MARGARIA, T.; STEFFEN, B. (Ed.). **Leveraging Applications of Formal Methods, Verification, and Validation**. Springer Berlin / Heidelberg, 2010, (Lecture Notes in Computer Science, v. 6415). p. 1–2. ISBN 978-3-642-16557-3. Available: <http://dx.doi.org/10.1007/978-3-642-16558-0_1>. Citation on page 35.

JAVAID, A. Y.; SUN, W.; DEVABHAKTUNI, V. K.; ALAM, M. Cyber security threat analysis and modeling of an unmanned aerial vehicle system. In: **2012 IEEE Conference on Technologies for Homeland Security (HST)**. [S.l.: s.n.], 2012. p. 585–590. Citations on pages 36, 44, 82 e 92.

JAWHAR, I.; MOHAMED, N.; AL-JAROODI, J.; AGRAWAL, D. P.; ZHANG, S. Communication and networking of UAV-based systems: Classification and associated architectures. **Journal of Network and Computer Applications**, n. 31, 2017. ISSN 10848045. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1084804517300814>. Citation on page 42.

JENA, D.; JENA, S. K. A novel and efficient cryptosystem for large message encryption. **Int. J. Inf. Commun. Techol.**, Inderscience Publishers, Inderscience Publishers, Geneva, SWITZERLAND, v. 3, n. 1, p. 32–39, 2011. ISSN 1466-6642. Available: <http://dx.doi.org/10.1504/IJICT.2011.039521>. Citation on page 83.

JINDAL, S.; MAINI, R. An Efficient Technique for Detection of Flooding and Jamming Attacks in Wireless Sensor Networks. **International Journal of Computer Applications**, Foundation of Computer Science (FCS), v. 98, n. 10, p. 25–33, 2014. Citation on page 92.

JO, M.; HAN, L.; TAN, N. D.; IN, H. P. A survey: energy exhausting attacks in MAC protocols in WBANs. **Telecommunication Systems**, v. 58, n. 2, p. 153–164, feb 2015. ISSN 1018-4864. Available: <http://link.springer.com/10.1007/s11235-014-9897-0>. Citation on page 93.

JOHNSON, D.; PERKINS, C.; ARKKO, J. **RFC 3775 - Mobility Support in IPv6**. [S.l.], 2004. 165 p. Available: <https://www.ietf.org/rfc/rfc3775.txt>. Citation on page 134.

KARTHIK, S. Underwater vehicle for surveillance with navigation and swarm network communication. **Indian Journal of Science and Technology**, v. 7, n. October, p. 22–31, 2014. ISSN 09745645. Citation on page 42.

KASHIKAR, M.; NIMBHORKAR, S. Designing acknowledgement based manet using public key cryptography. In: **2013 8th International Conference on Computer Science Education**. [S.l.: s.n.], 2013. p. 228–233. Citation on page 82.

KHAN, S.; PATHAN, A. S. K.; ALRAJEH, N. A. **Wireless Sensor Networks: Current Status and Future Trends**. Taylor & Francis, 2012. ISBN 9781466506060. Available: <https://books.google.co.uk/books?id=LDSlkuH86Z4C>. Citations on pages 92 e 93.

KHAN, Z. Drive-by-wireless teleoperation with network qos adaptation. v. 2, n. 2, p. 162–171, 2011. Available: <http://hal.archives-ouvertes.fr/hal-00560419/>. Citation on page 113.

KIKUCHI, H.; NAKAZATO, J. Modint: A Compact Modular Arithmetic Java Class Library for Cellular Phones, and its Application to Secure Electronic Voting. In: **Security and Protection in Information Processing Systems**. [S.l.: s.n.], 2004. p. 177–192. Citation on page 59.

KOCHER, P.; LEE, R.; MCGRAW, G.; RAGHUNATHAN, A.; RAVI, S.; KOCHER, P.; LEE, R.; MCGRAW, G.; RAGHUNATHAN, A. Security as a new dimension in embedded system design. In: **Proceedings of the 41st annual Design Automation Conference**. New York, NY, USA: ACM, 2004. (DAC '04, .), p. 753–760. ISBN 1-58113-828-8. Available: <http://portal.acm.org/citation.cfm?doid=996566.996771http://doi.acm.org/10.1145/996566.996771>. Citation on page 36.

KOODLI, R. **RFC 4068 - Fast Handovers for Mobile IPv6**. [S.l.], 2005. 42 p. Available: <https://tools.ietf.org/html/rfc4068>. Citation on page 134.

KOOPMAN, P. Embedded System Security. **Computer**, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 37, n. 7, p. 95–97, 2004. ISSN 0018-9162. Available: <http://dx.doi.org/10.1109/MC.2004.52>. Citation on page 95.

KSHETRI, N. The emerging role of big data in key development issues: Opportunities, challenges, and concerns. **Big Data & Society**, v. 1, n. 2, p. 2053951714564227, 2014. Available: <http://dx.doi.org/10.1177/2053951714564227>. Citation on page 101.

KUIPER, E.; NADJM-TEHRANI, S. Mobility Models for UAV Group Reconnaissance Applications. In: **Wireless and Mobile Communications, 2006. ICWMC '06. International Conference on**. [S.l.: s.n.], 2006. p. 33. Citation on page 39.

KUMAR, S. M. D.; KUMAR, B. P. V. An efficient multicast routing in manets: A genetic algorithm approach. In: **TENCON 2008 - 2008 IEEE Region 10 Conference**. [S.l.: s.n.], 2008. p. 1–6. ISSN 2159-3442. Citation on page 81.

KUROSE, J. F.; ROSS, K. W. **Computer networking: A top-down approach**. 5. ed. [S.l.]: Addison Wesley, 2009. 862 p. ISBN 9780273775638. Citations on pages 133 e 134.

LEE, M.-h.; LEE, Y.-j.; LIM, K.-j.; KWON, D.-s.; KIM, S.-h. Mobile WiMAX Performance Measurements and the selection of path loss model for UGV. In: **Information and Communication Technology Convergence (ICTC), 2010 International Conference on**. [S.l.: s.n.], 2010. p. 183–184. Citation on page 43.

LEIPOLD, F.; TASSETTO, D.; BOVELLI, S. Wireless in-cabin communication for aircraft infrastructure. **Telecommunication Systems**, v. 52, n. 2, p. 1211–1232, 2013. ISSN 1572-9451. Available: <http://dx.doi.org/10.1007/s11235-011-9636-8>. Citation on page 114.

LENSTRA, A. K.; VERHEUL, E. R. Selecting Cryptographic Key Sizes. **Journal of Cryptology**, v. 14, p. 255–293, 1999. Citation on page 83.

LI, J.; ZHOU, Y.; LAMONT, L.; DÉZIEL, M. A token circulation scheme for code assignment and cooperative transmission scheduling in CDMA-based UAV ad hoc networks. **Wireless Networks**, v. 19, n. 6, p. 1469–1484, jan 2013. ISSN 1022-0038. Available: <http://link.springer.com/10.1007/s11276-013-0545-5>. Citation on page 41.

LIAW, H.-T. A secure electronic voting protocol for general elections. **Computers & Security**, v. 23, n. 2, p. 107–119, 2004. ISSN 0167-4048. Available: <http://www.sciencedirect.com/science/article/pii/S0167404804000276>. Citation on page 59.

LRM, U. C. Laboratorio de R. M. **Carro Robótico Inteligente para Navegação Autônoma - CaRINA**. 2017. Available: <http://lrm.icmc.usp.br/web/index.php?n=Port.ProjCarina2Info>. Citation on page 108.

LUO, C.; WARD, P.; CAMERON, S.; PARR, G.; MCCLEAN, S. Communication provision for a team of remotely searching UAVs: A mobile relay approach. In: **2012 IEEE Globecom Workshops**. [S.l.]: IEEE, 2012. p. 1544–1549. ISBN 978-1-4673-4941-3. Citations on pages 36, 41 e 49.

