Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Computer Science and Engineering

Ciclo XXIX

Settore Concorsuale di afferenza: 01/B1 - INFORMATICA
Settore Scientifico disciplinare: INF/01 - INFORMATICA

Next-generation public safety systems based on autonomous vehicles and opportunistic communications

Presentata da: **Angelo Trotta**

Coordinatore Dottorato	**Relatore**
Prof. Paolo Ciaccia	**Prof. Luciano Bononi**

Esame finale anno 2017

Abstract

An emergency scenario is characterized by the unpredictability of the environment conditions and by the scarcity of the available communication infrastructures. After a natural or human disaster, the main public and private infrastructures are partially damaged or totally destroyed. These infrastructures include roads, bridges, water supplies, electrical grids, telecommunications and so on. In these conditions, the first rescue operations executed by the public safety organizations can be very difficult, due to the unpredictability of the disaster area environment and the lack in the communications systems.

The aim of this work is to introduce next-generation public safety systems where the main focus is the use of unmanned vehicles that are able to exploit the self-organizing characteristics of such autonomous systems. With the proposed public safety systems, a team of autonomous vehicles will be able to overcome the hazardous environments of a post disaster scenario by introducing a temporary dynamic network infrastructure which enables the first responders to cooperate and to communicate with the victims involved.

Furthermore, given the pervasive penetration of smart end-user devices, the emergence of spontaneous networks could constitute promising solutions to implement emergency communication systems. With these systems the survivors will be able to self-organize in a communication network that allows them to send alerts and information messages towards the rescue teams, even in absence of communication infrastructures.

Contents

I Introduction and State of the Art 7

1 Introduction 9
1.1 Thesis aims 9

2 Public Safety Networks 13
2.1 Operational context and requirements 13
2.2 Wireless network technologies 14

3 Unmanned vehicles aided emergency systems 17
3.1 Autonomous vehicles network 17
3.1.1 Aerial networks 18
3.2 Networked autonomous vehicles 19
3.2.1 Communication networks for unmanned vehicles 20
3.2.2 Team of autonomous vehicles with controlled mobility . . 25
3.3 Unmanned vehicles support for emergency wireless networks . . . 28

II Networked robotic systems for emergency scenario 29

4 Emergency scenario peculiarities 31
4.1 Main characteristics 31
4.2 Communication Challenges 33

5 Architectural design for future systems 37
5.1 Self-organizing systems 37
5.1.1 The STEM-Network idea 37
5.1.2 The architecture of a single STEM-Node 38
5.1.3 The main architecture 44
5.1.4 The STEM-Network in a disaster recovery scenario 45
5.2 Networked behaviors 46
5.2.1 Architectural design for networked behaviors 46

5.2.2 Communication Protocols . . . 47
5.2.3 Mobility Control Systems . . . 49
5.2.4 Networked Behaviors Transversal Planes . . . 51
5.2.5 Networked behaviours in emergency scenarios . . . 54

6 Re-establishing network connectivity 57
6.1 Self-organizing autonomous vehicles . . . 57
6.1.1 Cognitive Radio technology for emergency scenarios . . . 58
6.1.2 Disaster recovery scenario modeling . . . 58
6.1.3 Devices modelling . . . 59
6.1.4 Problem Formulation . . . 60
6.2 Centralized Solution for the MCC Problem . . . 60
6.2.1 The MC-GDC Problem . . . 65
6.2.2 The MVC Problem . . . 70
6.2.3 Final step . . . 74
6.3 The Distributed Solution for the MCC Problem . . . 77
6.3.1 Creation and Maintenance of a RMN . . . 77
6.3.2 Exploration method . . . 81
6.3.3 Communication among the RUs . . . 82
6.4 Evaluation of the Distributed MCC algorithm . . . 83
6.4.1 Connectivity Index Analysis . . . 85
6.4.2 Convergence Analysis . . . 87
6.4.3 Environment Analysis . . . 89
6.5 Experimental test-bed . . . 91

7 Energy constrained systems 95
7.1 System Model . . . 96
7.2 Distributed Swarm Mobility . . . 97
7.2.1 Exploration method . . . 100
7.3 Distributed Charging Scheduling . . . 101
7.4 3D Scenario Modelling . . . 102
7.5 Charging scheduling performance . . . 103

8 TVWS frequencies for rescue missions 107
8.1 Enhancing TVWS Database with unmanned aerial vehicles . . . 108
8.1.1 Environment model . . . 109
8.1.2 Scenario modelling . . . 110
8.1.3 Problem formulation . . . 110
8.2 Scan Algorithm . . . 111
8.2.1 Field Force Based Movement . . . 111
8.2.2 Creating the scanning reports . . . 114

8.2.3 Creating the shadowing map 115
8.2.4 Combining the final reports 116
8.3 Performance Evaluation 117

III Spontaneous networks 121

9 Spontaneous networks for disaster recovery 123
9.1 Scenario Model 124
9.2 Stem Nodes (SNs) implementation 125
9.2.1 Roles Definition 125
9.2.2 Policy Manager (PM) 127
9.2.3 Cooperation Manager (CM) 127
9.2.4 Decision and Control Brain 128
9.3 The STEM-Net Creation 129
9.3.1 Distributed Gateways Election 129
9.3.2 Distributed Relays Election and Routing Process 135
9.3.3 Network Bootstrap and Maintenance 136
9.4 Role selection evaluation 137
9.5 Connection Analysis 140
9.5.1 Load Analysis 141
9.5.2 Dynamic and Parameter Characterization Analysis 142

IV Tools for design and performance analysis 147

10 Simulators for networks and controlled mobility systems 149
10.1 Actual simulation tools 150
10.1.1 Flying Drones Simulators 150
10.1.2 Network Simulators 151
10.1.3 Others coupled simulators 151
10.2 Architecture definition 152
10.2.1 FL-AIR 152
10.2.2 Network Simulator 3 153
10.2.3 Interworking 155
10.3 CUSCUS performance 158

V Conclusions 163

11 Conclusions 165

11.1 Thesis results 165
11.2 Future works 168

Part I

Introduction and State of the Art

Chapter 1

Introduction

In emergency scenarios, like the aftermath of a natural or human disaster, the public safety organizations manage groups of rescue teams that act directly on the field in order to save as many lives as possible. In these situations the first hours are critical for the outcome of the rescue operations. For this reason, a fast communication system must be deployed in the disaster area to allow a fast cooperation among the first responders. Unfortunately, the catastrophic event may totally destroy, or partially damage, the communication infrastructures that were present in the area. This event will jeopardize communications and cooperation among the rescue teams.

In order to help the public safety organizations, a fast and self-organizing system should be deployed just after the disaster. This system must be capable of creating a support network infrastructure to help the communication among the public safety teams and the survivors.

1.1 Thesis aims

The aim of this thesis is the creation of next-generation public safety communication systems that are able to cope with the hazardous conditions of an emergency situation where the communication infrastructure are not sufficient to support the rescue teams. The speed of the intervention by the public safety agencies is of utmost importance for the outcome of the rescue operations. For this reasons the ambition of this thesis is twofold: i) the analysis and the development of self-organizing autonomous systems that are capable of rapidly cover the disaster area in order to re-establish the network connectivity; ii) study and design of autonomous systems that will enable the end-user devices, owned by both the survivors and the rescue teams, to create spontaneous communication networks that will facilitate the cooperation during the rescue operations.

The development of systems able to deal with these kind of problems needs a complete architecture that will help the analysis and the planning of the future emergency systems. This thesis, in fact, describes a new architectural design for the deployment of new devices that will have self-organizing properties and will be able to self-configure in order to face the difficult on the unknown environment of a post disaster scenario.

Furthermore, one of the promising technologies for a fast deployment of emergency vehicles envisages the use of a team unmanned autonomous vehicles. This method, however, brings two main research challenges: i) it needs an adequate mobility controller that allows the proper movements and positioning of all the vehicles in the emergency scenario; ii) at the same time, the communication and the cooperation among this vehicles need an enhanced communication system due to the autonomous mobility that produces a communication network with fast topology changes. For this reason, this thesis deeply analyses the research literature in order to make a general vision of the actual works about these topics and examines the challenges that arise when working with both mobility controller systems and communication protocols. The proposed architecture will be then used in the subsequent Chapters to deal with numerous issues that arise from the creation of a self-configuring wireless networks, exhibiting its flexibility and potentiality.

After the introduction of the general architecture, this thesis deals with the problem of network repair systems. In the aftermath of a disaster, in fact, the communication infrastructures are insufficient to support the rescue operations. This study proposes a distributed method that considers the use of unmanned vehicles to re-establish the network connectivity in a disaster area in order to speed up the rescue operations.

One of the main problems of using unmanned vehicles, especially if aerial, is the battery power management. In fact, for these kind of vehicles the battery drain is very fast. In this thesis, the energy issue is analysed and a distributed mechanism is designed that will enable the vehicles to exploit the presence of recharging stations. The scheduling algorithm will deal with the choice of the right vehicle that must recharge, while avoiding the interruption of the connectivity service that the fleet of unmanned vehicles is providing.

Generally, during the rescue operations, the scarcity of communication bandwidth is an issue that can make the operations more difficult. Recent studies [1, 2] proposed the use of dynamic spectrum access for wireless communications to exploit the channels that are free, on underutilized, in order to increase the communication capabilities. The most promising technology is the deployment of *Cognitive Radio* (CR) networks [3] over *TV-White Spaces* (TVWSs) [4, 5]. These special frequency bands are now either underutilized or left free after the digital TV switch over. However, in order to use these frequencies a special geolocation spectrum database (*GLDB*) has been developed to make aware the TVWS users of the free channels ready to use in a specific geographic area [6, 7]. To exploit these recent research efforts in wireless technologies, this thesis proposes the use of *Unmanned Aerial Vehicles* (UAVs) to increase the accuracy of the geolocation spectrum database and hence to enable the use of this technology also in areas where right now are not used due to the actual mathematical prediction models that are not sufficiently accurate. In this work a scanning method for improving the geo-located database accuracy without knowing a priori the environment characteristics will be analysed.

To exploit the pervasiveness of smart devices owned by the end-users, in this thesis a system is described that enables the spontaneous creation of wireless networks among both the rescue teams and the survivors. This fosters the cooperation and enables the communication between survivors that, otherwise, would be disconnected from the main communication network. This work presents a swarm intelligence technique that is able to deal with the creation of a spontaneous multi-hop communication network among the battery-powered end-user devices based on the individual hardware characteristics, like the residual energy.

One of the main method used by the researchers to analyse and to design new technologies, is the use of simulation tools [8]. However, at present, there are no tools that enable an in-depth study of systems that run at the same time network protocols and mobility control methods with sufficient level of details. To analyse such systems, in fact, the researchers need tools that simulate the mobility controllers for the unmanned vehicles, but at the same time they need to examine the communication issues derived from the movements and the interaction of multiple vehicles. To fill this void, in the final part of this thesis a new simulation tool is described. The new simulation tool developed for this thesis work enables the researchers to simultaneously model and simulate mobility controllers and communication protocols.

Chapter 2

Public Safety Networks

Public safety sector encompasses a wide range of activities, anti-terrorism, policing, criminal justice, emergency management, homeland security [9]. The public safety agencies can generally rapidly respond to the routine events like fire in residential buildings or car accidents. However, large-scale unexpected events bring with them challenges that can strain public safety agencies, raising a lot of issues in organizing the cooperation among all the active organizations and in creating fast and reliable communication networks.

Public Safety Networks (*PSNs*) [10] are wireless communication networks established by first responders during a public safety operation. The communication capabilities provided by the PSN should manage all the difficulties originated by the challenging environment that the first responders face in the disaster area. In fact, the critical infrastructures like energy and communication stations, are often degraded or totally destroyed by the catastrophic event. The actual PSNs are unfortunately unable to satisfy all the requirements of a public safety operation in an emergency scenario [11].

2.1 Operational context and requirements

The main functions that a public safety organization should supply are [12]:

- *Law enforcement* - is the function of enforcement of the law by discovering, deterring, rehabilitating, or punishing people who violate the active regulations.

- *Emergency Medical and Health Services (EMHS)* - is the emergency service dedicated to providing out-of-hospital acute medical care and transport of patients with illnesses and injuries to a safe and controlled environment, like hospitals.

- *Border security* - is the function of ensuring the security of the national border, controlling them against intruders, narcotics smuggling and illegal importation.

- *Environment protection* - is a practice of protecting the natural environment by government organizations for the benefit of both the environment and humans. It consists in monitoring water, and land resources.

- *Fire-fighting* - is the act of extinguishing fires to protect lives and to prevent the destruction of property (houses or buildings) and of the natural environment.

- *Search and Rescue* - is the research of missing persons and the provision of first aid and transport to a safe place.

- *Emergency crisis* - is a general definition of a more complex system that integrates and manages multiple public safety functions, like search and rescue and EMHS, in order to deal with a larger crisis scenario.

The public safety systems must take into account different management strategies depending on the specific environment characteristics. The operational domains are typically defined as: border areas, urban and rural environment, ports and airports. For each domain, in fact, a specific operational task force is specialized to operate in that conditions [12].

2.2 Wireless network technologies

The ability of the first responders to communicate in real time is critical to establish command and control at the scene of an emergency, to maintain event situational awareness and to operate within a broad range of incidents. Wireless communications is the most effective way of transmitting and receiving information in any emergency situation where the affected area may not have the infrastructure in place to serve first responders, or where such infrastructure may have been destroyed because of the event itself. The main requirements for a PSN are:

- The ability of emergency responders to establish and sustain communications in support of mission operations. Equipments used at this level includes portable, mobile and base station radios.

- The ability of emergency responders to communicate among jurisdictions, disciplines, and levels of government, using a variety of frequency bands, as needed and as authorized. Interoperability is achieved by means of equipment standardization, frequency coordination, and by using systems that are capable of simultaneously cross-connecting different radio networks.

- The ability of emergency response agencies to maintain communications in the event of damage to or destruction of the primary infrastructure. This is especially serious when traditional means (i.e. wired telephone systems and the wireless cell phone networks) are rendered ineffective or taken out of service.

Given these requirements, the basic services that a public safety communication network must supply are: *voice*, *video streaming*, *data connectivity*, *asynchronous messaging* and *security services*. In the current PSNs, the primarily functionality is the voice transfer that must be guaranteed in order to enable fast requests and responses among the public safety responders [12]. Furthermore, an high level of security is also needed to protect the communications from eavesdroppers in order to protect the sensitive informations carried out by the PSN [13, 14].

During the last years, the public safety organizations have shifted from the legacy wireless technologies based on analogical transmission to digital systems. The current wireless technologies used by the public safety agencies are:

- *TETRA*. Terrestrial Trunked Radio (TETRA) [15] is a standard defined by ETSI (the European Telecommunications Standards Institute) specifically designed to be used by government agencies, emergency services, (police forces, fire departments, ambulance) for public safety networks. The main characteristic is that it allows the interoperability among different communication equipments owned by the different public safety agencies. The TETRA infrastructure is based on a cellular radio propagation technology using the frequency range 380-400 MHz in Europe.

- *TEDS*: The TETRA Enhanced Data Service (TEDS) [16] standard has been developed within the Technical Committee TETRA of the European Telecommunication Standards Institute as an upgrade to the existing narrowband ETSI TETRA system to supply professional users with high-speed IP packet data services over wireless mobile channels. The innovative key features introduced by TEDS are multicarrier-based signal modulation, powerful payload and header encoding and link adaptation which provide TETRA users with much higher data speeds. The standard allows up to *691 kb/s*, but limitations in spectrum availability typically give users a net throughput of around *100 kb/s*.

- *APCO Project 25*. The APCO project 25 (APCO P25) [17] is a suite of standards for digital radio communications mainly used in North America by the public safety organizations to enable them to communicate with other agencies and mutual aid response teams in emergencies. P25 radio systems can be used in both simplex mode without infrastructure and with infrastructures consisting in fixed and mobile equipments. The P25 Common Air Interface (*CAI*) defines the modulation techniques, frame types and physical layer representations that must be implemented by all P25-compliant radios.

- *Satellite networks*. This technology uses satellite radio transmission and hence it not rely on terrestrial communication infrastructures. This characteristic make it particularly adapt for the public safety agencies [18], especially to communicate from the emergency field towards the headquarters.

- *LTE*. The LTE technology is widely deployed as the mobile broadband standards. This qualifies it as one of the standard in the public safety communications. The pervasive presence of LTE devices among public safety agencies, other organizations and end-users facilitate the intercommunication among different entities. Moreover, the LTE system provides useful features for the public safety agencies [19, 20] like: Quality of Service (QoS), reliability, resiliency, roaming and spectrum efficiency. This technology, however, needs a pre-planning strategy for an efficient deployment of the base stations [21].
  Furthermore, the migration from the actual land mobile radio (*LMR*) dedicated systems to the LTE technology is a big issue given the high deployment cost. Moreover, the LTE technology needs some technical adjustment in order to fulfil all the public safety communication network requirements. For this reason a possible hybrid solution concerning both legacy public safety systems and LTE system is proposed [22].

Chapter 3

Unmanned vehicles aided emergency systems

3.1 Autonomous vehicles network

After a disaster, it is important for the public safety operators to be able to communicate and coordinate in order to fast response to the emergency scenario. The recent advanced in personal communication technologies like LTE or WiMAX are, unfortunately, useless for mainly two reasons: (i) most of the public safety organizations communicate between themselves using their own separate network [23] and (ii) the communication infrastructure can be partially/totally damaged or insufficient to supply the needs of the public safety operations [24].

Recently, the use of unmanned vehicles (*UV*) have gained the attention in public safety systems due to the fast deployment and the autonomous mobility that characterize these kind of vehicles [25]. The IEEE Robotics and Automation Society's *Technical Committee on Safety Security and Rescue Robotics* (TC-SSRR), in fact, created in 2012 a research a development roadmap to accelerate the adoption of ground, aerial and marine unmanned vehicles by public safety agencies [26]. In particular the unmanned aerial vehicles (*UAVs*), like multi-copters or fixed-wings planes, have gained attention in public safety communications for the high rapidly deployment and for the communication advantages due to the line-of-sight link connections among the aerial network [27]. Moreover, the use of *UVs* can improve security and emergency response in any high risk security zone [28] where unmanned systems provide versatility, efficiency, accuracy while reducing drastically the human risks by avoiding the direct participation of human operators.

3.1.1 Aerial networks

From the beginning of the years 2010, new advancements in the Unmanned Aircraft Systems [29] technology research in military and civilian areas have lead to more efficient and superior UAV classes of all shapes and sizes. In detail, a tremendous amount of development has been carried out for small-size vehicles. A robotic research approach has been applied to these Lightweight UAVs to form conglomerates of Unmanned Aerial Systems (UASs), with their own architectures and challenges. The research issues on this last topic can be approached from different standpoints but, namely, the most important ones are *Networking* and *Robotic Control* ones [30, 31, 32, 33]. From a robotic standpoint, the capability of these vehicles to be either remotely controlled or completely autonomous made them ideal for a long list of applications, like, for example, public safety, search and rescue and disaster recovery management. Many more applicative fields could arise in both near and far future as the Internet of Drones (*IoD*) paradigm is emerging [34, 35]. From a networking standpoint instead, the processing, control, storage and networking capabilities of these flying drones are on-par with VANET nodes.

Due to their high mobility and low cost, *UAVs* have found a wide range of applications [36] like, for instance, public safety, search and rescue missions and disaster recovery systems [37]. Multiple UAVs, in fact, can be utilized to create self-organizing flying swarms, specially designed for rescue operations [38, 39, 40, 41, 42]. Aerial ad-hoc networks provide the advantage to be deployable also on critical scenarios where terrestrial mobile devices might not operate, however their implementation is challenging from the point of view of communication network protocols, of mobility management and of coverage lifetime. In this case, an enabling technology for this kind of structural conglomerate, consists in a set of efficient communication methods supporting control and vice-versa. Therefore, recently, the domains of Robotic Control and Networking research have merged in the field of *Networked Robotic Systems*. Communication networks for aerial vehicles are recalled in literature as Flying Ad-Hoc Network (*FANET*) [43]. These kind of networks differs from the Mobile Ad-hoc Network (MANET) and the Vehicular Ad-hoc Network (VANET) due to the flying characteristics of the vehicles. Even though the advance in embedded and mechanics systems for creating ever more powerful UAVs, the collaboration and the coordination among multiple small UAV could enable the accomplishment of such missions that are difficult, or even impossible, to fulfil with a single multi-functional UAV. The drawback of using multiple UAVs, instead of a single UAV, is that the vehicles need to coordinate themselves through a communication network that can be difficult to maintain.

The main differences between the FANETs with respect to the other ad-hoc network architecture are:

- *High mobility degree*: the mean velocity of an aerial vehicle is generally higher than the terrestrial ones. Moreover, the fixed-wings models have constrained mobility pattern due to the mobility dynamics.

- *Controlled mobility (unmanned vehicles)*: the networks of unmanned aerial vehicles have the characteristic of autonomous mobility capabilities. This brings a lot of opportunities due to the possible topology self-configuration and self-deployment.

- *Rapid topology variation*: the high speed in aerial mobility cause a fast change in the network topology. This characteristic originates many issues, especially in maintaining stable communication multi-hop paths between UAVs.

- *Energy constraints*: the flying networks have more energy issues that the terrestrial counterparts; the battery operating time, in fact, is very low due to the energy drain caused by the propeller engines.

- *Radio propagation model*: generally the UAV networks work in line-of-sight, especially in outdoor missions. In this case the communication links are direct and without obstacles, bringing to longer communication links.

- *Six degree-of-freedom movements*: UAVs are capable to implement a six-degrees-of-freedom control: three degrees of freedom for the position control and three for the orientation one. This feature leads to non-negligible problems for the lowest communication layers of the ISO/OSI stack. In fact, the vehicles' movements have direct impact on the main transceiver antennas' properties like polarization, orientation and directionality [44].

Furthermore, for an aerial vehicle we can mainly distinguish two kinds of different communication links: air-to-air (A2A) [45] and air-to-ground (A2G) (or ground-to-air (G2A)) ones [46]. The former are the direct links between UAVs generating the multi-hop communication network, while the latter are the links between UAVs and the ground stations.

3.2 Networked autonomous vehicles

In this Section are reviewed those systems that deal with the problem of networked autonomous vehicles. For these kind of networks can be extracted two main research field: *communication networks* and *mobility control systems*. These two fields are, in fact, the two main issues regarding a network of autonomous vehicles. However, as we will see in this Section, the research works present in literature are split in two sub-categories: *mobility-aware communication protocols* and *communication-aware mobility control systems*. The former are those works whose aim is to design communication protocols that foresee/react to the vehicles movements and, consequently, to the networks topology dynamics; the latter are those control systems that actually control the motion of the vehicle while withstanding to the communication control constraints.

Even though there already exists a lot of research in connectivity support missions, there are many open research issues. In fact, the micro movements made by a vehicle influence directly all the indicators of the lower communication layers indexes of the nearly neighbourhood like Bit-Error-Rate, change of collision domains, increasing/reducing the hidden nodes problem and so on. On the other side, changing the communication parameters, like transmission power, coding schemes, frequency and timing in the channel access, link quality estimation in packet routing decisions and so on, would directly impact all the constraints in the vehicle motion allowing more, or less, possibility of movements. It is clear that a deep work in this joint research field is essential to reach optimal and realistic results.

3.2.1 Communication networks for unmanned vehicles

Different research lines has been followed in order to develop networking solutions that are able to cope with mobility issues typically found in MANETs. However, as will be explained in Section 5.2.2, just recently the research community has began to study communication problems for networks of unmanned autonomous vehicles [47].

An important characteristic of this kind of networks is the extreme time variability of their topology. Due to this peculiarity a path routing algorithm should continuously recalculate its routing tables in order to cope with this issue. It is moreover infeasible to recalculate the entire path end-to-end in a reactive end-to-end way. A solution could however be found in a local routing tables redefinition. Many works, in fact, considered the possibility to directly include mobility-aware concepts into the routing algorithms, in order to foresee the topology variations. The OLSR protocol, for example, is a predictive routing protocol for MANETs that works upon a link-state index that indicates the host-to-host link quality. This link-state index can be designed in order to include also the vehicles' position and velocity [48]. In this way, when two nodes are moving closer, the communication link become more advisable; when instead the nodes are moving away from each other, the link perception worsens. Furthermore, in the estimation of the link-state index quality can be integrated not only by instant positions information, but also with long term link-stability indexes [49]. In this case each node, through a modified version of the OLSR control messages, disseminates the link stability and load indexes to the others nodes that are able to coordinate themselves by choosing communication paths that are more stable than others while balancing the communication load.

The Ad hoc On-Demand Distance Vector (AODV) is a reactive routing protocol that uses control messages to setup the source-destination route. The used control message is called Route Request (RREQ) message that is used to find the destination node. In order to include the concept of reliable distance in founding a good solution in unmanned vehicles network, the RREQ message mechanism can be modified to avoid routes that are not stable and, hence, only the stable paths are considered in the route creation operation [50].

Unmanned Aerial Vehicles are capable to implement a six-degrees-of-freedom control: three degrees of freedom for the position control and three for the orientation one. This feature leads to non-negligible problems for the lowest communication layers of the ISO/OSI stack. In fact, the vehicles' movements have direct impact on the main transceiver antennas' properties like polarization, orientation and directionality [44]. For these described peculiarities of autonomous flying vehicles, spectrum efficiency and spatial reuse (Figure 3.1) are very important features to be included into both the communication protocols and the mobility control system [51]. Moreover a deepened study is needed for the design of a context-aware communication link [52] in order to deal with the actual environment characteristics where the vehicles are deployed. An opportunity to increase the system performance for an UAV network is the use of multiple and/or directional antennas [53] [54] [55]. In order to exploit the possibilities that arise from the use of multiple and directive antennas, new MAC protocols to access the channel must be developed. With multiple antennas the control messages and the data packets can be sent on different channels. In the meantime, the directionality of the antenna can be used to increase the quality of the received signal, the transmission range and, hence, to improve the spatial efficiency of the transmissions. In order to exploit the directivity of the antennas, the vehicle must be aware of the destination position to steer the antenna's main lobe in the right direction. By using omnidirectional antennas [56] [57] or directional ones [58], the vehicle position is shared among the neighborhood enabling the right calibration of the directivity of the antenna towards the destination in order to transmit with the maximum efficiency.

The authors in [59] suggested the use of a switched circularly multi-directional antenna arrays in a fleet of UAVs where the main actor UAV (the leader) is at the center of the formation while all the others UAVs are positioned in circle around the leader. In this way by using the directional beam forming array antenna, the

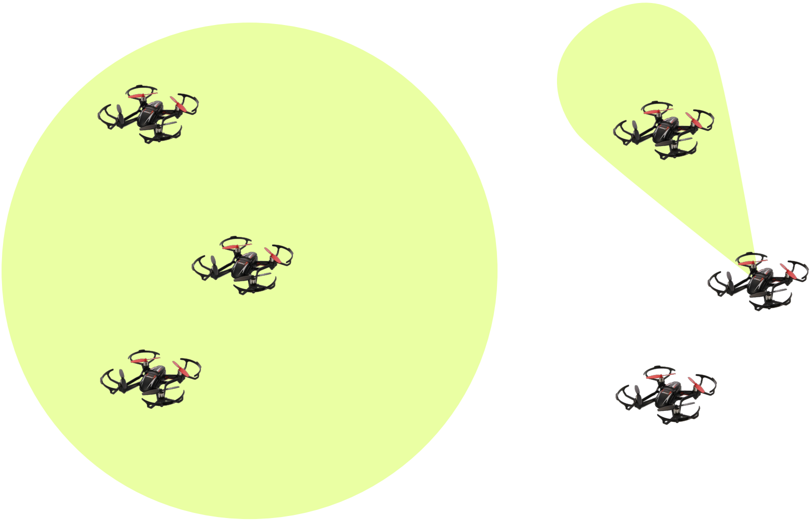


Figure 3.1: Example of spatial reuse using directional antennas (on the right), compared with an omnidirectional antenna (on the left)

proposed MAC protocols are able to implement spatial reuse in order to minimize communication interferences among the UAVs. Moreover, in order to exploit the presence of radio diversity resulting from multiple antennas, a network coding technique is developed in [60] where, without substantial increasing in complexity and overhead, the designed link-layer protocol is able to support high data rate for ground-to-air links.

In communication network protocols, guaranteeing service differentiation with Quality of Service (QoS) is of primary importance since it permits multiple kind of communication flows while enabling the deployment of reliable services. One of the main issue of UAV networks, is the network connectivity maintenance while executing the mission task (see Section 3.2.2). However even if that approach offer a minimum QoS in hop-to-hop communications, we need a more complex QoS system that is able to guarantee an higher level of quality for end-to-end QoS that include efficiency and reliability [61].
Services differentiation and QoS should be provided in accordance to the application needs that are then mapped in different priority classes. Furthermore, it can be implemented a resource reservation mechanism in order to guarantee the availability of the application requested resources. In this case, to manage the generated data traffic, the protocol is in charge of exchange signalling messages with the neighbours nodes in order to negotiate the requested level of QoS [62]. If it is possible to reserve the requested services, then the communication flow is admitted to start and the available resources are decreased. With this method,

only higher priority traffic will have reserved resources while the lower priority traffic will experience best effort modality. The QoS requirements can be satisfied also by introducing a centralized QoS management system that enable to design a QoS-aware data flow manager that is in charge of deciding the channel access strategies to use in order to fulfill the quality requirements [63]. These strategies are then disseminated in the UAVs network by the central entity end, hence, executed.
A different approach is given by [64], where a network transport layer is built on top of a robust routing protocol that is able to guarantee QoS for the communication flows. More precisely, the routing protocol is designed to provide for each robot a minimum guaranteed amount of data that it can inject into the network. To protect the robots to move in position where the QoS requirement are no more provided, the path planning procedure is made only inside a feasible space of possible configuration where the network integrity constraints are imposed. Finally it is proposed a modified version of the TCP protocol that uses the acknowledgement system hop-by-hop instead of end-to-end like the legacy TCP that exploit the QoS guarantee provided by the lower network layer.

To provide QoS in an unmanned vehicle network with high mobility, an efficient routing architecture is needed in order to cope with the constant network topology change issue. Geographical routing is one candidate solution that is able to define a route path basing the routing decisions on the position of the destination vehicle [65]. In this method, in fact, there is no need for the network to maintain route information, end hence is preferable over other routing solutions [66]. An important problem to consider for this kind of solution, is the accuracy and the reliability of the positions information, especially in the case of autonomous unmanned vehicles network where the average speed of the vehicles is, generally, very high. Furthermore, the knowledge and the management of the vehicles position is not a trivial problem [67] [68]. To bypass the problem of node locations management, a reactive routing protocol (like AODV) can be used to discover the destination node and to exchange location informations. After this step, all the nodes along the routing path have the knowledge of the destination location. Then, if the routing path become invalid because of link breakage, the geographical routing protocol is used as support method to back up the communications until a new route discover is invoked by the source node [69]. By knowing the position and velocity of the neighbours vehicles of the source node and of the destination node, it can be described a link quality index that specify the stability of the neighbours communication links compared to the destination position [70]. This indicator includes positions prediction, link expiration time, distance and direction of interception that rely on the relations between the source node, its neighbours and the destination. It hence define a stable routing decision mechanism that provide a reliable communication path for unmanned vehicle with high

velocity.
An hierarchical routing protocol can be built on top of a geographical one to improve the communication management in an airborne ad-hoc network. With this technique, in fact, the communication flows follow a routing path that is organized in multi-layers. In [71], in fact, is developed a two levels hierarchy architecture where at the higher lever there are the aircraft with a known mobility pattern, while at lower lever there are node with not predetermined mobility patterns. Obviously a network connectivity is not guaranteed at each time. The developed algorithm takes into account this possibility and enables the airborne at the higher layer to store the information until a path toward the destination is found, or a maximum time-out expires. Furthermore an expected zone is calculated to estimate the position of the destination node in order to improve the network routing.
The geographical routing protocols relies on the Greedy Geographic Forwarding (GGF) method that is in charge of forwarding the messages towards nodes that are closer to the destination, until reaching the target node. One of the main problem for this strategy is the so called *void node* (or *routing void*), i.e. the situation where a node, while routing towards the destination, has no neighbour that is closer to the goal. Different strategies were proposed for MANETs [72] [73], but the characteristics of an autonomous unmanned vehicles network bring various opportunities that can be exploited [74]. Given the high vehicles mobility, a strategy for the void node, could be just hold on the packet for few instants and then retry the GGF procedure. After the waiting time the probability that the fast topology changing will bring a neighbour vehicle to be in the direction of the destination, is high. Otherwise, another strategy is to forward the packet to a neighbour that is moving towards the destination or even send the packet to the further node if there is the suspect that the void node is into an impasse. Furthermore, since the probability of getting in the void node problem is dependent from the communication range, including the two-hop information in the GGF procedure will drop drastically the probability to end up in this problem [75].

At higher layer, one of the most common application in UAVs mission is the video streaming; in fact in both surveillance, monitoring and emergency scenarios, targets video streaming is requested to accomplish the mission task and hence the Quality of Experience (QoE) of the video is very important for the final results of the entire mission.
In spite of the large number of works that exhibit the video streaming as a use case, there are very few works that analyse in deep the video streaming aspect in a group of autonomous unmanned vehicles. Adaptive video streaming techniques are designed to be able to modify the transmitted video quality based on the transmission quality estimation [76]. This estimation is calculated considering the number and the delay of acknowledgement received for the video frames transmission.

In [77], instead, the authors faced the video transmission issue for FANETs by working on the routing level. In fact, they proposed a cross-layer protocol that enhances the the transmission of video flows by using a geographical routing protocol along with a path quality estimator that enable the protocol to foresee the link quality deterioration end hence being able to initialize a new path flow creation before the link expires avoiding video pause during the streaming video.

3.2.2 Team of autonomous vehicles with controlled mobility

If we look at the mobility control systems research area, we see that the main focus for the community is the problem of guaranteeing network connectivity while moving the vehicles in order to accomplish the mission task. For this reason, the designed control systems face the problem of modelling the communication characteristics in order to develop a control system that is able to deal with the communication constraints. We can call this kind of approach *communication-aware* mobility control system, since it includes, inside the controller, the knowledge of the communication peculiarities.

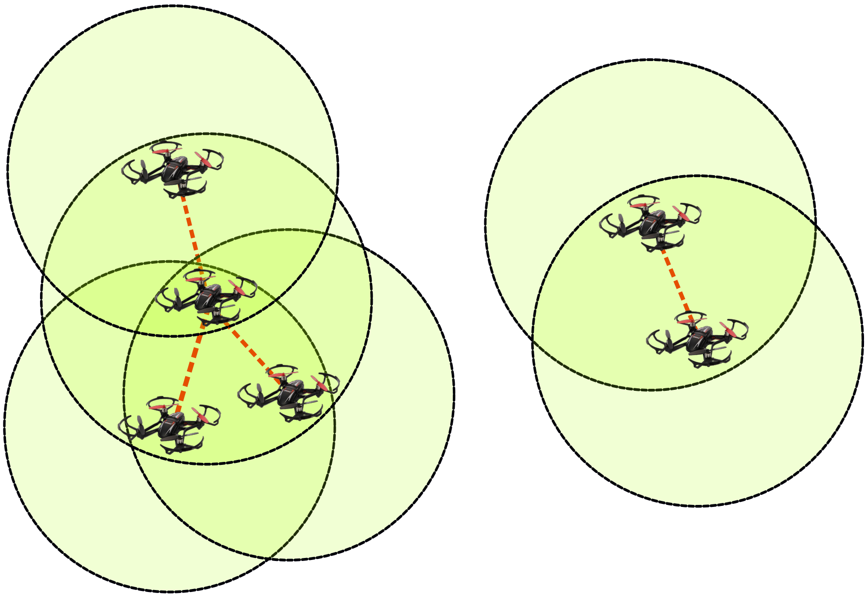


Figure 3.2: Model of the communication link by using a disk with a constant radius. The communication link exists only if the two vehicles are in the communication radios of each other.

The main approach to address this issue is to model the network communications ability of the unmanned vehicles with a connectivity fixed radius disk

area and then to create a motion control system including this communication constraint in the control design (Figure 3.2). With this approach, can be created motion control systems that are able to provide continuous network connectivity during the entire mission.
The possible applications for this kind of systems can vary form target visit and track to surveillance, from map exploration to area coverage. In order to maintain the connectivity among the vehicles in the network, the controlling input action range can be restricted to those inputs that guarantee the network connectivity also after the vehicle movement [78] [79] [80] [81]. In literature was proposed a set of control-theoretic method for connectivity preservation, ranging from spectral graph theory and semi-definite programming to maximize the algebraic connectivity of a network, to gradient-descent algorithms and hybrid systems to ensure topology control in a least restrictive manner [82].
Differently, if a link breakage is detected, recovery strategies can be activated in order to avoid the disconnection. For example, a leader-follower technique considers that the two vehicles belonging to the breaking link, become one the leader and the other the follower [83]. In this way the leader will continue with its movements strategy, while the follower stops its own movement and just follows the leader. Another technique to avoid network disconnection is to check the second eigenvector (λ_2) of the graph Laplacian [84]. This value is indeed an index that indicate if the communication network graph is connected or not. Furthermore, the Laplacian matrix can be extended in order to include in its definition not only the connectivity of the network graph, but also others parameters like desired link quality, obstacles and collision avoidance indexes and the maximum communication range [85]. Another way to include serveral properties in a single controller, is by designing a behavior-based approach that enable to compose a set of prioritized task functions, describing the elementary behaviors, in a single controller [86].

Given the distributed characteristic of the problem, nature-inspired approaches has the advantage over classical methods to solve computational complex problems [87]. The use of potential field to create repelling and attractive forces, for example, enable the design of motion control solutions that are self-organizing by using only local informations [88]. With this technique, in fact, is possible to steer the autonomous vehicles in a distance-safe zone where the vehicles are not too much close each others, to avoid physical collisions, neither too much away, to avoid network disconnections [89].