MAN, Q.; MA, S.; XIA, L.; WANG, Y. Research on security monitoring and health management system of medium-range uav. In: **2009 8th International Conference on Reliability, Maintainability and Safety**. [S.l.: s.n.], 2009. p. 854–857. Citations on pages 44 e 82.

MARCONATO, E.; PIGATTO, D.; BRANCO, K.; BRANCO, L. LARISSA: Layered architecture model for interconnection of systems in UAS. In: **2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014 - Conference Proceedings**. [S.l.: s.n.], 2014. ISBN 9781479923762. Citation on page 90.

MARCONATO, E. A.; RODRIGUES, M.; PIRES, R. M.; PIGATTO, D. F.; FILHO, L. C. Q.; PINTO, A. S. R.; BRANCO, K. R. L. J. C. AVENS – A Novel Flying Ad Hoc Network Simulator with Automatic Code Generation for Unmanned Aircraft System. In: **The Hawaii International Conference on System Sciences (HICSS)**. [S.l.: s.n.], 2017. Citations on pages 33 e 157.

MARCONATO, E. E. A.; MAXA, J. A.; PIGATTO, D. D. F.; PINTO, A. S. R. A.; LARRIEU, N.; BRANCO, K. R. L. J. C. K. IEEE 802.11n vs. IEEE 802.15.4: A Study on Communication QoS to Provide Safe FANETs. In: **2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)**. IEEE, 2016. p. 184–191. ISBN 978-1-5090-3688-2. Available: <http://ieeexplore.ieee.org/document/7575372/>. Citations on pages 33, 131, 147 e 157.

MARWEDEL, P. **Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems**. Springer, 2010. (Embedded Systems). ISBN 9789400702578. Available: <https://books.google.co.uk/books?id=EXboa4sXlRsC>. Citation on page 95.

MAZA, I. I.; CABALLERO, F. F.; CAPITÁN, J. J.; MARTINEZ-DE-DIOS, J.; OLLERO, A. b. A.; DIOS, J. R. Martínez-de; OLLERO, A. b. A. Experimental Results in Multi-UAV Coordination for Disaster Management and Civil Security Applications. **Journal of Intelligent & Robotic Systems**, v. 61, n. 1-4, p. 563–585, jan 2011. ISSN 0921-0296. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-79951517007{&}partnerID=40{&}md5=16bb6815a7cb9ff76f1974045691b41d>. Citations on pages 36 e 49.

MINERVINI, M.; SCHARR, H.; TSAFTARIS, S. A. Image analysis: The new bottleneck in plant phenotyping [applications corner]. **IEEE Signal Processing Magazine**, v. 32, n. 4, p. 126–131, July 2015. ISSN 1053-5888. Citation on page 101.

MISHRA, A.; SHIN, M.; ARBAUGH, W. An empirical analysis of the IEEE 802.11 MAC layer handoff process. **ACM SIGCOMM Computer Communication Review**, ACM, v. 33, n. 2, p. 93, apr 2003. ISSN 01464833. Citation on page 132.

MISRA. **MISRA-C:2004 - Guidelines for the use of the C language in critical systems**. [S.l.], 2004. Available: <http://caxapa.ru/thumbs/468328/misra-c-2004.pdf>. Citation on page 90.

MIXON, M. **Todd Humphreys' Research Team Demonstrates First Successful GPS Spoofing of UAV**. 2013. Available: <http://mail.ae.utexas.edu/news/archive/2012/todd-humphreys-research-team-demonstrates-first-successful-gps-spoofing-of-uav>. Citation on page 110.

MODARES, H.; SALLEH, R.; MORAVEJOSHARIEH, A. Overview of Security Issues in Wireless Sensor Networks. In: IEEE. **Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on**. IEEE, 2011. p. 308–311. ISBN 978-1-4577-1797-0. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6076376>. Citation on page 92.

MOHAMED, N.; AL-JAROODI, J.; JAWHAR, I.; LAZAROVA-MOLNAR, S. A Service-Oriented Middleware for Building Collaborative UAVs. **Journal of Intelligent & Robotic Systems**, v. 74, n. 1-2, p. 309–321, Oct. 2013. ISSN 0921-0296. Citation on page 38.

MOZAFFARI, M.; SAAD, W.; BENNIS, M.; DEBBAH, M. Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs. **IEEE Transactions on Wireless Communications**, v. 15, n. 6, p. 3949–3963, June 2016. ISSN 1536-1276. Citation on page 35.

MULLA, D. J. Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. **Biosystems Engineering**, v. 114, n. 4, p. 358–371, 2013. ISSN 1537-5110. Available: <http://www.sciencedirect.com/science/article/pii/S1537511012001419>. Citation on page 102.

MUNHOZ, L. T.; OLIVEIRA, G. C. de; PIGATTO, D. F.; BRANCO, K. R. L. J. C. Avaliação de desempenho de procedimentos de handoff em redes IPv6 e uma discussão sobre a viabilidade de aplicação em sistemas críticos. In: **IV Workshop on Communication in Critical Embedded Systems (WoCCES), Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2016)**. Salvador, BA: [s.n.], 2016. p. 12. Citations on pages 33, 131 e 157.

MURAD, M.; SHEIKH, A. a.; MANZOOR, M. A.; FELEMBAN, E.; QAISAR, S. A Survey on Current Underwater Acoustic Sensor Network Applications. **International Journal of Computer Theory and Engineering**, v. 7, n. 1, p. 51–56, 2014. ISSN 17938201. Available: <http://www.ijcte.org/index.php?m=content{&}c=index{&}a=show{&}catid=61{&}id=1112>. Citation on page 42.

NAIR, S.; VARA, J. L. de la; SABETZADEH, M.; BRIAND, L. An extended systematic literature review on provision of evidence for safety certification. **Information and Software Technology**, Elsevier B.V., v. 56, n. 7, p. 689–717, jul 2014. ISSN 09505849. Available: <http://dx.doi.org/10.1016/j.infsof.2014.03.001http://linkinghub.elsevier.com/retrieve/pii/S0950584914000603>. Citation on page 55.

NANKANI, A. **Horizontal Handoffs within WLANs**. Phd Thesis (PhD Thesis) — Master Thesis, Department of Microelectronics and Information Technology, Royal Institute of Technology (KTH) Sweden ftp://ftp. it. kth. se/Reports/DEGREE-PROJECT-REPORTS/28 August, 2005. Citations on pages 136 e 137.

NGUYEN, P. Developing an unmanned aerial vehicle (uav) multi-sensor payload carrier for air quality monitoring. In: . [s.n.], 2016. Available: <http://eprints.qut.edu.au/94748/>. Citation on page 106.

OMNeT++ Discrete Event Simulator. **What is OMNeT++?** 2017. Available: <https://omnetpp.org/intro>. Citations on pages 135 e 136.

OSSA-GOMEZ, C.; MOARREF, M.; RODRIGUES, L. Design, construction and fly-by-wireless control of an autonomous Quadrotor helicopter. In: **2011 4th Annual Caneus Fly by Wireless Workshop**. IEEE, 2011. v. 224, p. 1–4. ISBN 978-1-4577-0971-5. ISSN 0954-4100. Available: <http://dx.doi.org/10.1243/09544100JAERO566http://ieeexplore.ieee.org/document/5965559/>. Citation on page 125.

OSSA-GÓMEZ, C.; MOARREF, M.; RODRIGUES, L. Design, construction and fly-by-wireless control of an autonomous Quadrotor Helicopter. **4th Annual Caneus Fly-By-Wireless Workshop, FBW 11**, p. 79–82, 2011. Citation on page 125.

PADMAVATHI, D. G.; SHANMUGAPRIYA, M. D. A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks. **International Journal of Computer Science and Information Security, IJCSIS**, v. 4, n. 1 {&} 2, p. 9, 2009. Available: <http://arxiv.org/abs/0909.0576>. Citation on page 94.

PANDEY, M.; VERMA, S. Performance evaluation of AODV for different mobility conditions in WSN. In: **Multimedia, Signal Processing and Communication Technologies (IMPACT), 2011 International Conference on**. [S.l.: s.n.], 2011. p. 240–243. Citation on page 41.

PATSAKIS, C.; DELLIOS, K.; BOUROCHE, M. Towards a distributed secure in-vehicle communication architecture for modern vehicles. **Computers & Security**, v. 40, p. 60–74, feb 2014. ISSN 01674048. Available: <http://www.sciencedirect.com/science/article/pii/S016740481300165X>. Citation on page 95.

PENDLI, P. K. **Contribution of Modelling and Analysis of Wireless Communication for Safety related Systems with Bluetooth Technology**. [S.l.]: kassel university press GmbH, 2014. Citation on page 95.

PERKINS, C.; JOHNSON, D.; ARKKO, J. **RFC 6275 - Mobility Support in IPv6**. [S.l.], 2011. 169 p. Available: <https://tools.ietf.org/html/rfc6275>. Citation on page 135.

PIGATTO, D.; De Castro, A.; BRANCO, K.; MARTIN, T. Aplicação de Fuzzy para a redução do consumo de energia de módulos internos em veículos aéreos não tripulados. In: **2016 8th Euro American Conference on Telematics and Information Systems, EATIS 2016**. [S.l.: s.n.], 2016. ISBN 9781509024360. Citations on pages 33, 60, 113 e 157.

PIGATTO, D. F. Ampliando os sistemas de aeronaves não tripuladas: especificação de uma arquitetura de comunicação de dados segura e com vista à mobilidade. In: **I Workshop of Communication in Critical Embedded Systems**. [S.l.: s.n.], 2013. Citations on pages 32 e 156.

PIGATTO, D. F.; GONCALVES, L.; PINTO, A. S. R.; ROBERTO, G. F.; Fernando Rodrigues Filho, J.; BRANCO, K. R. L. J. C. HAMSTER - Healthy, mobility and security-based data communication architecture for Unmanned Aircraft Systems. In: **2014 International Conference on Unmanned Aircraft Systems (ICUAS)**. IEEE, 2014. p. 52–63. ISBN 978-1-4799-2376-2. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6842238>. Citations on pages 32, 42, 47, 113 e 156.

PIGATTO, D. F.; GONÇALVES, L.; ROBERTO, G. F.; Rodrigues Filho, J. F.; Floro da Silva, N. B.; PINTO, A. R.; Lucas Jaquie Castelo Branco, K. R. The HAMSTER Data Communication Architecture for Unmanned Aerial, Ground and Aquatic Systems. **Journal of Intelligent & Robotic Systems**, p. 1–19, mar 2016. ISSN 0921-0296. Citations on pages 32, 42, 47, 75, 113, 116, 117 e 156.

PIGATTO, D. F.; SILVA, N. B. F. D.; BRANCO, K. R. L. J. C. Performance Evaluation and Comparison of Algorithms for Elliptic Curve Cryptography with El-Gamal based on MIRACL and RELIC Libraries. **Journal of Applied Computing Research**, v. 1, n. 2, p. 95–103, feb 2012. ISSN 2236-8434. Available: <http://www.unisinos.br/revistas/index.php/jacr/article/view/1789>. Citation on page 83.

PIGATTO, D. F.; SMITH, J.; LUCAS, K. R.; BRANCO, J. C. Sphere: A novel platform for increasing safety & security on Unmanned Systems. In: **2015 International Conference on Unmanned Aircraft Systems (ICUAS)**. IEEE, 2015. p. 1059–1066. ISBN 978-1-4799-6010-1. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7152397>. Citations on pages 33, 42, 47, 75, 156 e 157.

PIRES, R. d. M.; ARNOSTI, S. Z.; PINTO, A. S. R.; BRANCO, K. R. L. J. C. Experimenting Broadcast Storm Mitigation Techniques in FANETs. In: **2016 49th Hawaii International Conference on System Sciences (HICSS)**. IEEE, 2016. p. 5868–5877. ISBN 978-0-7695-5670-3. Available: <http://ieeexplore.ieee.org/document/7427915/>. Citation on page 96.

PONTI, M.; CHAVES, A. A.; JORGE, F. R.; COSTA, G. B. P.; COLTURATO, A.; BRANCO, K. R. L. J. C. Precision agriculture: Using low-cost systems to acquire low-altitude images. **IEEE Computer Graphics and Applications**, v. 36, n. 4, p. 14–20, July 2016. ISSN 0272-1716. Citation on page 102.

PUCHATY, E. M.; DELAURENTIS, D. A. A performance study of UAV-based sensor networks under cyber attack. In: **Proceedings of 2011 6th International Conference on System of Systems Engineering: SoSE in Cloud Computing, Smart Grid, and Cyber Security, SoSE 2011**. Albuquerque, NM: [s.n.], 2011. p. 214–219. ISBN 9781612847825. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-80052192882{&}partnerID=40{&}md5=e5c3968fe2f7f46503d085b7383c3862>. Citation on page 44.

QADRI, S. F.; AWAN, S. A.; AMJAD, M.; ANWAR, M.; SHEHZAD, S. Applications, Challenges, Security of Wireless Body Area Networks (WBANS) and Functionality of IEEE 802.15.4. **Science International Lahore**, v. 25, n. 4, p. 697–702, 2013. Citation on page 93.

QUARITSCH, M.; KRUGGL, K.; WISCHOUNIG-STRUCL, D.; BHATTACHARYA, S.; SHAH, M.; RINNER, B. Networked UAVs as aerial sensor network for disaster management applications. **Elektrotechnik und Informationstechnik**, v. 127, n. 3, p. 56–63, mar 2010. ISSN 0932-383X. Available: <http://link.springer.com/10.1007/s00502-010-0717-2>. Citations on pages 41 e 49.

RAJ, E. G. D. P.; SELVAKUMAR, S.; LEKHA, J. R. Node admission protocols for secure communications. In: **2011 International Conference on Emerging Trends in Electrical and Computer Technology**. [S.l.: s.n.], 2011. p. 69–73. Citations on pages 45 e 83.

RAMACHANDRAN, A.; ZHOU, Z.; HUANG, D. Computing Cryptographic Algorithms in Portable and Embedded Devices. In: **Portable Information Devices, 2007. PORTABLE07. IEEE International Conference on**. IEEE, 2007. p. 1–7. ISBN 1-4244-1039-8. Available: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4216942http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4216942>. Citation on page 84.

RAOL, J. R.; GOPAL, A. K. **Mobile Intelligent Autonomous Systems**. [S.l.]: Taylor & Francis, 2012. ISBN 9781439863008. Citation on page 36.

RAVI, S.; RAGHUNATHAN, A.; KOCHER, P.; HATTANGADY, S. Security in embedded systems: Design challenges. **ACM Trans. Embed. Comput. Syst.**, ACM, New York, NY, USA, v. 3, n. 3, p. 461–491, 2004. ISSN 1539-9087. Citation on page 36.

RAWAT, P.; SINGH, K. D.; CHAOUCHI, H.; BONNIN, J. M. Wireless sensor networks: a survey on recent developments and potential synergies. **The Journal of Supercomputing**, 2013. ISSN 0920-8542. Available: <http://link.springer.com/10.1007/s11227-013-1021-9>. Citation on page 37.

RIEKE, M.; FOERSTER, T.; BROERING, A. Unmanned Aerial Vehicles as mobile multisensor platforms. In: **Proceedings of the 14th AGILE International Conference on Geographic Information Science, Utrecht, NL, USA**. [S.l.: s.n.], 2011. p. 18–21. Citation on page 68.