To make more realistic the modelling of the communication links, it can be used effective communication indexes, i.e. link budget, signal-to-noise ratio or bit-error-ratio, instead of a constant value for the vehicles inter-distances. The simplest link quality indicator is the Received Signal Strength Indicator (RSSI) that is the actual power arrived at the receiver. This value can be easly calculated at the receiver side and can be used, for exaxmple, for area exploration in a cooperative manner [90]. The link budget metric, also called fading margin, measures the communication reliability, i.e. it indicates when the link is going to break. This index can be used to model a virtual spring force that steers the vehicles in order to get the desired link quality while executing the main mission task [25]. Others used signal estimation index to model the communication link are either the Signal-to-Noise (SNR) or the Signal-to-Noise-and-Interference-Ratio (SNIR) [91]. These indexes can be used to estimate the channel capacity and hence design a mobility controller on top of it.

Modifying radio parameters while monitoring a given area, in fact, could be useful to improve the performance of the system [92]. Tuning in the right way the radio parameters could be the the ace in the hole for creating reliable functionalities in the local behaviour layer.

In real missions, maintaining the network connectivity among the unmanned vehicles could be very difficult or even impossible. This is due to the unpredictable events that could happen that are beyond the operational possibilities of the vehicles, i.e. failure of one or more relaying nodes, unexpected drop in the communication links quality, the number of unmanned vehicles are not enough to accomplish the mission, and so on. Furthermore in some kind of applications, full connectivity is not always necessary and hence a temporary disconnection of the whole network could be accepted. Therefore the network protocol and the motion control system musty be aware of this breakage possibility and avoid to stop the mission if it could be the opportunity to end. The designed systems should work in sparse connectivity environment where the communications among the autonomous vehicles happen in smaller time frames whenever connections become available [93]. These kind of systems are called *Delay and Disruption Tolerant Networks* (DDTN).

For spatial exploration mission, for example, a release and return strategy (RRS) for a fleet of UAVs can be deployed such that the autonomous vehicles can temporary disconnect from the ground base station and than reconnect to it after fulfilling the mission operations [46]. This strategy is activated only if the fleet of drones is not able to explore the whole area maintaining the connectivity constraint; in fact a communication aware potential field strategy, based on the RSSI index, is used to maintain the connectivity among the swarm, but when the system detects that it is no more able to accomplish the task, it releases the swarm from the ground base station, allowing the UAVs a free exploring of the whole area. At the end of the mission, the UAVs will return to the base station in order to transmit the newly acquired data.

If both fixed relay stations and UAVs are deployed for the mission, a communication-aware path planning can be designed to use the fixed network to relay the information towards the base station avoiding the UAVs to constantly come back to deliver the data [94]. Since the fixed relay stations are used only for mission support, the UAVs are able to deal with unexpected scenarios, like failures in the fixed relay network, where the UAVs can autonomously recalculate the path and hence accomplish the task.

3.3 Unmanned vehicles support for emergency wireless networks

Even though there already exists a lot of research in connectivity support missions, there are many open research issues.
One of the main lack in this field is the absence of a generic architecture for the design of the next-generation wireless devices that is able to face with an emergency scenario. These devices need to be provided with self-organizing and self-configuring capability, since they must deal with an emergency scenario that is in general unknown. Furthermore, an emergency scenario has the characteristic of shortage in communication infrastructure. For this reason the public safety agencies need a temporary communication network that is capable of fast deployment with autonomous configuration ability in order to help the rescue operations. The use of unmanned vehicle is a prominent solution for a fast creation of support communication networks.
Given the pervasive presence of smart devices among the end-users, the creation of spontaneous wireless networks is a promising solution to implement an emergency communication network. With this technique is possible to increase the coverage of the emergency network increasing the possibility to reach disconnected survivors in order to coordinate with the rescue teams.

These issues are addressed in this thesis by giving a general architecture, called STEM-Node, for an unmanned vehicles equipped with configurable wireless network devices. With this general architecture is then proposed a series of solutions to deal with the numerous problems that arises when tackling with emergency scenarios: from restoring the wireless network connectivity to the exploiting of the radio frequency resources, from taking advantage of the presence of smart devices to the design of energy efficient algorithms to support such emergency wireless networks.

Part II

Networked robotic systems for emergency scenario

Chapter 4

Emergency scenario peculiarities

In this chapter will be described the main characteristics of an emergency scenario and the communication challenges that must be managed in these particular environments.

4.1 Main characteristics

In several reports from emergency management agencies, the first 72h aftermath the occurrence of a disaster, also referred as “Golden 72 hours” [95], are considered the most critical hours to organize first response actions and to save human lives [96]. Recent man-made or natural disasters (e.g. the Fukushima disaster in Japan, the Katrina hurricane in USA, the Emilia and the Amatrice earthquakes in Italy) have demonstrated worldwide the fragility of the cellular infrastructure and the challenges of coordinating rescue operations among safety organizations when different communication technologies are used [97, 5].

In this emergency context, a key role is played by the communication network, that must be fully operative in order to enable rescue teams to coordinate operations and to keep the population informed [2, 98, 99]. In these disastrous scenario, in fact, the life of the trapped survivors is strictly dependent on the speed of the rescue teams. For this reason, the emergency communication network is a life saving support system that provides emergency facilities to survivors and helps the support operations to the rescue teams [95, 100].

Unfortunately, this requirement is far to be guaranteed by terrestrial communication infrastructures, whose fragility has been confirmed by most of recent (catastrophic) events worldwide.

We can distinguish three main characteristics in a post-disaster scenario [101]:

- *Unpredictability*: the main disasters such as earthquakes, landslides, eruptions and tsunamis are natural calamities that are very difficult to predict, especially with a prediction time and accuracy that would enable appropriate countermeasures [102];

- *Limited resources in disaster areas*: due to the unpredictability of the natural event and to the collapse of the main connection facilities, it is very difficult to reach the disaster area to bring the resource needed like food, water and communication resources [103];
- *Dynamic changing of the environment*: the situation in a disaster area is dynamically changing, including the movement of people, and hard to predict due to the extraordinary event that happened and the specific peculiarity of the actual disaster area [103].

Given these communication challenges, there is strong motivation towards the realization of backup communication systems that are able to quickly self-deploy in the aftermath of an emergency and ensure temporary network services in the affected area. One of the possible scenarios can be the one depicted in Figure 4.1 where the rescue teams are connected to the headquarter through either backbone networks or satellite communications. However, due to the effect of the disaster, some or all of the network infrastructures can be damaged and hence, this kind of network structure is not feasible for fast response in emergency scenario.

One big issue is constituted by the occurrence of network partitions caused by damages to the communication components (e.g. base stations), and by the lack of adequate self-healing capabilities of devices and of the infrastructure. A dramatic evidence of this fact was provided by the Japan earthquake in 2011: the total number of cellular base stations damaged overcame the 14,000 units, while fixed network lines out of service reached 1 million units [97, 99]. Network bandwidth constitutes also a key issue, since peak traffic demands generated by mobile devices owned by survivors or by rescue teams are usually observed in the aftermath of a disaster. In fact, in the less destructive scenario (e.g. the earthquakes in Italy), the cellular infrastructure was not damaged, but its functionalities were severely compromised by the exceptional traffic load generated by end-users, so that both conventional and emergency communication services could not be supported [105, 106].

Since, current 3G/4G cellular networks do not foresee dynamic bandwidth management schemes, conventional and emergency communication services might likely be disrupted as a consequence of the excessive traffic demands. On the other side, in more apocalyptical scenarios (e.g. the Great East Japan earthquake), the rescue teams faced a chaotic scenario where most of communication infrastructures were disrupted, and the wireless network was partitioned into several islands of connectivity [107, 97, 5]. Furthermore, it took nearly two months to recover the communication infrastructure after the Great East Japan Earthquake

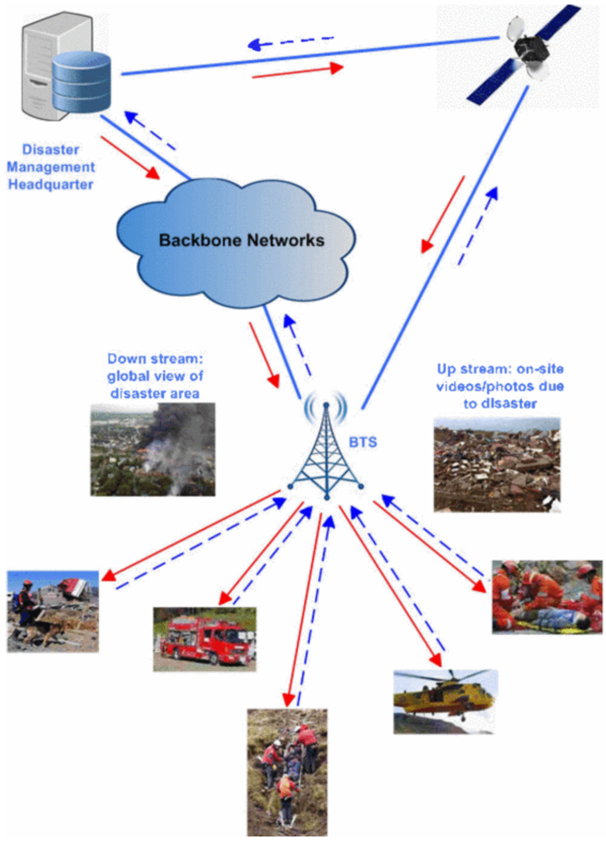

Disaster Management Headquarter
Backbone Networks
Down stream: global view of disaster area
Up stream: on-site videos/photos due to disaster
BTS

Figure 4.1: An actual possible solution of relief operations of a post-disaster scenario [104].

[108]. Moreover, some area can have insufficient presence of cellular base stations and other traditional communication infrastructure, like in Nepal, where after the earthquake that had a magnitude of 7.8 and epicentre at Lamjung [109], teams from all over the world thronged Nepal for offering rescue and relief services, but they had communications difficulties for the rescue operations due to the intermittent connectivity provided by the actual network infrastructure [110].

4.2 Communication Challenges

To overcome the communication limitations caused by the emergency scenario, most of Protection for Disaster Relief (PPDR) organizations rely on alternative mission-critical communication systems (e.g. TETRA and TEDS [16] in Europe) that support device-to-device and multi-hop communication. However, the utilization of such technologies is limited to specific applications (voice communication), and is not suitable to create large-scale emergency networks, due to the technological limitations in terms of coverage, bandwidth, and interoperability with heterogeneous devices. Furthermore, in a post-disaster scenario network may be intermittently connected, providing separated ”islands” of well-connected nodes, that are not connected with nodes in other partitions [111, 25]. In fact, different search and rescue teams and groups of survivors can be separated from each other and, hence, are unable to communicate each other. Moreover, due to the mobility of the rescue teams, the network topology is constantly changing. Starting from the unique requirements of post-disaster scenarios, and the limitations of existing technologies, most of researches on wireless systems have investigated how to deploy wireless networks with self-configuration and self healing capabilities, in order to guarantee service continuity also in case of failures of specific network components [112].

The communication networks established during an emergency, also called Emergency Communication Network (ECN) [101], can be mainly categorized into two types: infrastructure based and infrastructure-free. The infrastructure networks are based on the assumption that the communication infrastructure remain active also after the disaster, i.e. cellular networks and users local area networks [108, 113]. These kind of solutions are, however, difficult to preserve after a disaster where the communications infrastructure is seriously damaged. The infrastructure-free approaches have, instead, more flexibility and can restore damaged communication networks where response operations can be established in an ad-hoc manner [114, 115]. The actual proposed solutions are Mobile Ad Hoc Networks (MANETs) [116, 117, 118], Delay Tolerance Networks (DTNs) [119, 120], Movable Base Stations (MBSs) [107, 121, 122] and more complex system that include low altitude tethered balloons [123, 124].

However, besides the well-known results from the literature on Multi-hop Ad Hoc Networks (MANETs) [125, 126, 127, 128] and hibrid cellular-MANET systems [129], we register an increasing interest on two novel enabling technologies of self-organizing wireless systems, i.e., (i) emergency networks composed of wireless devices with dynamic spectrum re-configuration capabilities [113], and (ii) emergency networks composed of wireless devices with autonomous mobility [130]. In the first case, emergency wireless networks are deployed through multi-stack Cognitive Radio (CR) devices [3, 131] which are able to adjust their transmitting parameters at the MAC/PHY layers and to interconnect heterogeneous devices [132, 133]. Several recent studies have proposed novel MAC protocols and spectrum sharing models enabling the opportunistic utilization of licensed band in order to support the Quality of Service (QoS) requirements of emergency services [2, 134, 5, 135, 136, 137, 138]. In the second case, self-organizing emergency networks are deployed through the utilization of small, inexpensive unmanned terrestrial and aerial vehicles [98, 139, 140, 130]. To this purpose, several distributed swarm mobility algorithms have been proposed for coverage and recognition applications [141].

Given the advantage of using the aerial unmanned vehicles in emergency scenarios, researchers are focusing the attention also on the LTE technology with the use of Aerial evolved Long Term Evolution Base Stations (*AeNBs*) [142] in order to increase the capacity of the communication infrastructure [143] and give robustness to the communication network [144].

Chapter 5

Architectural design for future systems

In the previous Chapter 4, it was depicted the peculiarities of a post-disaster scenario and the communication issues that needs to be addressed in order to speed up and improve the search and rescue operations.
In order to deal with the challenges introduced before, in this Chapter will be introduced the architectural perspective for the design and the study of the *next-generation networked robotics systems*.

5.1 Self-organizing systems

5.1.1 The STEM-Network idea

In this section will be introduced the concept of the staminal node which constitutes a novel and effective approach to realize wireless systems with self-organizing capabilities.

The stem network (STEM-Net) paradigm [145] foresees the realization of a next generation of wireless devices, called Stem Nodes (SNs), that are capable of self-configure at multiple layers of the protocols stack on the basis of their hardware configurations assuming multiple network roles (i.e. gateway, relays, etc.) and can evolve during its lifetime. These multi-purpose characteristics bring the SN to match with the biological counterpart. To the well-known properties of self-configuration and self-organization of distributed systems [146, 147], the architecture of a SN adds the following novel features [148]: (i) mutation, which refers to the ability of a SN to self-configure and switch among the supported roles based on the system and application needs, (ii) cooperation, which refers to the ability of swarms of SNs to implement system-wide behaviors and (iii) evolution, which refers to the ability of a SN to upgrade its own software characteristics and hence, assume a role it was not initially designed for.

The STEM-Net potentiality will bring multiple advantages compared with more traditional approaches:

- *Evolvability (evolution ability).* Stem Nodes can extend their specific skills by adding new roles and behaviours through information exchange and cooperation with other devices in the neighbourhood. Technologies out-date fast. The capability of stem nodes to reconfigure their own software capabilities will face this issue.

- *Bridging the heterogeneity and fragmentation of available network technologies.* Given the evolutionary and adaptive nature of a SN, the stem network will be able to connect heterogeneous devices without the need of specific devices for each different wireless technology.

- *Adaptability.* Stem nodes adapt the communication parameters at each layer in the protocols stack to face the dynamic network conditions. Moreover, they can also take advantage from their controlled mobility capabilities to guarantee better QoS or to reduce energy consumption.

- *Distributed control.* Stem nodes have self-managing and self-configuration capabilities. They can dynamically adapt their features and behaviours by taking local decisions and exchanging information with neighbours without the need of a centralised control. Swarm intelligence philosophy are applied to make possible simple nodes swarm able to cooperate and accomplish complex tasks.

- *Resilience and fast response.* In emergency or anomalous situations, some mobility-controlled stem nodes provided by network operators can promptly react in order to restore connectivity and help the first responders.

- *High reliability.* Stem nodes can use different available communication interfaces and programmable software modules to achieve a high fault tolerance level and reliable communications.

5.1.2 The architecture of a single STEM-Node

The described properties of a stem node are accomplished through the composition of five capabilities: (intra-stack) protocol reconfiguration, multi-homing or inter-stack protocol reconfiguration, learning, cooperation with other devices and controlled mobility. The protocol reconfiguration capability of a stem node depends on its built-in features (e.g. its hardware) and can be implemented by a combination of advanced solutions [such as cognitive radio technologies on software-defined radios (SDRs)]. These technologies are capable of reconfiguration process and of learning capabilities (e.g. reinforcement learning and swarm intelligence) that drive and select the reconfiguration process, possibly in cooperation with the other nodes.

Some stem nodes can be equipped with motion capability (e.g. mobile sensors and robots) that can move the node in 2D or 3D. In this case, the controlled mobility feature can characterise the node with an additional reconfiguration dimension. Figure 5.1 illustrates the logical high-level architecture of a stem node. Without loss of generality, it can be assumed that each stem node is provided with

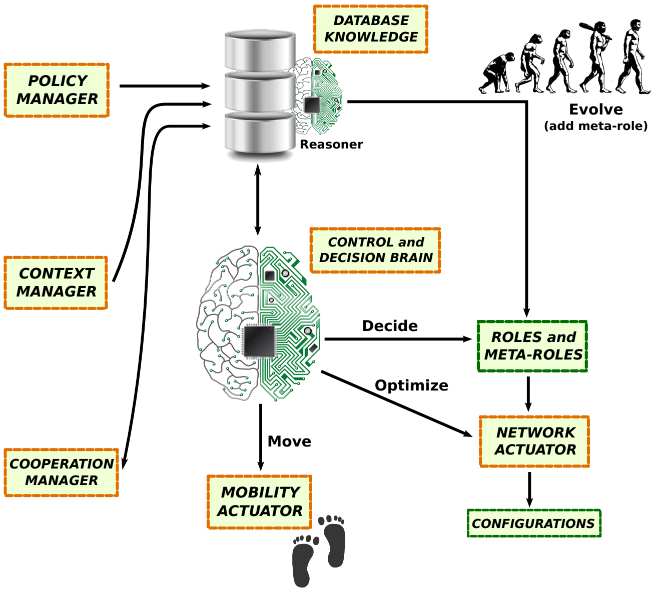


Figure 5.1: The architecture of a Stem-Node

a set of M roles $R = r_1, r_2, r_3, \ldots r_M$ that the node is capable to play. A role denotes the network capabilities/functionalities that the device can provide. The set of available roles depend on the node characteristics, such as the hardware features and the physical constraints (e.g. the number and the type of available radio interfaces in a multi-radio stem node). Besides these hardware-specific roles, a stem node can also play some “meta-roles”. A meta-role identifies a capability that is not currently available at the stem node and can be learnt through cooperation with other nodes. From the practical point of view, a meta-role is enabled by a software module, named *evolution module*, that can be dynamically downloaded from other stem nodes and locally installed in order to expand the capabilities of a device. This process can be viewed as an evolution of the stem node, because it dynamically expands the device’s functionalities. As a result, for example, a stem node can dynamically change the routing or transport protocol in use, upload the radio interface drivers or offer a new Web Service. Each time a new software module is transferred to a stem node, a new meta-role is added to the set of meta-roles *MR*= $m_1, m_2, m_3, \ldots, m_{M'}$ that the node is capable of offering. During its operation, a stem node can decide the role $r_i(i = 1..M)$ or meta-role $m_j(j = 1..M')$ to play, on the basis of the environment, the user's preferences and the system requirements. The decision process is performed by the *Control and Decision Brain* (CDB) module, which implements the cognitive component of the stem node. The process of role and meta-role switching is a mutation of the stem node, because it alters the current device's profile without modifying its fixed or acquired capabilities.