RILEY, D.; EYISI, E.; BAI, J.; KOUTSOUKOS, X.; XUE, Y.; SZTIPANOVITS, J. Networked control system wind tunnel (NCSWT): an evaluation tool for networked multiagent systems. In: **Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques**. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011. (SIMUTools '11), p. 9–18. ISBN 978-1-936968-00-8. Available: <http://dl.acm.org/citation.cfm?id=2151054.2151057>. Citation on page 42.

RIVERBED. **OPNET Simulator**. 2014. Available: <http://www.riverbed.com/products/performance-management-control/opnet.html>. Citations on pages 40 e 43.

RTCA. **DO-178B Software Considerations in Airborne Systems and Equipment Certification**. 1992. Available: <http://www.rtca.org/store_product.asp?prodid=581>. Citations on pages 89 e 90.

_____. **DO-178C Software Considerations in Airborne Systems and Equipment Certification**. 2011. Available: <http://www.rtca.org/store_product.asp?prodid=803>. Citations on pages 89, 90 e 95.

RTCA Inc. **RTCA/DO-254, Design Assurance Guidance for Airborne Electronic Hardware**. [S.l.], 2000. Available: <http://www.do254.com/>. Citation on page 95.

RUBIN, I.; ZHANG, R. Placement of UAVs as Communication Relays Aiding Mobile Ad Hoc Wireless Networks. In: **Military Communications Conference, 2007. MILCOM 2007. IEEE**. [S.l.: s.n.], 2007. p. 1–7. Citation on page 43.

SAHINGOZ, O. K. Networking Models in Flying Ad-Hoc Networks (FANETs): Concepts and Challenges. **Journal of Intelligent & Robotic Systems**, v. 74, n. 1-2, p. 513–527, apr 2014. ISSN 0921-0296. Citations on pages 37, 42 e 67.

SALEEM, S.; ULLAH, S.; YOO, H. S. On the Security Issues in Wireless Body Area Networks. **International Journal of Digital Content Technology and its Applications**, v. 3, p. 1–4, 2009. ISSN 19759339. Citations on pages 92, 93 e 95.

SAMPIGETHAYA, K.; POOVENDRAN, R. Cyber-Physical Integration in Future Aviation Information Systems. **Proceedings of 31st AIAA/IEEE Digital Avionics Systems Conference**, p. 1–12, 2012. ISSN 21557195. Citation on page 114.

______. Aviation Cyber–Physical Systems: Foundations for Future Aircraft and Air Transport. **Proceedings of the IEEE**, v. 101, n. 8, p. 1834–1855, aug 2013. ISSN 0018-9219. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6480779>. Citation on page 113.

SAMPIGETHAYA, K.; POOVENDRAN, R.; SHETTY, S.; DAVIS, T.; ROYALTY, C. Future e-enabled aircraft communications and security: The next 20 years and beyond. **Proceedings of the IEEE**, IEEE, v. 99, n. 11, p. 2040–2055, nov 2011. ISSN 0018-9219. Available: <http://ieeexplore.ieee.org/document/6018242/>. Citation on page 89.

SARKAR, S. K.; BASAVARAJU, T. G.; PUTTAMADAPPA, C. **Ad hoc mobile wireless networks: principles, protocols and applications**. 1st. ed. Boca Raton: Auerbach Publications, 2008. 312 p. ISBN 13: 978-1-4200-6221-2. Citation on page 67.

SASTRY, A. S.; SULTHANA, S.; VAGDEVI, S. Security threats in wireless sensor networks in each layer. **International Journal of Advanced Networking and Applications**, Eswar Publications, v. 4, n. 4, p. 1657, 2013. Citations on pages 92, 93 e 94.

SCHOITSCH, E. Design for Safety and Security of Complex Embedded Systems: A Unified Approach. In: **Cyberspace Security and Defense: Research Issues**. Berlin/Heidelberg: Springer-Verlag, 2005. p. 161–174. Available: <http://link.springer.com/10.1007/1-4020-3381-8_9>. Citation on page 95.

SENSEFLY. **eBee senseFly**. 2017. Available: <https://www.sensefly.com/drones/ebee.html>. Citation on page 102.

SEVERINGHAUS, R.; TUMMALA, M.; MCEACHEN, J. Availability of Ad Hoc Wireless Networks of Unmanned Ground Vehicles with Group Mobility. In: **System Sciences (HICSS), 2013 46th Hawaii International Conference on**. [S.l.: s.n.], 2013. p. 5097–5105. ISSN 1530-1605. Citation on page 41.

SHIRAZIPOURAZAD, S.; GHOSH, P.; SEN, A. On connectivity of Airborne Networks in presence of region-based faults. In: **MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011**. [s.n.], 2011. p. 1997–2002. ISSN 2155-7578. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.

0-84856954578{&}partnerID=40{&}md5=e8af2b5cec50ba62ef14db8858233701>. Citation on page 40.

SILVA, N. B. F. D.; PIGATTO, D. F.; MARTINS, P. S.; BRANCO, K. R. L. J. C. Case Studies of Performance Evaluation of Cryptographic Algorithms for an Embedded System and a General Purpose Computer. **Journal of Network and Computer Applications**, Elsevier, v. 60, p. 1–14, 2015. ISSN 10848045. Available: <http://dx.doi.org/10.1016/j.jnca.2015.10.007>. Citations on pages 33, 75, 87, 88 e 156.

SINGH, A. K.; SHAFIQUE, M.; KUMAR, A.; HENKEL, J. Mapping on multi/many-core systems: Survey of current and emerging trends. In: **2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)**. [S.l.: s.n.], 2013. p. 1–10. ISSN 0738-100X. Citation on page 35.

SINGH, K. The Clone Attack in Sensor Network – Analysis and Defense. **International Journal in Applied Studies and Production Management**, v. 1, n. 3, p. 224–234, 2015. Citations on pages 93, 94 e 147.

SPARKFUN. **Xbee RF Module datasheet**. 2016. Url{https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf}. Citation on page 123.

STÄHLE, H.; HUANG, K.; KNOLL, A. Drive-by-wireless with the eCar demonstrator. In: **Proceedings of the 4th ACM SIGBED International Workshop on Design, Modeling, and Evaluation of Cyber-Physical Systems - CyPhy '14**. New York, New York, USA: ACM Press, 2014. p. 19–22. ISBN 9781450328715. Available: <http://dl.acm.org/citation.cfm?doid=2593458.2593470>. Citation on page 113.

STALLINGS, W. **Criptografia e segurança de redes: princípios e práticas**. [S.l.]: Prentice Hall Brasil, 2008. ISBN 9788576051190. Citation on page 58.

STANKUNAS, J.; RUDINSKAS, D.; LASAUSKAS, E. Experimental Research of Wireless Sensor Network Application in Aviation. **Electronics and Electrical Engineering**, v. 111, n. 5, p. 41–44, jun 2011. ISSN 2029-5731. Available: <http://www.eejournal.ktu.lt/index.php/elt/article/view/353http://www.erem.ktu.lt/index.php/elt/article/view/353>. Citation on page 125.

STARK, B.; STEVENSON, B.; CHEN, Y. ADS-B for small Unmanned Aerial Systems: Case study and regulatory practices. In: **Unmanned Aircraft Systems (ICUAS), 2013 International Conference on**. [S.l.: s.n.], 2013. p. 152–159. Citation on page 29.

STUDNIA, I.; NICOMETTE, V.; ALATA, E.; DESWARTE, Y.; KAANICHE, M.; LAAROUCHI, Y. Security of embedded automotive networks: state of the art and a research proposal. In: **SAFECOMP 2013 - Workshop CARS (2nd Workshop on Critical Automotive applications : Robustness & Safety) of the 32nd International Conference on Computer Safety, Reliability and Security**. [S.l.: s.n.], 2013. Citation on page 36.

STUDOR, G. "Fly-by-Wireless": A Revolution in Aerospace Vehicle Architecture for Instrumentation and Control. jan 2007. Available: <http://ntrs.nasa.gov/search.jsp?R=20070013704>. Citation on page 113.

SUN, F.; ZHAO, Z.; FANG, Z.; DU, L.; XU, Z.; CHEN, D. A Review of Attacks and Security Protocols for Wireless Sensor Networks. **Journal of Networks**, v. 9, n. 5, may 2014. ISSN 1796-2056. Available: <http://ojs.academypublisher.com/index.php/jnw/article/view/12511>. Citations on pages 93, 94 e 95.

SUN, Z. Z.; WANG, P. P.; VURAN, M. C. M.; AL-RODHAAN, M. M. A.; AL-DHELAAN, A. A. M.; AKYILDIZ, I. F. I. b. BorderSense: Border patrol through advanced wireless sensor networks. **Ad Hoc Networks**, v. 9, n. 3, p. 468–477, may 2011. ISSN 15708705. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-78651354776{&}partnerID=40{&}md5=d95c68ba2dbbd4b9d4805179711b1d04>. Citations on pages 36, 37 e 50.

SáMANO-ROBLES, R.; TOVAR, E.; CINTRA, J.; ROCHA, A. Wireless avionics intra-communications: Current trends and design issues. In: **2016 Eleventh International Conference on Digital Information Management (ICDIM)**. [S.l.: s.n.], 2016. p. 266–273. Citation on page 113.

TAN, S. K.; MUNRO, A. Adaptive Probabilistic Epidemic Protocol for Wireless Sensor Networks in an Urban Environment. In: **Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on**. [S.l.: s.n.], 2007. p. 1105–1110. ISSN 1095-2055. Citations on pages 38 e 41.

TANENBAUM, A. S.; WETHERALL, D. **Computer Networks**. [S.l.]: Pearson Prentice Hall, 2011. 962 p. ISBN 9780132126953. Citation on page 136.

TEMEL, S.; BEKMEZCI, I. On the performance of Flying Ad Hoc Networks (FANETs) utilizing near space high altitude platforms (HAPs). In: **Recent Advances in Space Technologies (RAST), 2013 6th International Conference on**. [S.l.: s.n.], 2013. p. 461–465. Citations on pages 37 e 40.

TOVAR, E.; PEREIRA, N.; BATE, I.; INDRUSIAK, L.; PENNA, S.; NEGRÃO, J.; VIANA, J. C.; PHILIPP, F.; MAYER, D.; HERAS, J.; PACHECO, F.; LOUREIRO, J. **Networked Embedded Systems for Active Flow Control in Aircraft**. Porto, 2012. 737–750 p. Citation on page 89.

UK Civil Aviation Authority. **Unmanned Aircraft System Operations in UK Airspace – Guidance (CAP722)**. UK, 2012. Citation on page 29.

US Army. **Unmanned Aircraft Systems Roadmap 2010–2035**. Alabama, USA, 2010. 205 p. Citation on page 29.

VASUDEVAN, A.; KUMAR, D. A.; BHUVANESWARI, N. S. Precision farming using unmanned aerial and ground vehicles. In: **2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)**. [S.l.: s.n.], 2016. p. 146–150. Citations on pages 101 e 102.

VATN, J.-O. **An experimental study of IEEE 802.11 b handover performance and its effect on voice traffic**. [S.l.], 2003. Citation on page 132.

VENKATRAMAN, K.; DANIEL, J. V.; MURUGABOOPATHI, G. Various Attacks in Wireless Sensor Network: Survey. **International Journal of Soft Computing and Engineering (IJSCE)**, v. 3, n. 1, p. 208–211, 2013. Citation on page 93.

VERMA, A.; FERNANDES, R. Persistent unmanned airborne network support for cooperative sensors. In: **Proceedings of SPIE - The International Society for Optical Engineering**. Baltimore, MD: [s.n.], 2013. v. 8756, p. 87560J. ISBN 9780819495471. ISSN 0277786X. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84881189069{&}partnerID=40{&}md5=ffb4981b4b75aa0fafdbc8017bab60ce>. Citations on pages 36, 40 e 49.

VERMA, S.; PRACHI. Communication Architecture for Underwater Wireless Sensor Network. **International Journal of Computer Network and Information Security**, v. 7, n. 6, p. 67–74, 2015. ISSN 20749090. Available: <http://www.mecs-press.org/ijcnis/ijcnis-v7-n6/v7n6-8.html>. Citation on page 42.

WALLGREN, L.; RAZA, S.; VOIGT, T. Routing Attacks and Countermeasures in the RPL-based Internet of Things. **International Journal of Distributed Sensor Networks**, Hindawi Publishing Corporation, v. 2013, p. 1–11, 2013. ISSN 1550-1329. Available: <http://www.hindawi.com/journals/ijdsn/2013/794326/>. Citations on pages 93 e 94.

WEBBER, F. N.; HIROMOTO, R. E. Assessing the Communication Issues Involved in Implemening High-Level Behaviors in Unmanned Aerial Vehicles. In: **Military Communications Conference, 2006. MILCOM 2006. IEEE**. [S.l.: s.n.], 2006. p. 1–7. Citation on page 41.

WEI, G.; LING, Y.; GUO, B.; XIAO, B.; VASILAKOS, A. V. Prediction-based data aggregation in wireless sensor networks: Combining grey model and Kalman Filter. **Computer Communications**, v. 34, n. 6, p. 793–802, 2011. Available: <http://www.sciencedirect.com/science/article/pii/S0140366410004330>. Citations on pages 114 e 115.

WONG, J. Y. **Theory of Ground Vehicles**. [S.l.]: John Wiley & Sons, 2008. ISBN 9780470170380. Citation on page 50.

World Scientific. **Unmanned Systems**. 2017. Available: <http://www.worldscientific.com/worldscinet/us>. Citation on page 36.

WYNN, R. B.; HUVENNE, V. A. I.; Le Bas, T. P.; MURTON, B. J.; CONNELLY, D. P.; BETT, B. J.; RUHL, H. A.; MORRIS, K. J.; PEAKALL, J.; PARSONS, D. R.; SUMNER, E. J.; DARBY, S. E.; DORRELL, R. M.; HUNT, J. E. Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. **Marine Geology**, The Authors, v. 352, p. 451–468, 2014. ISSN 00253227. Available: <http://dx.doi.org/10.1016/j.margeo.2014.03.012>. Citation on page 36.

XIANG, X.; LIU, C.; LAPIERRE, L.; JOUVENCEL, B. Synchronized path following control of multiple homogenous underactuated AUVs. **Journal of Systems Science and Complexity**, v. 25, n. 1, p. 71–89, feb 2012. ISSN 1009-6124. Available: <http://link.springer.com/10.1007/s11424-012-0109-2>. Citations on pages 36 e 51.

YAN, R.-j.; PANG, S.; SUN, H.-b.; PANG, Y.-j. Development and missions of unmanned surface vehicle. **Journal of Marine Science and Application**, v. 9, n. 4, p. 451–457, 2010. ISSN 1993-5048. Available: <http://dx.doi.org/10.1007/s11804-010-1033-2>. Citation on page 106.

YEARBOOK, U. A. S. Unmanned aircraft systems – The Global Perspective 2011/2012. **Blyenburg & Co, June**, p. 1709–1967, 2011. Citation on page 29.

YEDAVALLI, R. K.; BELAPURKAR, R. K. Application of wireless sensor networks to aircraft control and health management systems. **Journal of Control Theory and Applications**, v. 9, n. 1, p. 28–33, feb 2011. ISSN 1672-6340. Available: <http://link.springer.com/10.1007/s11768-011-0242-9>. Citation on page 114.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. **Computer Networks**, v. 52, n. 12, p. 2292–2330, Aug. 2008. ISSN 13891286. Citation on page 35.