A stem node can be associated to a set of $C = c_1, c_2, c_3, \ldots, c_N$ possible configurations. A configuration denotes a combination of parameters at each layer of the protocol stack, from the physical up to the transport layer. Each role r_i or meta-role m_j is mapped to a specific network configuration $c_k(k = 1..N)$. Moreover, for each specific role/meta-role, a stem node may modify its configuration over time for self-optimisation purposes (e.g. switch from c_k to $c_{k'}$).

In the following, a description of each component of Figure 5.1 is provided:

- *Context Manager (CM)*. This component has the task of sensing and acquiring knowledge of the surrounding environment. Here, the concept of context is quite general and includes (i) the network environment (e.g. available wireless connections); (ii) the state of available resources (e.g. used frequencies); and (iii) feedbacks and metrics that provide qualitative and quantitative evaluation of the status of a stem node (e.g. channel quality).

- *Database Knowledge (DBK)*. This module is used by the stem node to store data about the context and acquire knowledge about the best role, and hence the configuration, to play in each context in order to meet the policy requirements. Stored information can help in recognising similar contexts in the future and in quickly selecting the role/configuration to play on the basis of previous experiences.

- *Policy Manager (PM)*. This component contains the current policies of the stem node, that is, the set of goals and requirements that guide the decision process and determine the node behaviour. Specifically, data inputs for the decisions of the PM can be user-specific preferences and needs (e.g. minimise costs of network connectivity), application requirements (e.g. bandwidth guarantees), network and context requirements (e.g. need to replace a network component) or other stakeholders' policies.

- *Control and Decision Brain (CDB)*. This is the decisional component of the stem node. On the basis of the current context provided by the CM, the stored knowledge in the DBK and the policy requirements in the PM, this component decides the role that the stem node must play in a set of available roles. Given the current role and configuration and relying on feedbacks from the CM, the brain determines the configuration adaptation to best match the requirements of the PM (the “Optimize” arrow in Figure 5.1). In case of a stem node that supports controlled mobility, the brain is also in charge of deciding and controlling the optimal movements of the mobile node.

- *Network Actuator*. This component implements the type of network adaptation decided by the CDB, by (i) loading the configuration associated to a specific role or (ii) performing a configuration adaptation for self-optimisation.

- *Mobility Actuator*. This is the component that is in charge of adjusting and controlling the node’s location, on the basis of the decisions of the brain.

- *Cooperation Manager (CoM)*. This is the component through which the stem node implements coordinated and collaborative operations with other stem nodes. The CoM is responsible of (i) managing information sharing with other nodes; (ii) updating the DBK on the basis of the information received from other nodes; and (iii) discovering and updating the list of meta-roles that can be played by the stem node.

Functions and enabling technologies

A lot of research has been conducted on self-configuring distributed systems, technologies for reconfigurable devices/networks and methods to provide the wireless nodes with cognitive and intelligent capabilities. Most of these solutions can be used for the realisation of the core components of a stem node, as discussed in the succeeding text.

Knowledge representation A stem node needs a large dataset to control its operations, including information about the environment, the experience from previous contexts and the feedbacks from other stem nodes. Since data can be made available by different STEM-Net components, methodologies to represent, share and access data in a uniform and efficient way must be considered for the design of a stem node. To this purpose, ontologies have emerged as a powerful technique to represent semantic knowledge and the relation among concepts in a non-ambiguous way [149]. Moreover, semantic engines can be implemented on top of ontologies to perform inference from stored knowledge in order to produce new data, which in turn can be used by the brain controller to guide the reconfiguration process.

Control and decision Control functionalities of the brain are enabled by decision-making algorithms that guide the reconfiguration process of each stem node. In the case of a multi-radio stem node, for example, equipped with Wi-Fi and 3G-4G radio interfaces, the brain could determine the time-schedule activation of the Wi-Fi/3G-4G radios and whether the node should mutate and temporarily serve as a Wi-Fi router (e.g. to extend the access network coverage). Among the approaches proposed in the literature, machine learning techniques [150] fit well the need of a stem node and are useful to determine the optimal configuration/role on the basis of current and previous experiences of the node itself. More specifically, reinforcement learning (RL) algorithms [151] are suitable decision-making strategies, because of the explicit representation of state/action pairs and the adaptiveness to changes in the environment. Moreover, the RL paradigm can be extended to a cooperative scenario where the stem nodes exchange the information learnt and collaborate to the implementation of system-wide policies and network behaviours. Genetic algorithms is another well-investigated area in self-organising systems that can be used to drive the selection/discovery of the optimal configuration of transmitting parameters [151].

Cooperation In nature, cooperation is an effective strategy to implement system-wide behaviours and enable self-organisation of groups of individuals (e.g. ants and bees) with limited capabilities [152, 153]. This is also the case of a stem network composed of heterogeneous devices with limited communication and computation capabilities, which are able to accomplish complex tasks through cooperation with other nodes. In STEM-Net, node cooperation techniques are investigated for the purpose of (i) self-optimisation, that is, stem nodes exchange information regarding their own characteristics (such as motion capability, channel quality and residual energy) and context parameters, in order to reduce the uncertainty about the environment and speed up the role learning process [150] and (ii) network creation, that is, stem nodes adapt their role on the basis of network-wide metrics and requirements such as maximise network coverage and increase reliability.

Evolution A node meta-role and the related behaviour are implemented through some evolution modules that are dynamically loaded and unloaded by a stem node and can modify its configuration. For this goal, a stem nodes can transfer the evolution modules to other stem nodes, making them able to acquire new behaviours and eventually mutate their own operations. To accomplish these goals, three main functional issues must be considered: (i) the design of control units that guide the evolution of single stem nodes and the whole network; (ii) the definition of cooperative dissemination strategies (e.g. based on flooding, epidemic routing or publish-subscribe approaches) for distribution of the modules in a stem network; and (iii) the design of security measures to avoid the propagation of malware/software threads contained in the modules.

Controlled mobility Controlled mobility represents an optional reconfiguration dimension for stem nodes equipped with motion capabilities. In STEM-Net, mobility can be considered as a feature to be exploited instead of a challenge to be faced [154]. The potential of controlled mobility can be exploited in many ways, for example, (i) to meet the QoS requirements of specific stem nodes (e.g. by moving them to less-interfered areas); (ii) to improve network performances by adjusting the topology according to system-wide policies implemented through collaborative behaviours; (iii) to enable a stem node to accomplish new tasks through context-based mutation (e.g. possibility for a stem node to add new meta-roles by moving towards specific nodes/locations); and (iv) to provide the access network with additional communication resources. Given the importance of the controlled mobility capability, a more deep analysis of the impact of the autonomous mobility on the communication network, will be discussed in the next chapters.

Radio device reconfiguration Currently, several technological solutions enable the deployment of wireless nodes with on-demand reconfiguration capabilities. Here, we distinguish between the intra-stack protocol reconfiguration capability, that is, the possibility for a stem node to dynamically adjust some/all of its parameters at different layers, and the multi-homing or inter-stack reconfiguration capability, that is, the possibility for a stem node to dynamically switch among different stacks/access technologies (e.g. Wi-Fi/WiMAX). For this latter issue, multiple stacks can be supported on the same radio interface by using programmable platforms, such as SDRs. Session persistence during the handover can leverage the current and future outputs on media independent handover from IEEE 802.21 [155] and more recent improvements [156]. For intra-stack reconfiguration, cognitive radio [3] and dynamic spectrum access constitute promising approaches to deploy highly reconfigurable stem nodes able to dynamically adjust transmitting parameters at multiple layers of the stack, including the medium access control (*MAC*) and physical layers.

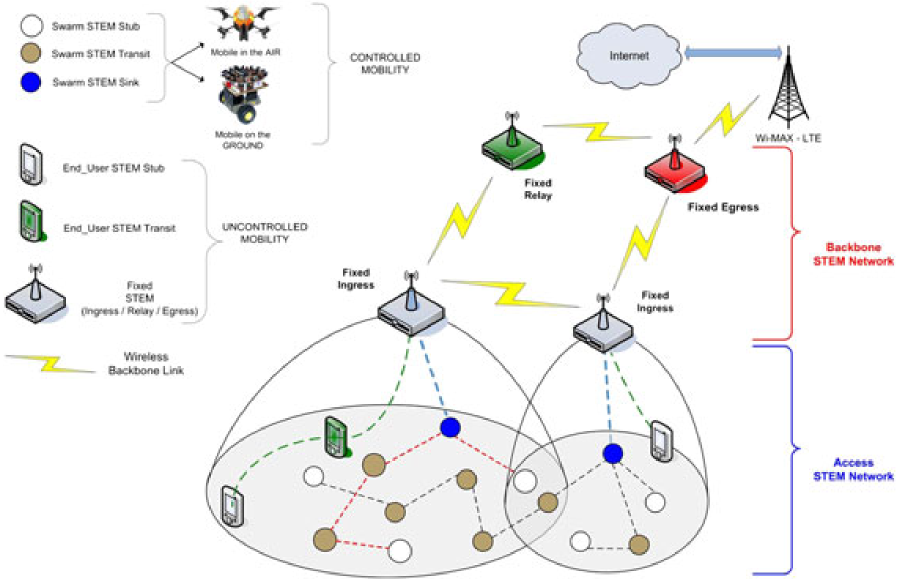


Figure 5.2: STEM-Net architecture [148]

5.1.3 The main architecture

The architecture of a STEM-Network is depicted in Figure 5.2. It includes two deployment layers: the *access STEM network* (ASN) and the *backbone STEM network* (BSN) [148].

Basically, the devices belonging to the BSN are based on well-established wireless technologies such as UMTS, WiMAX, Long Term Evolution and so on. The roles played by these devices can mutate/evolve from *ingress* to *relay* to *egress* nodes. An ingress node offers connectivity to the nodes in the ASN, a relay node participates in data forwarding between ingress and an egress node typically acts as a gateway to the Internet. The ASN is composed by nodes with controlled mobility, called *Swarm STEM Nodes* (S-SNs), and nodes with uncontrolled mobility, called *End-User STEM Nodes* (EU-SNs). Each S-SN has the ability to move and coordinate with other S-SNs in order to create a group devoted to a specific task. An S-SN can be elected to the role of *Swarm STEM Transit* to relay data to other nodes, or it can mutate/evolve into a *Swarm STEM Sink* that communicates directly to the BSN. EU-SNs are user-owned wireless devices (such as smartphones, PDAs, etc.) whose mobility is not controllable because it is user dependent. In analogy to S-SNs, also EU-SNs can work as *End-User STEM Transit* nodes and extend the network access beyond the single-hop or as *End-User STEM Sink* (EU-SS) nodes with multi-home capability to interface with the BSN.

Figure 5.3 shows examples of the lineage of stem nodes belonging to the ASN and the BSN. A line corresponds to a different class of stem devices (S-SNs,

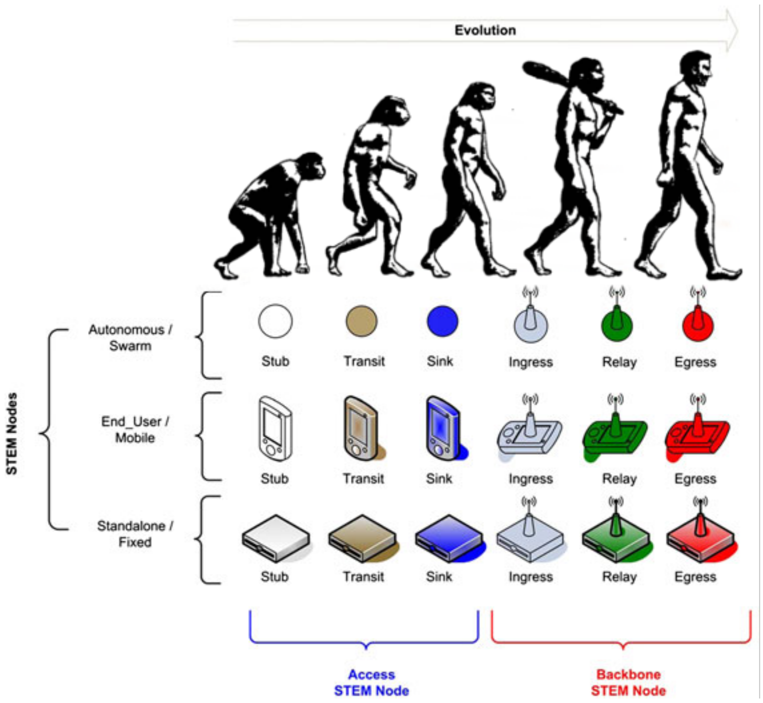


Figure 5.3: Examples of evolution line of a stem node

EU-SNs and fixed nodes). On each line, an evolution step (e.g. evolving from stub to transit and from transit to sink) occurs each time a novel capability is acquired by the stem node after the corresponding meta-role is learnt. During evolution, the old capabilities are not deleted; this implies that a stem node with sink capabilities can temporarily degrade and mutate to a stub node, on the basis of the decisions taken by the CDB module and of the policies managed by the PM module. Of course, the evolution should be supported by the hardware capabilities of the device (its set of built-in roles), which implies that not all the stem nodes might be able to complete the full evolution path namely, an EU-SN not equipped with a BSN interface will not evolve to an EU-SS.

5.1.4 The STEM-Network in a disaster recovery scenario

Environmental disasters could damage the network devices/infrastructures and prevent city-wide connectivity. The survivors' devices could be isolated because of the heterogeneity in the protocol stacks of devices in reciprocal coverage or because of the lack of the networking infrastructure. As a first reaction, the surviving devices create a spontaneous communication infrastructure that conveys some basic information to aid diagnosis or detect dangerous situations. Each S-SN or

EU-SN can self-reconfigure according to its features/roles in order to establish connectivity with neighbours and define a routing path towards an access point. However, there might be cases in which some devices, which have succeeded in creating network branches, lack of the necessary interconnection with the rest of the system. In this case, more sophisticated and advanced devices deployed by the rescue teams, supporting connectivity and controlled mobility capabilities, are sent to the disaster area with the task of restoring connectivity between the network segments created by S-SNs or EU-SNs in an autonomous and self-organised way. This is shown in Figure 5.4, where the nodes tagged as recovery devices represent autonomous nodes with different tasks and specific positions to be filled within the network.

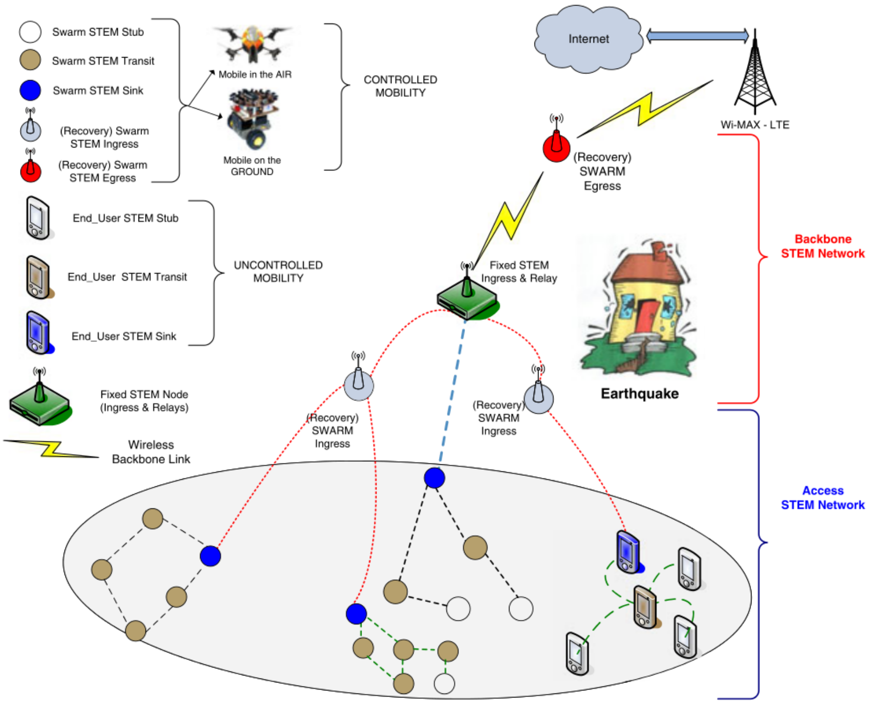


Figure 5.4: The STEM-Network in action in a disaster recovery scenario [148]

5.2 Networked behaviors

5.2.1 Architectural design for networked behaviors

The use of unmanned aerial vehicles in emergency operation, brings new challenges in coordination and cooperation in a fleet of drones. The motion control system of a vehicle, in fact, needs a specific algorithms that are able to manage the vehicle mobility in order to accomplish the actual mission. Furthermore, as explained in the previous Section 3.1.1, the communication issues that arise working with a fleet of unmanned aerial vehicles, need new network paradigms in order to be addressed.

In this case, an enabling technology for this kind of structural conglomerate consists in a set of efficient *communication methods* supporting *mobility control* and vice-versa. Therefore, recently, the domains of Robotic Control and Networking research have merged in the field of *Networked Robotic Systems* [33].

However, the actual works present in literature can be mainly categorized in two main groups: (i) *communication-aware mobility control systems* and (ii) *mobility-aware communication protocols* (see Chapter 3.2). The former includes the mobility control systems that deal with the communication issues as a constraint, while the latter includes the communication protocols that are designed to work with the mobility peculiarities of a fleet of aerial vehicles. There is, hence, a lack of research in mixed systems where *communication networks* and *mobility control* are designed together in order to reach more advanced *communication-control* systems. For this very reason, innovative solutions need to be used to approach the discussion of a topic so inherently vast. In this Section is presented a novel architectural design that is able to interconnect the communication network systems and mobility control system.

In Figure 5.5 is depicted a propositional general architecture for this representation. This architecture is composed of two vertical planes and three traversal planes. In the vertical planes, the two main topics can be visually identified: on the left lies the *communication* part, with a classical TCP/IP stack subdivision; on the right, three levels of mobility control lie instead. These levels are (i) the *micro*, (ii) the *macro* and (iii) the *mission-level* mobility controls. At a transversal level, a set of new *behavioral* planes, that connect the others horizontally, can be identified: *local*, *global* and *mission-based* subdivisions, referred as *networked behaviors*. In the following subsections we will individually describe our proposed compartments.

5.2.2 Communication Protocols

In this section are described very briefly the layering system used in the TCP/IP communication networks. For further details, the reader can refer to [157].

- *Physical* - This layer is dedicate to transmit physically the data from the transmitter to the receiver. Here the main issues are mechanical and electrical that must face with the physical transmission medium, which lies below the physical layer.

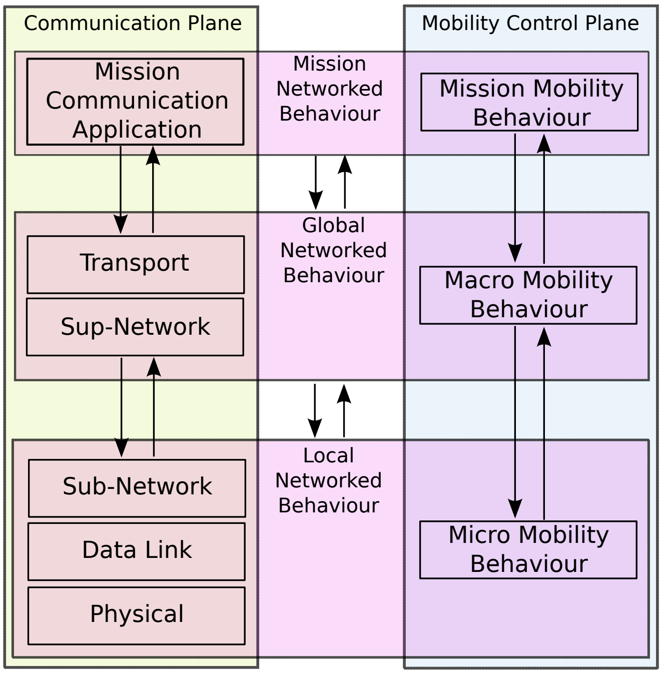

Communication Plane
Mobility Control Plane
Mission Communication Application
Mission Networked Behaviour
Mission Mobility Behaviour
Transport
Sup-Network
Global Networked Behaviour
Macro Mobility Behaviour
Sub-Network
Data Link
Physical
Local Networked Behaviour
Micro Mobility Behaviour

Figure 5.5: *Networked behavioral architecture* to interconnect the communication network protocols and the mobility control systems

- *Data Link* - The data link layer must hide all the physical problems to the higher levels. In fact it is in charge of choose the right configuration for the physical device in order to optimize the transmission performance. Furthermore, among the others issue that this layer must face, it has timing issues to respect in order to share the common transmission medium among all the other transmitters. To deal with this problem there is a special sub-layer called *Medium Access Control (MAC)*.

- *Network* - Generally the nodes in a network are not all connected directly. This produce the problem of finding a path through the network that is able to reach the destination starting from the source through a multi-hop transmission. The network layer is in charge of solving this problem of determining how packets are routed from source to destination.
  Given that the communication networks on which we are focusing our attention are composed by nodes provided with an autonomous high mobility, we split, in our analysis, this layer into two sub-layers: *sub-network* and *sup-network*.

  – *Sub-Network* - This layer is in charge of taking the next-hop routing decision using only local informations and, hence, without having the knowledge of the full multi-hop path between the informations' source and destination. An example of a system for this layer could be the decision engine in a routing protocols that assigns the local next-hops. In a fast-mobility, *highway for VANETS* or platooning scenario, systems implemented at this lever provide the response to the small changes a node could detect from its neighborhood. Given that the rapid change of the network topology, leads to continued disruption of the network's wireless links, a fast path change should be forecast in this layer to avoid issues in the communication flow.

  - *Sup-Network* - In the highest part of the network layer, the locality is decreased and thus, systems implemented at this level reflect the wider scope of actions. The changes happen at a slower pace compared to the other part of the Network layer. In fact, the protocols at this level must have a global knowledge of the network to calculate the communication flow path from the source to the destination.

- *Transport* - The main functionality of this layer is to accept data from the higher levels, split it up if necessary, send them to the lower layer and hence ensure that all the pieces arrive correctly at the destination. The protocols that lie in this layer are characterized by a special functionality that they offers to the higher layers: reliable connections, best effort delivery systems, flow and congestion control and so on.

- *Application* - On top of the transport layer is the application layer. It contains all the user-level protocol able to, for example, streaming videos, transferring files, browsing the Web and so on.

In literature, *mobility-aware communication protocols* for unmanned vehicles have been proposed to include mobility considerations into the protocol design as in [158] [159] [160].

Already from their definition, the protocols belonging to this family incorporate some knowledge about the mobility of the nodes on where they are run. They, in detail, try to anticipate the network topology changes and provide positional cooperation among nodes. This kind of approach has, however, no active control on the mobility and thus the former has to remain passive.

5.2.3 Mobility Control Systems

The movements logics of an autonomous mobility control system can be categorized by recalling the behavior-based control paradigm [161], where the key idea is to develop a set of controllers (i.e. *behaviors*) and then switch among them in response to environmental changes. In this way the over-planning introduced by one-size-fits-all controllers is avoided. Three different transversal planes can be selected to classify the various degrees of locality in mobility control, namely: (i) micro-mobility behavior; (ii) macro-mobility behavior and (iii) mission-mobility behavior. These three layers represent and can be linked to the different levels of locality and scope for the mobility control.

- *Micro mobility behavior*: this layer encompasses all the controllers whose actions scope can be considered local, reflecting the way the vehicles interact with the surrounding environment. The needed information can be retrieved just from the neighbourhood. The motions involved can, at the end, be generally summarized in micro steerings that allow the controller to complete its task. The decisional choices behind the control of this kind of mobility lie into the least abstract reasoning and are linked, for instance, to phenomena directly observable by the nodes.

- *Macro mobility behavior*: The controllers designed at this level have the possibility to make their object freely roam the environment and hence plan movements on a larger scope. In this scenario, the information about the surrounding environment follows the scope of the controller, enabling, in this way, long-term time-frames for the vehicle movement strategies. These action scopes are however concerned with a specific main mission sub-task and thus deemed as macro-movements.

- *Mission mobility behavior*: The controllers in this layer have the largest scope and their objective is to fulfill the main mission goal, enabling the user-defined applications to be executed. Since these user applications require, in general, more complex systems, the mission behavior is in charge of creating new mobility control systems by pooling a set of dedicated subsystems.

This kind of layering fosters the creation of hierarchical control structures, a fundamental requirement for a system to have to treat high complexity systems. In [162] it was presented a similar vision for the mobility control system. However, this proposed methodology is different because:

- is presented a more fine grained classification among the mobility plane, adding a mission layer;

- the mobility control system is mapped mainly using the locality of the needed information for the control algorithm.

To have more accurate and meaningful models, the study of mobility control systems has to take communication aspect into extensive consideration. In fact, in most control problems formulations, the unmanned vehicles need to communicate within each other to implement cooperation. An underlying control system has to model diverse network characteristic to became a *communication-aware control system*. The most common approach that is present in literature, however, takes the communication parameters only to model the environment, without using them as control inputs.

5.2.4 Networked Behaviors Transversal Planes

To integrate the two different topics of control and communication, is proposed to interconnect transversely the different layers of the two stacks and generate the *networked behavioral planes*. The advantage of considering layers across different research fields in a single logical unit is twofold:

- is useful to better focus the research scope and to give a context to studies that consider at the same time communication and controlled mobility issues in a single structure;
- the systems that belong to the two different research fields are strictly correlated between each other. Hence, the creation of systems that are both communication-aware and mobility-aware, will generate a complete networked behavioural systems that will enhance the capabilities of the actual systems.

Shown in Fig. 5.5 are the three proposed locality-based behaviors: *local*, *global* and *mission* networked behaviours.

Local Networked Behavior

This layer bounds together the *micro mobility* and the set of the *physical*, *data-link* and *sub-network* layers. The main objective of this aggregation is to group behaviors that deal with local informations and use them for optimizing both the communication strategies and the micro-mobility control. All the components of this transversal layer share a locality of scope. From one side, the operating action range of the network protocols that fall under the umbrella of this behavior is just within the neighbourhood scope. On the other side, the micro mobility control systems are subject only to local information. It is, therefore, straightforward to combine these layers in a single logical plane we call local networked behavior. Possible application scenarios for these *local networked behaviors* are:

- **Connectivity Support** - One among the many problems that regard autonomous unmanned vehicles applications is the "network connectivity maintenance ". In fact, maintaining constant connectivity between vehicles is not trivial in and hence a lot of solutions have been developed. The actual definition of connectivity varies greatly but, generally, to assure it means that in every moment during the mission lifetime, there should be a communication path either between each vehicle in the network.

- **Dynamic Path Maintenance** - The topology of networks of autonomous vehicles is intrinsically unstable (see Section 3.1.1). Due to this feature, a hypothetical routing algorithm should continuously recalculate its routing tables to cope with the ever-changing environment. In this situation, to use reactive routing protocols that try to keep end-to-end paths could be an infeasible solution. An example of a system that can be described as belonging to this class of behaviors is the local redefinition of the routing tables to fast patch the continuous path disruptions while moving the autonomous vehicle in order to either create new communication links or try to keep alive some of the actual communication links.

- **Multiple and Directional Antennas** - UAVs can, among other features, move in the space with six degrees-of-freedom. This peculiarity lead to non-negligible problems for the communication layers of the lowest part of the ISO/OSI stack. In fact, the main properties of node transceiver's antennas like polarization, orientation and directionality are strictly related to the vehicles' movements and have direct impact on the communication performances.

Global networked behavior

Covering the *macro-mobility* and both the *sup-network* and *transport* layers, this networked behavior is functional in creating reliable and adaptable systems. The informations conveyed in this level have, indeed, a global scope. The networked behaviors at this layer have the knowledge of the global status of the system, regarding the specific behavior needs, and hence can build a system with a specific desired service quality by including both movement controls and communication optimizations in the protocol design.

At this layer, the local informations are no more sufficient and, hence, a cooperation system or a centralized one, is necessary in order to create *global networked behaviours*.
Example applications that can be mapped to a global networked behavior are:

- **Disruption and Delay Tolerant Networks (DDTN)** - In real missions, maintaining the network connectivity among the unmanned vehicles could be a difficult or even an impossible task. The difficulty is due to events that are beyond the operational possibilities of the network: vehicles breakdowns, failures of one or more relaying nodes and unexpected drop in the communication links quality are some examples. Furthermore, in some kind of applications, full connectivity is not always necessary and hence a temporary disconnection of the whole network could be accepted. The networked behaviours must be aware of the possibility of disruptive event to happen and be resilient to this kind of events.

- **Service differentiation and QoS** - Maintaining the network connectivity while executing the mission task could be not enough for the mission goal. In some scenarios, guaranteeing a certain Quality of Service (QoS) is of prime importance, since sometimes the final mission goal is strictly connected to the quality of the offered service. In monitoring tasks or disaster recover missions, for example, is necessary to maintain a given level of end-to-end QoS for all the mission duration.

- **Geographical Routing** - The routing problem for autonomous vehicles network is a complex issue to solve. The main proposed solution is the geographical routing, where the chosen next-hop, in a multi-hop communication algorithm, is calculated depending on the actual position of the sender and the final destination. In this case, a networked behaviour will be able either to generate new multi-hop path between the sender and the final destination or to maintain actual communication paths in order to keep them active.

Mission Networked Behavior

The mission networked behavior layer contains all the user *application*-specific goals. Here (i) the communication recipients are identified, (ii) the available resources are determined and (iii) the application's QoS requirements are set. At the same time, the protocols at this layer can manage all the vehicle movements providing mission-specific position controls to the user needs. Functional examples of the mission networked behaviors can be:

- **Data streaming** - One of the most common application in UAVs mission is the data streaming, especially for video data. In fact in both surveillance, monitoring and emergency scenarios, targets video streaming is requested to accomplish the mission task and hence the Quality of Experience (QoE) of the video is very important for the final results of the entire mission.

- **Coverage** - In wireless sensor network (WSN) the main characteristic is the coverage percentage of the sensing area. The use of unmanned vehicles in these kind of application is becoming very common due to the facility of the deployment and the autonomous mobility of the sensing nodes.

- **Network restore** - In some application, like emergency in a post disaster scenarios, the unmanned vehicles have the mission to restore the communication network that is insufficient to support the rescue operations. Here, the autonomous vehicles network must be able of guaranteeing the communication quality while maximizing the network restored area.

5.2.5 Networked behaviours in emergency scenarios

To better explain the described architecture, in this Section is presented one possible application scenario that could show, in detail, how to use the locality metrics to classify research issues among communication and mobility control, and where this two planes are intrinsically connected.

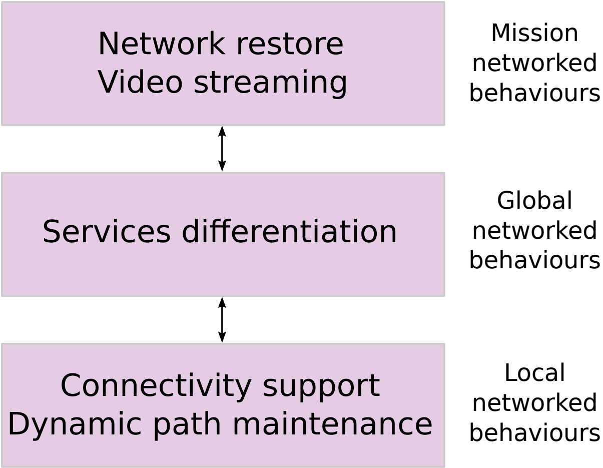


Figure 5.6: Example of the architecture: network recovery in emergency scenario

In Figure 5.6 is depicted a possible networked behavioral solution for an emergency scenario. A fleet of unmanned vehicles must explore the disaster area to give relief network services to the population, while the vehicles will also support rescue teams in their duties. If we look at the mission level, we see that the two main requests that the system must provide are network restoration and video data streaming. The former has the goal to give connectivity into the emergency area where the network infrastructure is destroyed because of the disaster. The latter helps the recovery team by transmitting real time video streaming of the scenario. We can notice that both of these functionalities have both communication and mobility control issues. In fact, the network restore module should maintain network connectivity while covering the largest possible percentage of the disaster area. The video streaming needs altogether a continuous flow of data from different points of interest.

To support the mission requests, a "services differentiation" networked behavior is put at the global level. This module will be able to give different Quality of Service (QoS) at different data flows. The video streaming flow should have more priority with respect to asynchronous data that can be transmitted in the network during the rescue operations. For these reasons, it would be desirable to have both a positioning system able to provide the presence of a multi-hop path through multiple autonomous vehicles and a communication protocol able to provide strict QoS guarantees.

Finally, at the bottom layer we have, as local networked behaviors, connectivity support and dynamic path maintenance. Connectivity support is an essential feature the vehicles group should have to be able to provide to the upper layer the services just described above. Due to the fast evolution of the vehicles' topology in an emergency scenario, there is the need of a smart dynamic path maintenance in response to the continuous repositioning.

Chapter 6

Re-establishing network connectivity

In this Chapter will be investigated the application of *Cognitive Radio* (CR) technology to address the interoperability issues, and to re-establish the network connectivity in large-scale wireless networks whose functionalities might be compromised by the occurrence of a natural calamity. Here, we consider multiple wireless devices operating on different frequencies/technologies accesses (e.g. WiFi, WiMAX, etc) that are interconnected together through dedicated infrastructure components (e.g. bridges, routers, etc) in a pre-disaster scenario. As explained in the previous chapters, after the occurrence of a natural calamity, parts of the original infrastructure might be disrupted, and network partitions (called islands) might be formed on different frequencies/technologies. In these scenarios, the utilization of dedicated and self-configuring repairing units might dynamically replace the functionalities of a compromised node or a chain of nodes. Each repairing unit is an highly adaptive self-organizing wireless device combining the spectrum reconfigurability offered by CR technologies over *Software Defined Radio* (SDR) devices, with self-positioning and dynamic routing capabilities.

6.1 Self-organizing autonomous vehicles

In this Section, is introduced the system model and the assumptions adopted. First, is detailed the characteristics of the scenario, and of the Repairing Unit (RU) devices. Then, is formulated the problem of optimal RMN deployment, which is further analysed in Sections 6.2 and 6.3.

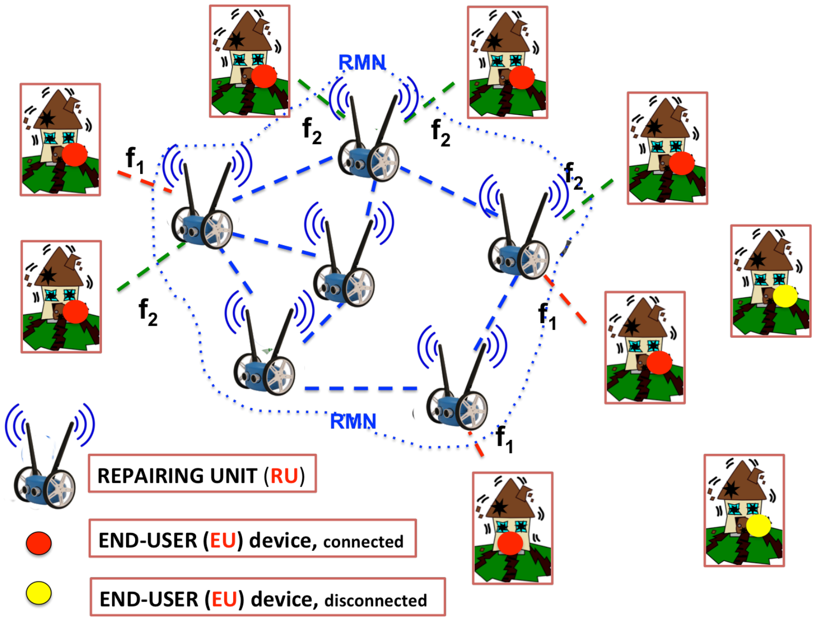

RMN
f$_1$
f$_2$
f$_2$
f$_2$
f$_2$
f$_2$
f$_1$
f$_1$
RMN
REPAIRING UNIT (RU)
END-USER (EU) device, connected
END-USER (EU) device, disconnected

Figure 6.1: Deployment of a Repairing Mesh Network (RMN) in a post-disaster scenario.

6.1.1 Cognitive Radio technology for emergency scenarios

Research on wireless systems has largely investigated the potentials of ad-hoc networks with self-organizing properties to re-establish the connectivity in a post-disaster scenario. However, while most of attention has been focused on reconfigurability issues at the network layer [128, 163] and on adaptive routing protocols [164], few works have addressed the problem of network interoperability [135], i.e. on how to guarantee seamless communication in scenarios where network fragments, public safety operators and rescue teams operate on different spectrum bands and utilize heterogeneous wireless technologies. For this purpose, technologies for dynamic spectrum reconfigurability like Cognitive Radio (CR) [3] appear as one of the best candidate solutions for network interoperability.

6.1.2 Disaster recovery scenario modeling

Without loss of generality, let us consider a post-disaster network scenario like the one showed in Figure 6.1. Before the catastrophic event, the wireless network is fully connected, and composed of N' heterogeneous end-user (EU) devices. Each EU device (i.e. u_i) is mapped to a given operating frequency f_i, denoting a specific wireless access technology. Let us assume a maximum number N_{freq} of different frequencies. The problem of network deployment that this study wants to analyse, is general and agnostic of the specific technology in use; hence, will be no further elaborated the composition and characteristics of the N_{freq} frequencies. After the occurrence of a natural calamity, the original infrastructure is disrupted, and only a fraction $N \leq N'$ of the original EU devices are still working, even though they might be disconnected one from the others. In order to favour the localization and discovery by the RUs, is assumed that each EU device u_i broadcasts periodic *HELLO* messages on frequency f_i.