Zada Khan, W.; XIANG, Y.; Y Aalsalem, M.; ARSHAD, Q.; KHAN, W. Z.; XIANG, Y.; AALSALEM, M. Y.; ARSHAD, Q.; Zada Khan, W.; XIANG, Y.; Y Aalsalem, M.; ARSHAD, Q. The Selective Forwarding Attack in Sensor Networks: Detections and Countermeasures. **International Journal of Wireless and Microwave Technologies**, v. 2, n. 2, p. 33–44, 2012. ISSN 20761449. Available: <http://www.mecs-press.org/ijwmt/ijwmt-v2-n2/v2n2-6.html>. Citation on page 93.

APPENDIX A

UML DOCUMENTATION

This appendix provides the UML documentation from the modelling project designed and exported from Astah Professional software. It complements the understanding on HAMSTER platforms interactions as presented in Chapter 3. Since it was directly exported from the modelling tool, fonts and styles are different.

Overview

This file is a specification of the model created by UML definition

Package List

Full Name	Summary
[Root]	

[Root] Package

Classifier List

Name	Type	Summary
Attitude Manager	Class	
BrigdeInternet	Class	
CAGE	Class	CAGE is an acronym that stands for Control and monitoring AGEncy. Under HAMSTER it represents a base or control station.
Connectivity_Module	Class	Identifies a communication interface possessed by a HAMSTER element. It may be inner or outer and also wired or wireless.
ControlStation	Class	
Data_Storage_Manager	Class	A Database is used by a HAMSTER entity in case it needs to store information.
Flying HAMSTER Entity	Class	
Function	Class	This class is associated to vehicle as a way of identifying their available functions.
HAMSTER_Cluster	Class	A HAMSTER Cluster of Modules is connected to more than one module on the inner part of the vehicle.
HAMSTER_Entity	Class	This HAMSTER element has interfaces both to the inner and outer parts of the vehicle, translating messages using IMC or NIMBLE resources.
HAMSTER_Module	Class	A HAMSTER Module is connected to a single module (sensor or actuator) on the inner part of vehicle.
HAMSTER_Object	Class	This is the inner HAMSTER unit associated to modules that implements the main functions of HAMSTER according to information and instructions from NCI, NP, and SPHERE.
HAMSTER_Unit	Class	The main class for HAMSTER architecture. It contains all information needed for a HAMSTER inner unit or an entity
Module	Class	A vehicle's ordinary module (sensor or actuator).
NCI	Class	This class is the Node Criticality Index associated to a HAMSTER entity. It identifies how critical an entity is considering several variables. For each type of element, the calculation will consider different information.
NIMBLE	Class	NIMBLE is the platform for mobility. Messages addressed to the outer part of the vehicle are dealt by NIMBLE's ADHOC and INFRA functions.
NP_Agent	Class	This is the agent which will specify changes to the module

		behaviour according to the current Navigation Phase.
NP_Manager	Class	It is the manager of current Navigation Phases, which is defined by analysing altitude, distance, flight time and other variables.
NP_Unit	Class	NP is the Navigation Phases class.
Phase	Class	This is a Phase on the Navigation Phases proposal. It defines how groups of modules should operate while in each specific phase of operation.
Running HAMSTER Entity	Class	
SPHERE_Central	Class	The centralised SPHERE module on a HAMSTER Entity for inner communications.
SPHERE_Local	Class	This is the SPHERE class associated with HAMSTER inner units, such as modules and clusters of modules.
SPHERE_Unit	Class	SPHERE is the platform for security and safety on HAMSTER architecture. It is logically composed by Central Security Unit (CSU) and Safety Management Unit (SMU).
Swimming HAMSTER Entity	Class	
UAV	Class	Unmanned Aerial Vehicle.
UAVCamera	Class	
UAVCluster	Class	
UGV	Class	Unmanned Ground Vehicle.
USV	Class	Unmanned Surface Vehicle.
UUV	Class	Unmanned Underwater Vehicle.
UUVCamera	Class	
UUVCluster	Class	
UWV	Class	Unmanned Water Vehicle.
list	Class	A list object.

Diagram

HAMSTER[Class Diagram]

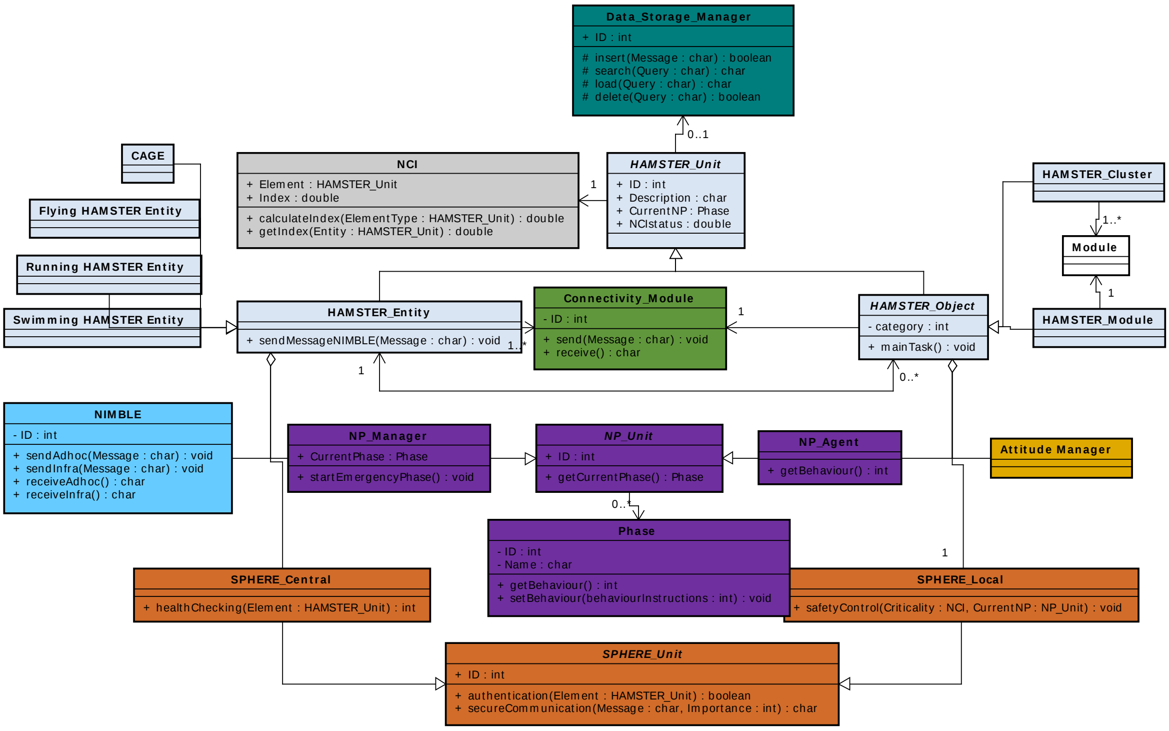

Data_Storage_Manager
+ ID : int
# insert(Message : char) : boolean
# search(Query : char) : char
# load(Query : char) : char
# delete(Query : char) : boolean
0..1
CAGE
NCI
+ Element : HAMSTER_Unit
+ Index : double
+ calculateIndex(ElementType : HAMSTER_Unit) : double
+ getIndex(Entity : HAMSTER_Unit) : double
1
HAMSTER_Unit
+ ID : int
+ Description : char
+ CurrentNP : Phase
+ NCIstatus : double
HAMSTER_Cluster
1..*
Module
1
HAMSTER_Module
Flying HAMSTER Entity
Running HAMSTER Entity
Swimming HAMSTER Entity
HAMSTER_Entity
+ sendMessageNIMBLE(Message : char) : void
1..*
1
Connectivity_Module
- ID : int
+ send(Message : char) : void
+ receive() : char
1
HAMSTER_Object
- category : int
+ mainTask() : void
0..*
NIMBLE
- ID : int
+ sendAdhoc(Message : char) : void
+ sendInfra(Message : char) : void
+ receiveAdhoc() : char
+ receiveInfra() : char
NP_Manager
+ CurrentPhase : Phase
+ startEmergencyPhase() : void
NP_Unit
+ ID : int
+ getCurrentPhase() : Phase
NP_Agent
+ getBehaviour() : int
Attitude Manager
0..*
Phase
- ID : int
- Name : char
+ getBehaviour() : int
+ setBehaviour(behaviourInstructions : int) : void
1
SPHERE_Central
+ healthChecking(Element : HAMSTER_Unit) : int
SPHERE_Local
+ safetyControl(Criticality : NCI, CurrentNP : NP_Unit) : void
SPHERE_Unit
+ ID : int
+ authentication(Element : HAMSTER_Unit) : boolean
+ secureCommunication(Message : char, Importance : int) : char

figure 1: HAMSTER[Class Diagram]

Definition

HAMSTER environment.

Class

Attitude Manager [Class]

Declaration

public class Attitude Manager

Namespace

[Root]

BrigdeInternet [Class]

Declaration

public class BrigdeInternet extends HAMSTER_Entity

Namespace

[Root]

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	BrigdeInternet	
	Generalization	HAMSTER_Entity	

CAGE [Class]

Declaration

public class CAGE
extends HAMSTER_Entity

Namespace

[Root]

Definition

CAGE is an acronym that stands for Control and monitoring AGEncy. Under HAMSTER it represents a base or control station.

Relation (From Source To Target)

Name	Type	Target	Summary
	Generalization	HAMSTER_Entity	

Connectivity_Module [Class]

Declaration

public class Connectivity_Module

Namespace

[Root]

Definition

Identifies a communication interface possessed by a HAMSTER element. It may be inner or outer and also wired or wireless.

Attribute

Name	Type	Summary
ID	private int	[Definition] Identifies a Communication Interface.

Operation

Name	Type	Summary
send (char)	public void [Parameters] in Message: char	[Definition] Translates a message according to the protocol used and sends.
receive	public char	[Definition] Receives a message and translated to HAMSTER network.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	Connectivity_Module	

ControlStation [Class]

Declaration

public class ControlStation
extends CAGE

Namespace

[Root]

Relation (From Source To Target)

Name	Type	Target	Summary
	Generalization	CAGE	

Data_Storage_Manager [Class]

Declaration

public class Data_Storage_Manager

Namespace

[Root]

Definition

A Database is used by a HAMSTER entity in case it needs to store information.

Attribute

Name	Type	Summary
ID	public int	[Definition] Identifies a Database object.

Operation

Name	Type	Summary
insert (char)	protected boolean [Parameters] in Message: char	[Definition] Adds information to the database associated to a HAMSTER entity.
search (char)	protected char [Parameters] in Query: char	[Definition] Searches information on the database associated to a HAMSTER entity.
load (char)	protected char [Parameters] in Query: char	[Definition] Loads information from the database associated to a HAMSTER entity.
delete (char)	protected boolean [Parameters] in Query: char	[Definition] Deletes information from the database associated to a HAMSTER entity.

Flying HAMSTER Entity [Class]

Declaration

public class Flying HAMSTER Entity
extends HAMSTER_Entity

Namespace

[Root]

Relation (From Source To Target)

Name	Type	Target	Summary
	Generalization	HAMSTER_Entity	

Function [Class]

Declaration

public class Function

Namespace

[Root]

Definition

This class is associated to vehicle as a way of identifying their available functions.

Attribute

Name	Type	Summary
ID	private int	[Definition] Identifies a function object.
Shortdescription	private char	[Definition] A short description of a function.
FullDescription	private char	[Definition] A full description of a function, including eventual subfunctions.

Operation

Name	Type	Summary
getShortDescription	public char	[Definition] Returns the short description.
getFullDescription	public char	[Definition] Returns the full description.
getEligibleVehicles (char)	public int [Parameters] in FunctionNeeded: char	[Definition] Returns eligible vehicles according to a function requested.

HAMSTER_Cluster [Class]

Declaration

public class HAMSTER_Cluster
extends HAMSTER_Object

Namespace

[Root]

Definition

A HAMSTER Cluster of Modules is connected to more than one module on the inner part of the vehicle.

Relation (From Source To Target)

Name	Type	Target	Summary
	Generalization	HAMSTER_Object	

HAMSTER_Entity [Class]

Declaration

**public class HAMSTER_Entity
extends HAMSTER_Unit**

Namespace

[Root]

Definition

This HAMSTER element has interfaces both to the inner and outer parts of the vehicle, translating messages using IMC or NIMBLE resources.

Operation

Name	Type	Summary
sendMessageNIMBLE(char)	public void [Parameters] in Message: char	[Definition] This operation deals with the message using NIMBLE's INFRA and ADHOC functions accordingly.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	NP_Unit	
	Association	NIMBLE	
	Association	SPHERE_Central	
	Association	NP_Manager	
	Association	NIMBLE	
	Association	NP_Manager	
	Association	Connectivity_Module	
	Association	HAMSTER_Object	
	Generalization	HAMSTER_Unit	

HAMSTER_Module [Class]

Declaration

**public class HAMSTER_Module
extends HAMSTER_Object**

Namespace

[Root]

Definition

A HAMSTER Module is connected to a single module (sensor or actuator) on the inner part of vehicle.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	Module	
	Association	Module	
	Generalization	HAMSTER_Object	

HAMSTER_Object [Class]

Declaration

public abstract class HAMSTER_Object
extends HAMSTER_Unit

Namespace

[Root]

Definition

This is the inner HAMSTER unit associated to modules that implements the main functions of HAMSTER according to information and instructions from NCI, NP, and SPHERE.

Attribute

Name	Type	Summary
category	private int	[Definition] Defines the type of HAMSTER Object according to its importance for the mission and unit operation: primary, mission-specific or any other category created by developers.

Operation

Name	Type	Summary
mainTask	public void	[Definition] Main tasks may relate to sense, actuate or both. Before operating, it checks for authorisation from NP, SPHERE and NCI.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	HAMSTER_Entity	
	Association	SPHERE_Unit	
	Association	Connectivity_Module	
	Association	NP_Agent	
	Association	SPHERE_Local	
	Association	Connectivity_Module	
	Association	Attitude Manager	
	Generalization	HAMSTER_Unit	

HAMSTER_Unit [Class]

Declaration

public abstract class HAMSTER_Unit

Namespace

[Root]

Definition

The main class for HAMSTER architecture. It contains all information needed for a HAMSTER inner unit or an entity

Attribute

Name	Type	Summary
ID	public int	[Definition] Identifies a HAMSTER object.

Description	public char	[Definition] Describes a HAMSTER element.
CurrentNP	public Phase	[Definition] Contains the current phase from Navigation Phases list to which the HAMSTER element is conditioned.
NCIstatus	public double	

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	Connectivity_Module	
	Association	Data_Storage_Manager	
	Association	NCI	
	Association	HAMSTER_Unit	
	Association	Data_Storage_Manager	

Module [Class]

Declaration

public class Module

Namespace

[Root]

Definition

A vehicle's ordinary module (sensor or actuator).

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	HAMSTER_Cluster	
	Association	HAMSTER_Cluster	

NCI [Class]

Declaration

public class NCI

Namespace

[Root]

Definition

This class is the Node Criticality Index associated to a HAMSTER entity. It identifies how critical an entity is considering several variables. For each type of element, the calculation will consider different information.

Attribute

Name	Type	Summary
Element	public HAMSTER_Unit	[Definition] Identifies the element to which the NCI object is associated.
Index	public double	[Definition] Identifies the score of a HAMSTER element.