6.1.3 Devices modelling

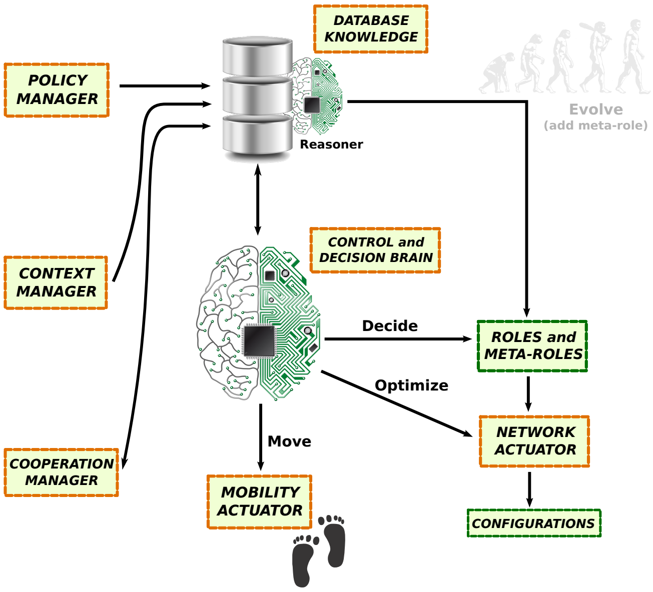


Figure 6.2: The used subset of the Stem-Node architecture

The Repairing Units (RUs) are mobile wireless devices which are injected by public safety agencies into the post-disaster scenario, with the aim of re-establishing the connectivity among isolated EU devices. Let us use the general architecture definition of the STEM-Node described in Chapter 5. In this scenario, however, it is used a subset of the whole architecture, i.e. without the *evolution* module (see Figure 6.2). Let M be the number of RUs injected in the scenario having as *Policy Manager* the task of re-establishing the EU connectivity. Each RU is provided with the *Mobility Actuator*, i.e. it can autonomously adjust its position and explore the environment. Furthermore, the *Network Actuator* of each RU is formed by $N_{sdr} + 1$ radio interfaces provided with Software Defined Radio (SDR) functionalities. The *Cooperation Manager* use only one radio to communicate with the other RUs on a Common Control Channel (CCC) (whose deployment is out of the scope of this study, interested readers can refer to [3]), while the remaining N_{sdr} radios can be used to transmit data on any of the N_{freq} frequencies, opportunely configuring the network stack based on the technology in use on frequency h, with $1 \leq h \leq N_{freq}$. Each radio with index k (1 $\leq k < N_{sdr}$) has a state variable `frequency[k]`, indicating the frequency h which it is tuned to. In case the RU is not connected to any EU device on radio k, then `frequency[k]=NULL`.

Each RUs can assume different *Roles*: i) *router* that routes the packets among the RUs mesh wireless network; ii) *gateway* that connects the isolated EU devices to the wireless mesh; iii) *scout* (described in Section 6.3) that has the task to explore the scenario. While exploring the scenario the *Context Manager* of each RUs have to be aware of the possible EUs present in the neighbourhood in order to choose the right frequency to use.

6.1.4 Problem Formulation

The objective of this Chapter is to investigate deployment strategies, so that the RMN is able to maximally connect the EU devices (shown in Figure 6.1). More formally, let us say that two EU devices u_1 and u_2 are interconnected if there exists a connection path inside the RMN from u_1 to u_2. Let U' be the set of interconnected EU devices within the scenario. The Connectivity Index (CI) is defined as follows:

$$CI = \frac{|U'|}{N} \tag{6.1}$$

The CI metric defines the ability of the RMN to re-establish the connectivity in a post-disaster scenario. In the next Sections will be addressed the problem of determining the optimal deployment of the RMN so that the value of CI is maximized. The deployment problem can be formulated in determining the optimal configuration of each RU in terms of: (*i*) placement within the scenario and (*ii*) frequency to utilize on each of the available N_{sdr} interfaces. This issue is referred as the Maximum *Multi-Channel Connectivity* (MCC) problem in the following.

6.2 Centralized Solution for the MCC Problem

In this Section, is analysed the optimal deployment of the RMN deriving an approximated (centralized) algorithm that is then compared with the distributed scheme proposed in Section 6.3. To this aim, is first formulated the MCC problem through graph theory concepts [25].

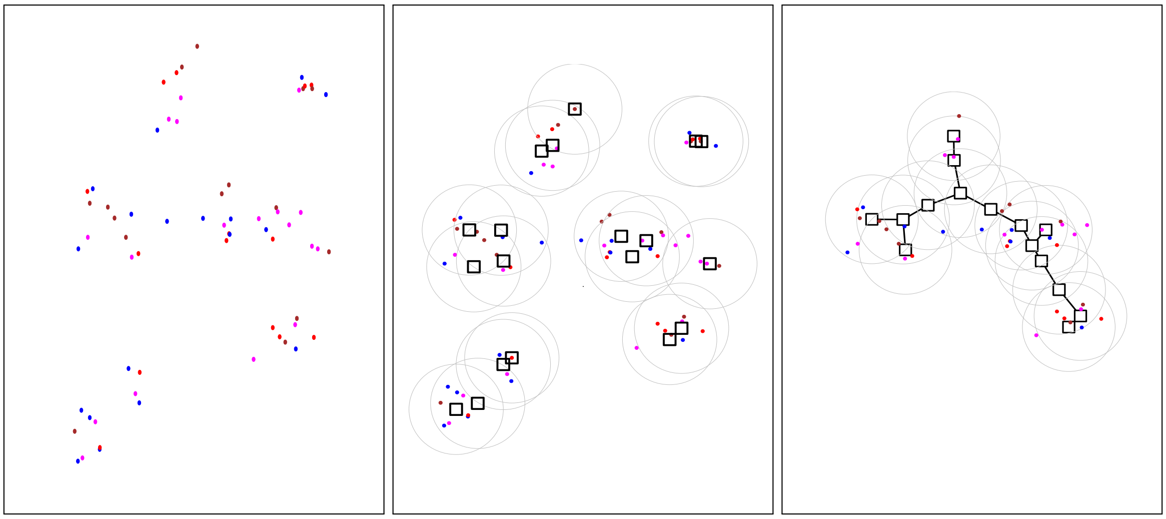
Let $U = \{u_1, u_2, ..., u_n\}$ be the set of EU devices in a metric space. Let $F = \{1, 2, ..., N_{freq}\}$ be the set of available frequencies, and $f_{EU} : U \to F$ a function mapping each u_i to the channel $f_{EU}(u_i) \in F$ in use. The RMN can be modelled as a graph $G(V, E, f_{RU}, cov_{EU})$ in a metric space, defined as follows:

- V is the set of vertexes of the graph. In this case, V represents the set of RUs used to build the RMN.
- $E = \{e(v_i, v_j)|v_i, v_j \in V\}$ is the set of edges of the graph. There exists an edge between v_i and v_j (i.e. $e(v_i, v_j)$) whether $d(v_i, v_j) \leq R$, where $d(v_i, v_j)$ is the Euclidean distance between the nodes, and R is the transmitting range of the RU, assumed equal for all. Here, it is assumed that the graph is connected, and thus a path must exist between each couple of vertexes, i.e.:
$$\forall v_i \in V, \forall v_j \in V, i \neq j, \exists e^*(v_i, v_j) \tag{6.2}$$
- $f_{RU} : V \times SDR \to F \cup \{CCC\}$, with $SDR = \{1, 2, ...N_{sdr} + 1\}$, is a function mapping each radio interface j ($1 \leq j \leq N_{sdr} + 1$) of each vertex $v_i \in V$ to a channel $f \in F \cup \{CCC\}$. Here, it is assumed that $f_{RU}(v_i, N_{sdr} + 1) = CCC, \forall v_i \in V$ and that $f_{RU}(v_i, sdr) \neq CCC, \forall v_i \in V, 1 \leq sdr \leq N_{sdr}$ i.e. each RU has only one dedicated radio tuned on the common control channel in order to communicate with the other RUs.
- $cov_{EU} : V \to U_{cov}$, with $U_{cov} \subseteq \wp(U)$, is a function mapping each vertex/RU v_i to the set of EU devices covered by it. Here, $v_i \in V$ covers $u_j \in U$ whether $d(u_j, v_i) \leq R$, and $\exists k$, $1 \leq k \leq N_{sdr}$ such that $f_{EU}(u_j) = f_{RU}(v_i, k)$, i.e. one of the radios of RU v_i is tuned to the same channel used by the EU device u_j.

The *cost* of graph G, i.e $C(G)$, is defined as the number of vertexes (RUs) composing it, i.e. $C(G) = |V|$. Similarly, the *weight* of graph G, i.e. $W(G)$, is the total number of EU devices connected by the graph. i.e. $W(G)$=$|\bigcup_{v_i \in V} cov_{EU}(v_i)|$. The MCC problem can be formulated as follows:

Definition 1 (MCC Problem)**.** *Given U, F, f_{EU} and M (number of available RUs), we want to determine a graph $G(V, E, f_{RU}, cov_{EU})$ with $C(G) \leq M$ such that the number of EU devices connected is maximized, i.e. $argmax_G(W(G))$.*

It is easy to notice that maximizing the $W(G)$ value is equivalent to maximizing the CI metric of Equation 6.1.
The solution of the MCC problem is NP-hard [25], since it is a generalization of the connectivity problem addressed in [24]. To tackle the complexity, let us follow the two-step approach described in [165], i.e.: let's decompose the MCC



(a) (b) (c)

Figure 6.3: The post-disaster scenarios with EU devices on different channels (represented by color) is shown in Figure 6.3(a). A covering structure produced by the MC-GDC algorithm is shown in Figure 6.3(b). The solution provided by the MVC algorithm over the covering structure previously computed is shown in Figure 6.3(c).

problem into two sub-problems, and then let's provide an upper bound to the cost of the combined solution, compared to the optimal case. First, it is addressed the *coverage* issue, namely how to determine the minimum set of RUs (called *covering structure* in the following) so that each EU device is covered by at least one RU. This problem is called *Multi Channel Geometric Disk Covering* (MC-GDC), and it is a generalization of the Geometric Disk Cover (GDC) problem [166]. In this case in fact the nodes to cover can use different communication channels. Then, it is addressd the *connectivity* issue, i.e. how to connect the covering structure, so that the resulting graph maximizes the number of EU devices and with a constrained number of RUs. This problem is called Maximum Vertex Clustering (MVC) [24]. The two problems are defined in a formal way as follows:

Definition 2 (Covering structure)**.** *Given U, F and f_{EU}, a covering structure is a tuple κ=$< V, f_{RU}, cov_{EU} >$, where V is a set of RUs in the metric space, f_{RU} and cov_{EU} are the functions previously introduced, and $\bigcup_{v_i \in V} cov_{EU}(v_i)$=$U$, i.e. each EU device of the set U is covered by at least one $v \in V$. In the following the use the following notations: $\kappa.V$, $\kappa.f_{RU}$, $\kappa.cov_{EU}$ to indicate the components of the covering structure κ.*

Definition 3 (MC-GDC Problem)**.** *Given U, F and f_{EU}, determine the optimal covering structure minimizing $|\kappa.V|$.*

Definition 4 (MVC Problem). *Is given κ=< V, f_{RU}, cov_{EU} > as the optimal covering structure. Determine the optimal tree $T(V' \cup S, D)$ in the metric space, with $V' \subseteq V$, S being an additional set of RUs, and $|V'| + |S| \leq M$, so that $W(T)$=$|\bigcup_{v_i \in V'} cov_{EU}(v_i)|$ is maximized.*

Definition 5 (MCC Problem revised). *Let κ=< V, f_{RU}, cov_{EU} > be an approximated solution of the MC-GDC problem, and $T(V' \cup S, D)$ an approximated solution of the MVC algorithm, the graph $G(V' \cup S, D, E, f_{RU}, cov_{EU})$ is an approximated solution of the MCC problem (bounds provided below).*

In Figure 6.3(a), is shown an example of a set of EU devices in a post-disaster scenario. A coverage structure over the same scenario is shown in Figure 6.3(b). A possible solution to the MVC problem is drawn in Figure 6.3(c).
Now will be derived the approximation bounds in terms of cost (i.e. number of RUs) introduced by the optimal MCC solution, and by the two-step solution, when connecting the same subset of EU devices. Approximation bounds in terms of weight (i.e. number of EU devices connected) are provided in Section 6.2.3.
Let $G^{opt}(V^{opt}, E^{opt}, f^{opt}_{RU}, cov^{opt}_{EU})$ be the optimal MCC solution. Let $G^{apx}(V^{apx}$, $E^{apx}, f^{apx}_{RU}, cov^{apx}_{EU})$ be the solution of the two-step approach, connecting the same set of EU devices than G^{opt} (i.e. $\bigcup_{v_i \in V^{opt}} cov^{opt}_{EU}(v_i) = \bigcup_{v_i \in V^{apx}} cov^{apx}_{EU}(v_i)$), where $C(G^{apx}) \leq M$. For easy of disposition, a graph is indicated by omitting its components. Through Theorems 1 and 2, we derive the upper bound $\gamma \geq 1$ on the cost of approximated solution compared to the optimal one:

$$C(G^{apx}) \leq \gamma \cdot C(G^{opt}) \tag{6.3}$$

Theorem 1. *If κ is an α-approximated solution for the MC-GDC problem, and T is a β-approximated solution for the MVC problem, then γ=$\alpha + \alpha \cdot \beta + \beta$.*

Proof. Let RU^{apx} the set of RUs used by G^{apx}. Let us split RU^{apx} into two subsets, i.e. RU^{apx}_{cover} and RU^{apx}_{relay}. The first subset (RU^{apx}_{cover}) is composed of RUs used for *coverage*, i.e. each RU connects to at least one EU device, but RUs might not form a connected graph. Formally, we have that $\forall v \in RU^{apx}_{cover}$, $cov_{EU}(v) \neq \emptyset$. The second subset ($RU^{apx}_{relay}$) is composed by RUs used for guaranteeing *end-to-end connectivity* among RU^{apx}_{cover}, without covering any EU device, i.e. $\forall v \in RU^{apx}_{relay}$, $cov_{EU}(v) = \emptyset$. Clearly, $|RU^{apx}|$=$|RU^{apx}_{cover}|$+$|RU^{apx}_{relay}|$. Similarly, let RU^{opt} be the set of RUs used by G^{opt}, i.e. the solution found by the optimal algorithm and covering the same set of EU devices than G^{apx}, but with minimum number of RUs. Again, we consider two subsets of RU^{opt}, i.e. RU^{opt}_{cover} and RU^{opt}_{relay}, with $|RU^{opt}$=$|RU^{opt}_{cover}|$+$|RU^{opt}_{relay}|$. We can say that:

$$\begin{aligned} C(G^{apx}) &= |RU^{apx}_{cover}| + |RU^{apx}_{relay}| && (6.4) \\ &\leq \alpha \cdot |RU^{opt}_{cover}| + \beta \cdot |RU^{opt}_{relay(RU^{apx}_{cover})}| && (6.5) \end{aligned}$$

where $RU^{opt}_{relay(RU^{apx}_{cover})}$ is the set of RUs calculated by the optimal MVC to connect nodes belonging to RU^{apx}_{cover}. We observe that the following inequality holds:

$$|RU^{opt}_{relay(RU^{apx}_{cover})}| \leq |RU^{opt}| + |RU^{apx}_{cover}| \tag{6.6}$$

In other words, the entire set of RUs can be used to found by the optimal MCC solution, plus a number of RUs equal to $|RU^{apx}_{cover}|$, to connect all the $|RU^{apx}_{cover}|$ RUs returned by the approximated coverage algorithm. This holds because: (*i*) RU^{opt} is a connected mesh covering the same set of EU devices than RU^{apx} (by construction), (*ii*) for each EU device u_k, the distance between node $v^{apx}_i \in RU^{apx}_{cover}$ and $v^{opt}_j \in RU^{opt}_{cover}$ covering u_k (i.e. $u_k \in cov_{EU}(v^{apx}_i)$ and $u_k \in cov_{EU}(v^{opt}_j)$) can be larger than R, but can not exceed $2 \cdot R$, (*iii*) at most other $|RU^{apx}_{cover}|$ nodes must be used to connect each RU of RU^{apx}_{cover} to the G^{opt} graph. Combining Equations 6.4 and 6.6 we get:

$$\begin{aligned}
C(G^{apx}) &\leq \alpha \cdot |RU^{opt}_{cover}| + \beta \cdot (|RU^{opt}| + |RU^{apx}_{cover}|) && (6.7)\\
&\leq \alpha \cdot |RU^{opt}_{cover}| + \beta \cdot (|RU^{opt}| + \alpha \cdot |RU^{opt}_{cover}|) && (6.8)\\
&= \alpha \cdot |RU^{opt}_{cover}| + \alpha \cdot \beta \cdot |RU^{opt}_{cover}| + \beta \cdot |RU^{opt}| && (6.9)\\
&\leq \alpha \cdot |RU^{opt}| + \alpha \cdot \beta \cdot |RU^{opt}| + \beta \cdot |RU^{opt}| && (6.10)\\
&= (\alpha + \alpha \cdot \beta + \beta) \cdot |RU^{opt}| && (6.11)\\
&= (\alpha + \alpha \cdot \beta + \beta) \cdot C(G^{opt}) && (6.12)
\end{aligned}$$

□

Theorem 2 (Corollary). *Let κ be an α-approximated solution for the MC-GDC problem. Let T be a β-approximated solution for the MVC problem. If $\forall v \in RU^{opt}_{cover}$ and $\forall u \in cov_{EU}(v)$, $d(v,u) = r \leq \frac{R}{2}$, then γ=$\alpha + \beta$.*

Proof 1. *Let us use the same structure of proof 6.2. However, let us refine the Equation 6.6, since now we need only $|RU^{opt}|$ additional RUs to connect the nodes belonging to RU^{apx}_{cover}. Indeed, let $u_k \in U$ be an EU device. Let $v^{apx}_i \in RU^{apx}_{cover}$ the RU device covering u_k in the approximated solution (i.e. $u_k \in cov_{EU}(v^{apx}_i)$), and v^{opt}_j the analogous in the optimal solution (i.e. $u_k \in cov_{EU}(v^{opt}_j)$). Since $d(v^{apx}_i, u_k) \leq \frac{R}{2}$ and $d(v^{opt}_j, u_k) \leq \frac{R}{2}$, this implies that $d(v^{apx}_i, v^{opt}_j) \leq R$. In other words, the G^{opt} graph can be also used to connect the nodes of RU^{apx}_{cover}. Therefore:*

$$
\begin{aligned}
C(G^{apx}) &= |RU^{apx}_{cover}| + |RU^{apx}_{relay}| && (6.13)\\
&\leq \alpha \cdot |RU^{opt}_{cover}| + \beta \cdot |RU^{opt}_{relay(RU^{apx}_{cover})}| && (6.14)\\
&\leq \alpha \cdot |RU^{opt}_{cover}| + \beta \cdot |RU^{opt}| && (6.15)\\
&\leq \alpha \cdot |RU^{opt}| + \beta \cdot |RU^{opt}| && (6.16)\\
&= (\alpha + \beta) \cdot |RU^{opt}| && (6.17)\\
&= (\alpha + \beta) \cdot C(G^{opt}) && (6.18)
\end{aligned}
$$

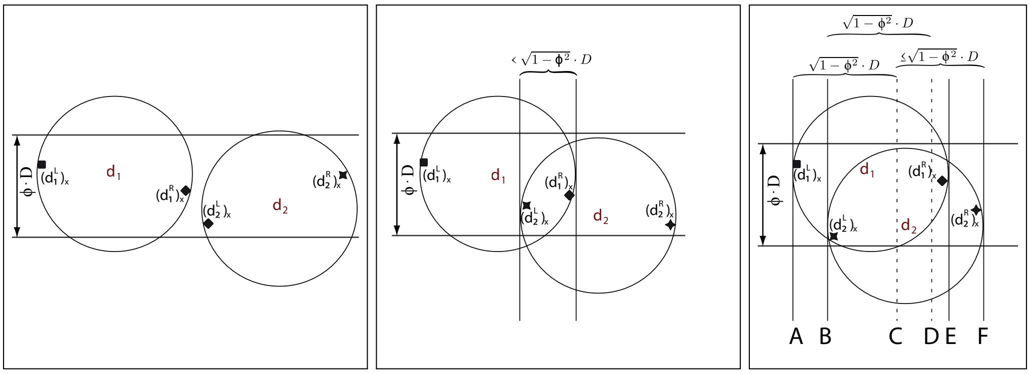
6.2.1 The MC-GDC Problem

The *Multi-Channel Geometric Disk Cover* (MC-GDC) problem [25] is a generalization of the Geometric Disk Cover (GDC) problem [166] for multi-channel environments, i.e. where the nodes to cover are mapped on different communication channels. In [166] the authors propose to split the area into strips of equal height; coverage is done locally on each strip, and then the global solution is obtained by combining the local solutions. In [165], a 4.5-approximation algorithm is described. Here, is proposed to extend the algorithm given in [165] for the multi-channel scenario, obtaining a 6-approximated solution.