Operation

Name	Type	Summary
calculateIndex (HAMSTER_Unit)	public double [Parameters] in ElementType: HAMSTER_Unit	[Definition] Calculates the NCI of a HAMSTER element.
getIndex (HAMSTER_Unit)	public double [Parameters] in Entity: HAMSTER_Unit	[Definition] Returns the current index of a HAMSTER element.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	HAMSTER_Unit	
	Association	HAMSTER_Entity	

NIMBLE [Class]

Declaration

public class NIMBLE

Namespace

[Root]

Definition

NIMBLE is the platform for mobility. Messages addressed to the outer part of the vehicle are dealt by NIMBLE's ADHOC and INFRA functions.

Attribute

Name	Type	Summary
ID	private int	[Definition] Identifies a NIMBLE object.

Operation

Name	Type	Summary
sendAdhoc (char)	public void [Parameters] in Message: char	[Definition] Sends messages to other vehicles on a HAMSTER network.
sendInfra (char)	public void [Parameters] in Message: char	[Definition] Receives messages from other vehicles on a HAMSTER network.
receiveAdhoc	public char	[Definition] Sends messages to infrastructure entities on a HAMSTER network.
receiveInfra	public char	[Definition] Receives messages from infrastructure entities on a HAMSTER network.

NP_Agent [Class]

Declaration

public class NP_Agent
extends NP_Unit

Namespace

[Root]

Definition

This is the agent which will specify changes to the module behaviour according to the current Navigation Phase.

Operation

Name	Type	Summary
getBehaviour	public int	[Definition] Returns the set of behaviour rules according to the current Navigation Phase.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	HAMSTER_Object	
	Generalization	NP_Unit	

NP_Manager [Class]

Declaration

public class NP_Manager
extends NP_Unit

Namespace

[Root]

Definition

It is the manager of current Navigation Phases, which is defined by analysing altitude, distance, flight time and other variables.

Attribute

Name	Type	Summary
CurrentPhase	public Phase	[Definition] Contains the current phase.

Operation

Name	Type	Summary
startEmergencyPhase	public void	[Definition] Sets a special Navigation Phase and spreads this information to all NP_Agents objects.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	HAMSTER_Entity	
	Association	NP_Agent	
	Generalization	NP_Unit	

NP_Unit [Class]

Declaration

public abstract class NP_Unit

Namespace

[Root]

Definition

NP is the Navigation Phases class.

Attribute

Name	Type	Summary
ID	public int	[Definition] Identifies a Navigation Phase object.

Operation

Name	Type	Summary
getCurrentPhase	public Phase	[Definition] Returns the current Navigation Phase.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	Phase	
	Association	Phase	

Phase [Class]

Declaration

public class Phase

Namespace

[Root]

Definition

This is a Phase on the Navigation Phases proposal. It defines how groups of modules should operate while in each specific phase of operation.

Attribute

Name	Type	Summary
ID	private int	[Definition] Identifies a Phase object.
Name	private char	[Definition] The phase name.

Operation

Name	Type	Summary
getBehaviour	public int	[Definition] Returns a set of instructions on how to operate.
setBehaviour (int)	public void [Parameters]	[Definition] Changes the set of instructions for a phase.

	in behaviourInstructions : int	

Running HAMSTER Entity [Class]

Declaration

public class Running HAMSTER Entity
extends HAMSTER_Entity

Namespace

[Root]

Relation (From Source To Target)

Name	Type	Target	Summary
	Generalization	HAMSTER_Entity	

SPHERE_Central [Class]

Declaration

public class SPHERE_Central
extends SPHERE_Unit

Namespace

[Root]

Definition

The centralised SPHERE module on a HAMSTER Entity for inner communications.

Operation

Name	Type	Summary
healthChecking (HAMSTER_Unit)	public int [Parameters] in Element: HAMSTER_Unit	[Definition] Health checking service of HAMSTER elements according to predefined beahaviour patterns, usually obtained from datasheets. This service is provided by SPHERE's SMU.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	HAMSTER_Entity	
	Generalization	SPHERE_Unit	

SPHERE_Local [Class]

Declaration

public class SPHERE_Local
extends SPHERE_Unit

Namespace

[Root]

Definition

This is the SPHERE class associated with HAMSTER inner units, such as modules and clusters of modules.

Operation

Name	Type	Summary
safetyControl (NCI, NP_Unit)	public void [Parameters] in Criticality: NCI, in CurrentNP: NP_Unit	[Definition] Service provided by SPHERE's SMU that intermitently verifies the module's NCI and current NP, taking apropriate actions if needed.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	HAMSTER_Object	
	Generalization	SPHERE_Unit	

SPHERE_Unit [Class]

Declaration

public abstract class SPHERE_Unit

Namespace

[Root]

Definition

SPHERE is the platform for security and safety on HAMSTER architecture. It is logically composed by Central Security Unit (CSU) and Safety Management Unit (SMU).

Attribute

Name	Type	Summary
ID	public int	[Definition] Identifies a SPHERE object.

Operation

Name	Type	Summary
authentication (HAMSTER_Unit)	public boolean [Parameters] in Element: HAMSTER_Unit	[Definition] Authentication service provided by SPHERE's CSU.
secureCommunication (char, int)	public char [Parameters] in Message: char, in Importance: int	[Definition] Encrypts a received message with the apropriate approach accordingly with the importance level. Service provided by CSU's Secure communication.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	HAMSTER_Entity	

Swimming HAMSTER Entity [Class]

Declaration

**public class Swimming HAMSTER Entity
extends HAMSTER_Entity**

Namespace

[Root]

Relation (From Source To Target)

Name	Type	Target	Summary
	Generalization	HAMSTER_Entity	

UAV [Class]

Declaration

public class UAV
extends Flying HAMSTER Entity

Namespace

[Root]

Definition

Unmanned Aerial Vehicle.

Attribute

Name	Type	Summary
MaximumAltitude	private int	[Definition] Maximum altitude reached by the UAV.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	HAMSTER_Cluster	
	Association	HAMSTER_Module	
	Generalization	Flying HAMSTER Entity	

UAVCamera [Class]

Declaration

public class UAVCamera

Namespace

[Root]

UAVCluster [Class]

Declaration

public class UAVCluster
extends HAMSTER_Cluster

Namespace

[Root]

Attribute

Name	Type	Summary
Modules	private Module[*]	

Relation (From Source To Target)

Name	Type	Target	Summary

	Generalization	HAMSTER_Cluster	

UGV [Class]

Declaration

public class UGV
extends Running HAMSTER Entity

Namespace

[Root]

Definition

Unmanned Ground Vehicle.

Attribute

Name	Type	Summary
CargoType	private int	[Definition] The UGV classification regarding the type of cargo it can transport.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	HAMSTER_Cluster	
	Association	HAMSTER_Module	
	Association	UGV	
	Generalization	Running HAMSTER Entity	

USV [Class]

Declaration

public class USV
extends UWV

Namespace

[Root]

Definition

Unmanned Surface Vehicle.

Attribute

Name	Type	Summary
MaximumDistance	private int	[Definition] The maximum distance the USV can reach.

Relation (From Source To Target)

Name	Type	Target	Summary
	Generalization	UWV	

UUV [Class]

Declaration

public class UUV

extends UWV, Swimming HAMSTER Entity

Namespace

[Root]

Definition

Unmanned Underwater Vehicle.

Attribute

Name	Type	Summary
MaximumProfundity	private int	[Definition] The maximum profundity the UUV can reach.

Relation (From Source To Target)

Name	Type	Target	Summary
	Association	HAMSTER_Cluster	
	Association	HAMSTER_Module	
	Generalization	UWV	
	Generalization	Swimming HAMSTER Entity	

UUVCamera [Class]

Declaration

public class UUVCamera

Namespace

[Root]

UUVCluster [Class]

Declaration

public class UUVCluster

Namespace

[Root]

Attribute

Name	Type	Summary
Modules	private Module[*]	

UWV [Class]

Declaration

public class UWV

Namespace

[Root]

Definition

Unmanned Water Vehicle.

list [Class]

Declaration

public class list

Namespace

[Root]

Definition

A list object.