Algorithm 1 takes in input the set of EU devices U to cover, the channels list F, and the channel mapping function f_{EU} and returns the covering structure κ. The algorithm works as follow: first the area is divided into K strips of height $q = \phi \cdot D$ where $D = 2 \cdot R$, and $\phi \leq 1$. Then, for each strip, the leftmost uncovered EU device is considered, and the function $Disk$ is invoked till all the EU devices of the current strip are covered. The set S_k represents the covering RUs allocated for the k^{th} strip. The vertexes of the covering structure (i.e. $\kappa.V$) are obtained by performing the union of the S_k sets, over all strips k.

Algorithm 2 describes the operations of the $Disk$ function. It takes as input an an EU device u_j on strip K and channel c, and places a RU v_i to cover it, by also updating the $f_{RU}(v_i)$ and $cov_{EU}(v_i)$ sets. Then (line 10) the algorithm considers the leftmost uncovered EU devices next to u_j (e.g. u_l), and checks whether u_l can also be covered by RU v_i. The loop continues till (*i*) all EU devices of strip k are covered by v_i, or (*ii*) condition at line 4 becomes false, i.e. v_i cannot cover other EU devices. Then, RU v_i is returned by the $Disk$ function, and added to the current set S_k (at line 7 of Algorithm 1).

Is now provided the approximation ratio for the MC-GDC algorithm. Let us assume that $N_{sdr} < N_{freq}$, i.e. the number of radios used by the RUs is less than available communication channels used by EU devices. In the case of $N_{sdr} \geq N_{freq}$ the problem become a single channel GDC problem. The approximation



(a) (b) (c)

Figure 6.4: Base cases for the proof of Theorem 3. Base case 1 with $(d_1^R)_x < (d_2^L)_x$ is shown in Figure 6.4(a). Base case 2 with $(d_1^R)_x > (d_2^L)_x$ and $(d_1^R)_x - (d_2^L)_x \leq \sqrt{1-\phi^2} \cdot D$ is shown in Figure 6.4(b). Base case 3 with $(d_1^R)_x > (d_2^L)_x$ and $(d_1^R)_x - (d_2^L)_x \leq \sqrt{1-\phi^2} \cdot D$ is shown in Figure 6.4(c).

ratio α is defined in terms of number of RU devices to cover all the $|U|$ EU devices of the scenario, i.e.:

$$\alpha = \frac{C_{APPROX}}{C_{OPT}} \tag{6.19}$$

where C_{OPT} is the cost of the optimal solution of the MC-GDC problem, and C_{APPROX} the cost of Algorithm 1.
To derive the bound α, let us use the following result described in [166]. Let α_{STRIP} be the approximation ratio on a *single strip*. All strips are assumed to be of equal height q. Then, the following inequation holds between α (i.e. the approximation ratio over all the strips of the scenario) and α_{STRIP}:

$$\alpha \leq \left(\left\lceil \frac{D}{q} \right\rceil + 1 \right) \cdot \alpha_{STRIP} \tag{6.20}$$

Based on Equation 6.20 we derive α_{STRIP} and α for the MC-GDC algorithm through the following theorems.

Theorem 3. *If $\phi \leq \frac{\sqrt{5}}{3}$, then $\alpha_{STRIP} \leq 2$, i.e. Algorithm 1 uses at most twice the RUs than the optimal MC-GDC solution to cover a single strip.*

Proof. Let us consider a strip, i.e. strip *k*. By hypothesis, each strip heigth is equal to *q*=$\phi \cdot D$. For easy of disposition of the proof, let us assume that $N_{freq} \cdot \frac{1}{2} < N_{sdr} < N_{freq}$, i.e. the number of radio used by covering nodes is lower than the number of frequencies used by EU devices, but still greater than the half of this

Table 6.1: Channel/area coverage table: covering nodes are on the rows, areas are on the columns. Each cell $< i, j >$ indicates the number of channels still un-covered on area segment j after step i of the approximated algorithm has been executed.

	AB	BC	CD	DE	EF
begin	$f_{RU}^{opt}(d_1)$	$f_{RU}^{opt}(d_1)$ $f_{RU}^{opt}(d_2)$	$f_{RU}^{opt}(d_1)$ $f_{RU}^{opt}(d_2)$	$f_{RU}^{opt}(d_1)$ $f_{RU}^{opt}(d_2)$	$f_{RU}^{opt}(d_2)$
a_1	X	$f_{RU}^{opt}(d_1) - f_{RU}^{apx}(a_1)$ $f_{RU}^{opt}(d_2) - f_{RU}^{apx}(a_1)$	$f_{RU}^{opt}(d_1)$ $f_{RU}^{opt}(d_2)$	$f_{RU}^{opt}(d_1)$ $f_{RU}^{opt}(d_2)$	$f_{RU}^{opt}(d_2)$
a_2	-	X	$f_{RU}^{opt}(d_1) - f_{RU}^{apx}(a_1)$ $f_{RU}^{opt}(d_2) - f_{RU}^{apx}(a_2)$	$f_{RU}^{opt}(d_1)$ $f_{RU}^{opt}(d_2)$	$f_{RU}^{opt}(d_2)$
a_3	-	-	X	$f_{RU}^{opt}(d_1) - f_{RU}^{apx}(a_3)$ $f_{RU}^{opt}(d_2) - f_{RU}^{apx}(a_3)$	$f_{RU}^{opt}(d_2) - f_{RU}^{apx}(a_3)$
a_4	-	-	-	X	X

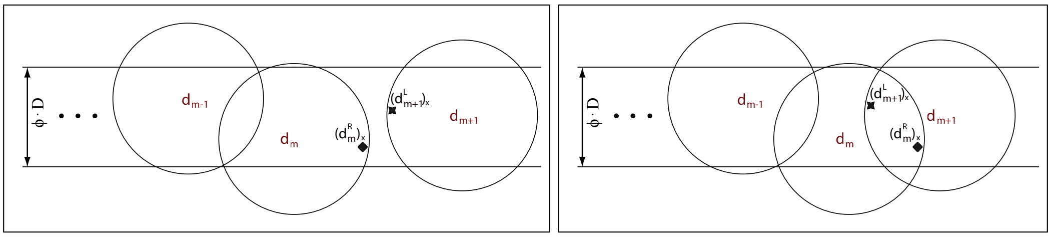
value.
Let $S_k^{opt} = \{d_1, d_2, \ldots, d_{|S_k^{opt}|}\}$ be the optimal set of covering nodes on strip *k*, and $f_{RU}^{opt}(d_i)$ be the function mapping each covering node $d_i \in S_k^{opt}$ to the list of channels in use. Similarly, let $S_k^{apx} = \{a_1, a_2, \ldots, a_{|S_k^{apx}|}\}$ be the approximated local solution produced by Algorithm 1 and $f_{RU}^{apx}(a_i)$ be the function mapping each covering node $a_i \in S_k^{apx}$ to the list of channels in use. We assume that the sets S_k^{opt} and S_k^{apx} are ordered based on the *x* coordinate of the nodes, namely $d_i \leq d_j$ if $(d_i^L)_x \leq (d_j^L)_x$ and $a_i \leq a_j$ if $(a_i^L)_x \leq (a_j^L)_x$, where $(d_i^L)_x$ and $(d_i^R)_x$ denote respectively the *x* coordinate of the leftmost and the rightmost EU covered by node *i*.

Following the schema of the proof in [165], and assuming that $\phi < 1$, we prove by induction that if $\phi \leq \frac{\sqrt{5}}{3}$ the approximation ratio is equal to 2, i.e. $|S_k^{apx}| = b_{|S_k^{opt}|} \leq 2 \cdot |S_k^{opt}|$.

Base case: There are three subcases to consider.

- The first case occurs when $(d_1^R)_x < (d_2^L)_x$, as shown in Figure 6.4(a). It's easy to see that a_1 coincides with d_1, with $f_{RU}^{opt}(d_1) = f_{RU}^{apx}(a_1)$, because all the EUs covered by the optimal disk d_1 can be covered by a single node placed according to rules of Algorithm 1. Therefore, $b_1 = 1$.

- The second case occurs when $(d_1^R)_x > (d_2^L)_x$ and $(d_1^R)_x - (d_2^L)_x \leq \sqrt{1-\phi^2} \cdot D$, as shown in Figure 6.4(b). In the following, it will be proved that $b_2 \leq 4$, i.e. in order to produce the same coverage of 2 nodes of the optimal solution we need at most 4 nodes of the approximated solution (in the worse case). Here, we are considering the worst case in terms of channel allocation from the EU devices, with $f_{RU}^{opt}(d_1) \cap f_{RU}^{opt}(d_2) = \emptyset$ and $f_{RU}^{opt}(d_1) \neq f_{RU}^{apx}(a_1)$ and



(a) (b)

Figure 6.5: Inductive cases for the proof of Theorem 3. Subcase 1 with $(d_m^R)_x < (d_{m+1}^L)_x$ is shown in Figure 6.5(a). Subcase 2 with $(d_m^R)_x > (d_{m+1}^L)_x$ is shown in Figure 6.5(b).

$f_{RU}^{opt}(d_2) \neq f_{RU}^{apx}(a_2)$. Referring to the scenario depicted in Figure 6.4(b), it can be noticed that all the EUs till $(d_2^L)_x$ can be covered by a single node a_1, with $f_{RU}^{apx}(a_1) \subseteq f_{RU}^{opt}(d_1)$. In the region between $(d_2^L)_x$ and $(d_1^R)_x$, all EU devices are mapped on channels in $f_{RU}^{opt}(d_1) \cup f_{RU}^{opt}(d_2)$. Since in a strip of height $\phi \cdot D$, the maximum segment covered (on the $x-$axis) by a single node is $\sqrt{1-\phi^2} \cdot D$, we need at most 2 nodes to cover all the EUs in the target region: one covering node (a_2) is tuned on channels given by $f_{RU}^{opt}(d_1)$, and the other (a_3) on channels given by $f_{RU}^{opt}(d_1)$. Finally, in the region on the right of $(d_1^R)_x$, all the EU devices are using channels in $f_{RU}^{opt}(d_2)$, hence a single node of the approximated solution (a_4) can cover all of them. Therefore $b_2 \leq 4$, having that $\phi \leq 1$.

- The third case occurs when $(d_1^R)_x > (d_2^L)_x$ and $(d_1^R)_x - (d_2^L)_x > \sqrt{1-\phi^2} \cdot D$, as shown in Figure 6.4(c). We distinguish six different points in the Figure, i.e. A, B, C, D, E, F. On Table 6.1 we depict the approximated covering nodes a_i (on the rows), and possible channels still un-covered on each area region; a cross on cell $<i,j>$ indicates that all channels in target area j have been covered by node a_i. As for the previous case, we consider the worst in terms of channel allocation from the EU devices, with $f_{RU}^{opt}(d_1) \neq f_{RU}^{apx}(a_1)$ and $f_{RU}^{opt}(d_2) \neq f_{RU}^{apx}(a_2)$. At each iteration of the MC-GDC algorithm, a new covering node is added, and the list of un-covered channels on each area is updated accordingly; after 4 iterations, all the EU devices of the scenario are covered, i.e. $b_2 \leq 4$. Moreover, the coverage of a_3 of the area from point C to point F is possible only if $2 \cdot \sqrt{1-\phi^2} \cdot D \geq 2 \cdot D - \sqrt{1-\phi^2} \cdot D$. Solving the inequation in terms of ϕ, we have that: $\phi \leq \sqrt{5}/3$.

Inductive case. By inductive hypothesis, let us assume that the approximated algorithm uses at most $2m$ covering nodes instead of m of the optimal solution, i.e. $b_m \leq 2m$. At the inductive step, let us suppose that one covering node is added to to the optimal solution, i.e. $|S_k^{opt}|= m+1$, and it must be proved that $|S_k^{apx}|= 2 \cdot (m+1)$. As for the base case, let us distinguish two possible subcases, based on reciprocal positions of $(d_m^R)_x$ and $(d_{m+1}^L)_x$. The first subcase is shown in Figure 6.5(a) and occurs when $(d_m^R)_x < (d_{m+1}^L)_x$. Following the same reasoning of the subcase base 1, we can see that the approximated algorithm adds only one covering node to S_k^{apx}, i.e.:

$$b_{m+1} = b_m + 1 \leq 2m + 1 \leq 2(m+1) \tag{6.21}$$

The second subcase is shown in Figure 6.5(b) and occurs when $(d_m^R)_x > (d_{m+1}^L)_x$. All EU segments in the area region till $(d_{m+1}^L)_x$ have been covered by inductive hypothesis. From $(d_{m+1}^L)_x$ to $(d_{m+1}^R)_x$, EU devices can be allocated on channels $f_{RU}^{opt}(d_m) \cup f_{RU}^{apx}(d_{m+1})$, and hence two more covering nodes of the approximated solution (a_{2m+1} and a_{2m+2}) are enough to produce the same coverage, i.e.:

$$b_{m+1} \leq b_m + 2 \leq 2m + 2 = 2(m+1) \tag{6.22}$$

□

Theorem 4 (Corollary). *If $\frac{1}{2} \leq \phi \leq \frac{\sqrt{5}}{3}$, then $\alpha \leq 6$, i.e. Algorithm 1 uses at most six time the number of RUs used by the optimal MC-GDC solution to cover the full scenario.*

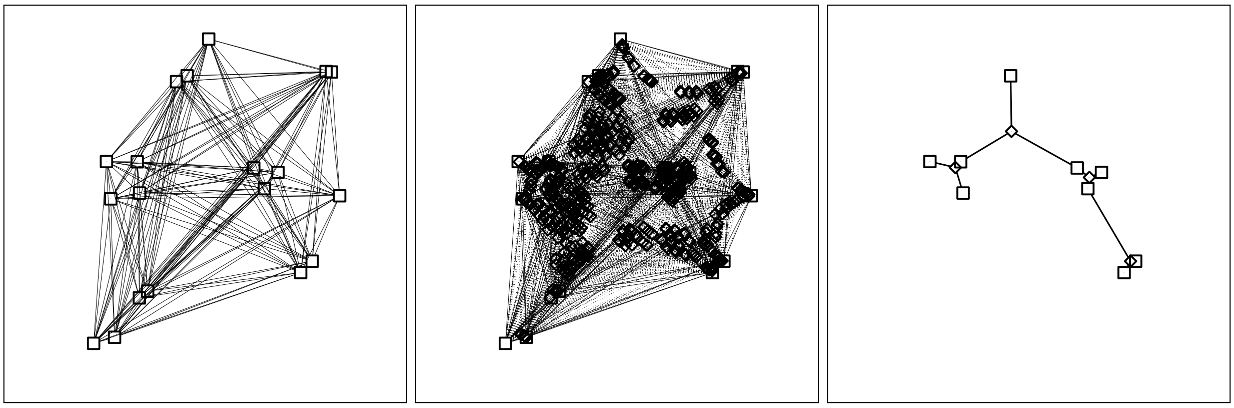
Proof 2. *Using Equation 6.20 and Theorem 3, we get $\alpha \leq \left(\left\lceil \frac{D}{q} \right\rceil + 1\right) \cdot 2$. Since $D = 2 \cdot R$, $q = 2 \cdot R \cdot \phi$, and $\phi \leq \frac{1}{2}$ by assumption, then: $\alpha \leq (2+1) \cdot 2 \leq (2+1) \cdot 2 \leq 6$.*

Algorithm 1: The MC-GDC Algorithm

```
Require: $U, F, f_{EU}, D = 2 \cdot R, \phi$
Divide the area into $K$ strips of height $q = \phi \cdot D$
$S_k \leftarrow \emptyset, \forall k = 1, \ldots, K$
for all strips $S_k, k = 1, \ldots, K$ do
    while there exist uncovered EU devices in $S_k$ do
        let $u_j \in U$ be the leftmost uncovered node in $S_k$ using channel $c = f_{EU}(j) \in F$
        $v_i \leftarrow$ call Disk($j$, $c$)
        $S_k \leftarrow S_k \cup \{v_i\}$
    end while
end for
$\kappa.V = \bigcup_k S_k$
return $\kappa =< V, f_{RU}(V), cov_{EU}(V) >$
```

Algorithm 2: The Disk(*j*, *c*) function

```
Require: $j, c$
$cov_{EU}(v_i) \leftarrow \emptyset$ {set of EU devices covered by $v_i$}
$f_{RU}(v_i, h) \leftarrow NULL\ \forall h \in 1 \ldots N_{sdr}$ {set of channels used by $v_i$}
loop
    if ($cov_{EU}(v_i) \cup \{u_j\}$ is coverable by a single disk) and ($\exists h \in 1 \ldots N_{sdr}$ such that ($f_{RU}(v_i, h) = c$ or
    $f_{RU}(v_i, h) = NULL$)) then
        $f_{RU}(v_i, h) \leftarrow c$
        $cov_{EU}(v_i) \leftarrow cov_{EU}(v_i) \cup \{u_j\}$
        Mark $u_j$ as covered
        Let $u_l$ be the leftmost uncovered node next to $u_j$
        if ($|x_{u_l} - x_{u_j}|) \leq D$ then
            Set $j \leftarrow l, c \leftarrow f_{EU}(u_l)$
        else
            break loop
        end if
    else
        break loop
    end if
end loop
Place $v_i$ such that it covers all the EU devices in $cov_{EU}(v_i)$
return $v_i$
```



(a) (b) (c)

Figure 6.6: The graph taken as input by the MVC problem is shown in Figure 6.6(a). The graph G' derived from G after Step 1 of Algorithm 3 is shown in Figure 6.6(b). The tree returned as output after Step 2 of Algorithm 3 is shown in Figure 6.6(c).

6.2.2 The MVC Problem

Through Step 1, an approximated covering structure $\kappa =< V, f_{RU}, cov_{EU} >$ has been determined where each EU $u \in U$ has been covered by a RU $v \in V$, and $|V|$ is minimized. However, the covering structure is not yet a solution to the MCC problem, since the corresponding RMN might not be connected. Hence, in this Step 2, it is proposed an algorithm to connect V (or a subset) through a

set of additional RUs (S), so that: (*i*) the number of RUs used for both covering/connecting is less than or equal to M (i.e. of the number of available RUs), and (*ii*) the number of EU devices interconnected by the RMN is maximized.
The Definition 4 of the MVC problem is extended by using the formulation proposed in [24]. Later in this Chapter, will be proved that two definitions are equivalent, and that the tree of Definition 4 can be derived by the solution of the general MVC problem described below.

Let $\kappa(U, f_{EU}) =< V, f_{RU}, cov_{EU} >$ be a covering structure. Let us construct a graph $G(V, E, c, w)$ defined as follows:

- V is the set of vertexes, and it is equal to $\kappa.V$ (i.e. of the RUs used by the covering structure κ).

- E is the set of edges: $e(v_i, v_j)$, with $v_i \in V$, $v_j \in V$. The graph G is fully connected, i.e. there exists an edge $e(v_i, v_j) \in E, \forall\, v_i \in V$, $v_j \in V$, $i \neq j$.

- $w : V \rightarrow \mathbb{N}$ is a function assigning a weight to each vertex. In this case, $w(v)$=$|\kappa.cov_{EU}(v)|$, i.e. the weight of vertex v is represented by the number of EU devices covered by it, according to the covering structure κ computed at Step1.

- A cost is assigned to each edge of the graph: $c : E \rightarrow \mathbb{N}$. In our case, $c(e(v_i, v_j))$ is the average number of RUs needed to connect v_i and v_j. Let $d(v_i, v_j)$ be the Euclidean distance between v_i and v_j, and R the transmitting range of a RU, then $c(e(v_i, v_j))$ can be approximated as follows:

$$c(e(v_i, v_j)) = \left\lfloor \frac{d(v_i, v_j)}{R} \right\rfloor \tag{6.23}$$

Figure 6.6(a) shows the graph G built over the covering structure depicted in Figure 6.3(b).

Definition 6 (MVC Problem revised)**.** *Given $G(V, E, w, l)$, let us determine the optimal tree $T_{MVC}(P, D)$, where P=$T \cup S$, with the terminal points $T \subseteq V$ and S a set of new points (with $w(s) = 0\; \forall\; s \in S$), such that the tree weight $W(T_{MVC} = \sum\limits_{v \in P} w(v))$ is maximized (i.e. the maximum number of EU devices are connected by the RUs), while the tree cost $C(T_{MVC}) = \sum\limits_{e \in E} c(e(v_i, v_j)) + |P| \leq M$ (i.e. the number of RUs is constrained by M).*

The MVC is similar to the well-known problem of computing the *Steiner Minimum Tree* (SMT) on a weighted graph in metric space. Given a graph G_s=$\{V_s, E_s\}$, the $SMT(G_s)$ is defined as the tree $T_s(P_s, D_s)$, with $V_s \subseteq P_s$, such that all vertexes are connected and the cost $C(T_{MVC})$ is minimized [167, 168, 169, 170]. Differently from the classical minimum spanning tree problem, the SMT includes both vertexes of the graph (called *terminal points*) and new vertexes p (with $p \in P_s$ and $p \notin V_s$), called *Steiner points*. In this case study, the tree $T_{MVC}(P, D)$ solution of the MVC problem is a Steiner tree but with two additional constraints: (*i*) the number of vertexes cannot exceed M (number of available RUs), and (*ii*) at each Steiner tree is associated a weight, representing the total number of EU devices connected.

Unfortunately, the computation of the SMT is known to be an NP-hard problem. For this reason, is proposed an approximated solution and is derived the upper bounds to the optimum. The approximation leverages the notion of *full components* (or, for short, components) of a SMT, defined as a maximal subtree whose terminals coincide with its leave [169, 170]. A k-restricted SMT is defined as a Steiner tree whose components contain no more than k terminals (k-component). For $k \geq 4$, computing the *k-SMT(G)* is a NP-hard problem too. However, for $k = 3$, the problem admits polynomial solution through the Melzak algorithm [168], and it is possible to prove that the approximation ratio ρ provided by the *k-SMT* solution is at most equal to $\frac{5}{3}$ of the optimum [167], i.e.:

$$\rho = \frac{C(k\text{-SMT(G)})}{C(\text{SMT(G)})} \leq \frac{5}{3} \tag{6.24}$$

Algorithm 3: The `3-MVC` algorithm

```
Given a graph $G$={$V, E, l, w, c$} and $M$:
Initialize: $V'$=$V$ and $E'$=$E$.
**Step1**: Build $G'(V', E')$:
**for** each $(v_i, v_j, v_k)$ triples, with $v_i \in V$, $v_j \in V$, $v_k \in V$ **do**
    Compute Steiner point $z$ through the Melzak algorithm [168].
    Set $V'$=$V' \cup \{z\}$.
    Set $E'$=$E' \cup \{e(z, v_i)\} \cup \{e(z, v_j)\} \cup \{e(z, v_k)\}$.
**end for**
**Step2**: Compute the approximated MVC:
  $T_{3-MVC}$= MVCSearch(G',M).
  Return $T_{3-MVC}$.
```

Based on this result, it is propose the `3-MVC` algorithm [25], which works in two steps. First, the graph G is modified in order to include all possible *3-SMT* structures. Then a solution is searched among the structures in order to compute the *3-SMT* with maximum weight.

The first Step (described in Algorithm 3) handle the original graph G by: (*i*) computing the Steiner point z for each triple of vertexes $v_i \in V$, $v_j \in V$, $v_k \in V$ of G, (*ii*) adding z to set of vertexes. Here, we assume that $z \neq v_i$, $z \neq v_j$ and $z \neq v_k$, otherwise no adding is done to V' and E'. Then (*iii*) adding the corresponding edges $u(v_i, z)$, $u(v_j, z)$, and $u(v_k, z)$ to the set of edges. As a result, a new graph $G'(V', E')$ is built at the end of Step 1, as shown in Figure 6.6(b). It is easy to notice that the resulting graph G' include all possible *3-SMT* structures, since by definition it calculates a Steiner point for each possible connected components having three vertexes of V as terminals.

The *3-SMT* with maximum weight is computed on G' through the `MVCSearch` algorithm (pseudo-code in Algorithm 4). Here, a T_{MVC} global structure is used to keep track of the best solution found so far, and is returned at the end of the algorithm (line 6). For each vertex belonging to G, the `MVCSearch` invokes the `FindBestTree` procedure, which explores all the possible subtrees T_v rooted at v and saves the one with maximum weight into T_{MVC}. Each iteration of `FindBestTree` keeps track of the current vertex v_{cur} visited by the algorithm, of the previous node v_{prev}, of the*frontier* F_r, defined as the set of vertexes not yet visited by the algorithm, and of the current tree T_v. At each iteration, the frontier F_r is increased by including the neighbors of v (denoted as $Neigh_v$ at line 13), the cost $C(T_v)$ is decreased of $c(u(v_{prev}, v_{cur}))$ (i.e. of the number of RUs needed to connect the two vertexes) at line 15, and then the algorithm is recursively invoked on the new frontier and on the graph $G \setminus \{v_{cur}\}$ at line 16. When $C(T_v) > M$, i.e. no more nodes can be connected, the optimality the tree is checked at line 22 by comparing its weight $W(T_v)$ with the best value found so far (i.e. $W(T_{MVC})$). In case $W(T_{MVC})$ is equal to $W(T_v)$, the solution with lower cost (i.e. requiring less RUs) is preferred.

Figure 6.6(c) shows the tree structure built by the algorithm.

In the following, it is derived the cost approximation ratio of T_{MVC}, i.e. the β factor introduced in Theorem 1.

Theorem 5. *Given a graph G=$\{V, E, w, c\}$, let $T_{3-MVC}(P_{3m}, E_{3m})$ be the tree computed by the 3-MVC algorithm, with terminal nodes $V_{3m} \subseteq P_{3m}$. Let SMT be the minimum Steiner Tree connecting all nodes $v \in V_{3m}$. Then, β=$\frac{C(T_{3-MVC})}{C(SMT)} \leq \frac{5}{3}$, i.e. Algorithm 3 uses at most $\frac{5}{3}$ time the number of RUs used by the optimal solution to connect the same subset of EU devices.*

Proof. Let $T^*_{3-MVC}(P^*_{3m}, E^*_{3m})$ be the optimal *3-SMT* with terminal points $V^*_{3m} \subseteq P^*_{3m}$ and $V^*_{3m} = V_{SMT}$ connecting the same subset of vertex $v \in V$. From Equation 6.24 we know that $\frac{C(T^*_{3-MVC})}{C(SMT)} \leq \frac{5}{3}$. The graph $G' = \{V', E', w', c'\}$ produced at Step1 of Algorithm 3 contains, by construction, all possible 3-components and hence also T^*_{3-MVC}, i.e. $P^*_{3m} \subseteq V'$ and $E^*_{3m} \subseteq E'$. Similarly, at Step2, Al-

Algorithm 4: The `MVCSearch` algorithm

```
Initialize: $T_{MVC}$={}.
for $v \in V$ do
    Initialize $T_v$={}.
    Invoke FindBestTree ({$v$},$G$,$M$,$T_v$,NULL).
end for
Return the tree $T_{MVC}$.

procedure FindBestTree ($Fr$,$G$,$b$,$T_v$,$v_{prev}$)
Initialize $checkTree$=true.
for $v_{cur} \in Fr$ do
    if $(c(u(v_{cur}, v_{prev}) + 1) \leq b)$ and $(u(v_{cur}, v_{prev})) \in E'$ then
        Set $checkTree$ = false.
        Update the frontier $F_r$=$(F_r \cup Neigh_v) \setminus \{v_{cur}\}$.
        Add vertex $v$ and edge $u(v_{cur}, v_{prev})$ to $T_v$.
        Update current cost $b'$=$b - c(u(v_{cur}, v_{prev})) - 1$.
        FindBestTree ($F_r$,$G \setminus \{v_{cur}\}$,$b'$,$T_v$, $v_{cur}$).
        Remove vertex $v$ and edge $u(v_{cur}, v_{prev})$ to $T_v$.
    end if
end for
if $checkTree ==$ true then
    Add $|T_v|$ to $C(T_{MVC})$
    if $(W(T_{MVC}) < W(T_v))$ or $((W(T_{MVC}) == W(T_v))$ and $(C(T_{MVC}) > C(T_v)))$ then
        Set the new solution $T_{MVC} = T_v$.
    end if
end if
```

gorithm 3 explores all the possible subtrees of G'; hence it will find, among all the others, also T^*_{3-MVC}. Since the `MVCSearch` procedure returns the tree T_{3-MVC} having minimum cost, and T^*_{3-MVC} is the optimum among all possible *3-SMT* subtrees, we can conclude that $C(T_{3-MVC}) = C(T^*_{3-MVC})$ and then $\beta = \frac{C(T_{3-MVC})}{C(SMT)} \leq \frac{5}{3}$. □

Figure 6.6(c) shows the T_{3-MVC} tree returned by the `3-MVC` algorithm. Given $T_{3-MVC}(P_{3m}, E_{3m})$, building the tree $T(V' \bigcup S, D)$ of Definition 4 is straightforward: we simply add new vertexes to P_{3m}, by splitting each edge $e \in E_{3m}$ into sub-edges of length equal to R, and placing a RU in correspondance of each end point.

6.2.3 Final step

Based on the operations of the `MC-GDC` and `3-MVC` algorithms previously described, and on α and β bounds on the graph cost defined by Theorems 5 and 6, the two-step solution is elaborated in terms of graph weight, i.e. of number of EU devices connected by the RMN.

Theorem 6. *Let G^{opt} be the optimal solution of the MCC problem, and G^{apx} the solution returned by our two-step approach. Let M be the number of available*

RUs. If $C(G^{opt}) \geq M \cdot \gamma$, with γ=α+β then $W(G^{apx})$=$W(G^{opt})$, i.e. the two-step approach maximizes the number of EU devices connected.

Before providing the proof of Theorem 6, let us introduce the following lemmas.

Theorem 7 (Lemma 1)**.** *Let $G^{opt}(V^{opt}, E^{opt}, f^{opt}_{RU}, cov^{opt}_{EU})$ be the optimal solution of the MCC problem over the set of EU devices U with $C(G^{OPT}) \leq \frac{M}{\gamma}$ and let U^{cov} and $\overline{U}^{cov}$ be the set of covered and uncovered EU devices, respectively, by G^{opt}. If $C(G^{opt}) \leq \frac{M}{\gamma}$ then $\forall \overline{u} \in \overline{U}^{cov}$ and $\forall u \in U^{cov}$, $d(u, \overline{u}) > 2 \cdot R$, where R is the transmitting range of the RUs.*

Proof. From the hypothesis that $C(G^{opt}) \leq \frac{M}{\gamma}$ it can be noticed that there are some RUs that are not utilized in the final solution G^{opt}. Also, since $\gamma = \alpha + \alpha\beta + \beta$ (or $\gamma = \alpha + \beta$ depending whether using Theorem 1 and Theorem 2) with $\alpha \geq 1$ and $\beta \geq 1$, we have that $\gamma \geq 3$. By absurd, let us suppose that there exist a couple of EU devices u and $\overline{u}$, with $u \in U^{cov}$, $u \in cov^{opt}_{EU}(v)$, $v \in V^{opt}$, and $\overline{u} \in \overline{U}^{cov}$, with $d(u, \overline{u}) \leq 2 \cdot R$. Then, $\overline{u}$ might be covered by a RU v' not utilized so far: we also notice that the maximum distance between v' and v can not exceed $3 \cdot R$. Since by hypothesis $\gamma \geq 3$, we have still 2 RUs unused, and one of them (say v'') can be placed in between v and v' to connect them together. The new graph $G'(V^{opt} \cup \{v', v''\}$,$E \cup \{e(v, v''), e(v'', v')\}$,$f^{opt}_{RU} \cup \{f^{opt}_{RU}(v'), f^{opt}_{RU}(v'')\}$, $cov^{opt}_{EU} \cup \{cov^{opt}_{EU}(v'), cov^{opt}_{EU}(v'')\})$ is thus a solution of the MCC problem, with $W(G')$ = $W(G^{opt})$ + 1. This is absurd, since it contradicts the hypothesis that G^{opt} is the optimal solution with maximum weight. Hence, $d(u, \overline{u}) > 2 \cdot R$, $\forall u \in U^{cov}, \forall \overline{u} \in \overline{U}^{cov}$.

□

Theorem 8 (Lemma 2)**.** *Let $\kappa =< V, f_{RU}, cov_{EU} >$ be a solution of the MC-GDC problem over a set of EU devices U. If we consider as input to the MC-GDC algorithm a new set of EU devices $U' = U \cup \{u'\}$, with $d(u, u') > 2 \cdot R, \forall u \in U$, then the new solution $\kappa' =< V', f'_{RU}, cov'_{EU'} >$ is built from the old one κ without affecting it, i.e. $V' = V \cup \{v'\}$.*

Proof. The proof derives from the operations of Algorithm 1 in a straight-forward way. There are two possibilities when adding u': (*i*) the EU device u' is located on a new strip compared to EU devices covered by κ, and hence is not affecting the old solution, or (*ii*) the EU device u' is located on the same strip. However, since $d(u, u') > 2 \cdot R, \forall u \in U$, it can not be covered by any RU already placed in $\kappa.V$, and hence, a new RU must be added to the covering structure through the `Disk` function (see Algorithm 2). □

Proof. (**Theorem 6**) Let U be the set of EU devices in the target scenario. Moreover, let G^{opt} be the optimal solution of the MCC problem, and U^{opt} and $\overline{U}^{opt}$ be the set of covered and uncovered EU devices by G^{opt}. Let G^{apx} be the solution returned by our two-step approximation algorithm over set U.
First, let us suppose to execute our two-step approximation algorithm over set U^{opt} only. Let G'^{apx} be the solution returned by the two-step algorithm. From the hypothesis $C(G^{opt}) \leq \frac{M}{\gamma}$ and from Equation 6.3, we conclude that $C(G'^{apx}) \leq M$, i.e. $C(G'^{apx})$ is feasible in terms of number of RUs and is able to fully connect the set U^{opt}.
Let $\kappa_{U^{opt}}$ be the covering structure produced by Algorithm 1 over set U^{opt}. Let $T(\kappa_{U^{opt}}.V \cup P, E)$ be the *3-SMT* tree returned by Algorithm 4 with terminal points $\kappa_{U^{opt}}.V$ and Steiner points P; the *3-SMT* tree connects all the RUs of the covering structure. Let us now suppose to repeat the execution of our two-step approximation algorithm over EU device set U'=$U^{opt} \cup \{\overline{u}\}$ with $\overline{u} \in \overline{U}^{opt}$, i.e. adding one more EU device which is not covered by the optimal solution G^{opt}. By Lemma 1, we have that $d(u, \overline{u}) > 2 \cdot R$, $\forall u \in U^{opt}$. By Lemma 2, we have that -at Step 1- Algorithm 1 produces a new covering structure $\kappa_{U'}$, with $\kappa_{U'}.V = \kappa_{U^{opt}}.V \cup \{v'\}$, and $cov_{EU}(v') = \{\overline{u}\}$. Let us now consider the execution of Step 2 (Algorithm 3) to connect the RUs of the covering structure. Let TR be the set of *3-SMT* trees having as terminal points all possible subsets of $\kappa_{U'}.V$. Let $max(T) \in TR$ be the optimal 3-MVC tree having maximum weight, and cost lower than M. We observe that: (*i*) $T(\kappa_{U^{opt}}.V \cup P, E) \subseteq TR$, since $\kappa_{U^{opt}}.V \subseteq \kappa_{U'}.V$ and (*ii*) $T(\kappa_{U^{opt}}.V \cup P, E) = max(T)$, since $T(\kappa_{U^{opt}}.V \cup P, E)$ connects the optimal set U^{opt}, and Algorithm 4 will certainly find it since it checks all possible 3-MVC trees belonging to TR. In other words, the solution produced by our two-step approach is not affected by the adding of node $\overline{u}$. Let's now add all other EU devices of $\overline{U}^{opt}$ one by one as before, and iterate the execution of the approximate algorithms, till all the set U of EU devices have been considered. For the same reasons adducted before, the output of Algorithm 4 is not affected, and still equal to $T(\kappa_{U^{opt}}.V \cup P, E)$. Hence G^{apx}=G'^{apx}.
Finally, since G^{apx} and G^{opt} connects the same subset of EU devices, they have the same weight, i.e. $W(G^{apx}) = W(G^{opt})$.

□

Looking at the *Computational Complexity* (CC) of the two-step approach, this is mainly dominated by the `3-MVC` algorithm. Indeed, the CC of `MC-GDC` algorithm is O(N^2), since the `Disk` function (having complexity $O(N)$) is executed on each EU device, till the full scenario is covered by the RUs. The CC of the `3-MVC` algorithm has been analyzed in [24]. Using the results presented in [24], we can say that the CC of the `3-MVC` algorithm is in the order of $O(n^3)+O(N^{\frac{M}{\eta}})$, where η is the mean of the cost values $c(u(v_i, v_j))$ of the graph. Hence, the CC of the MCC algorithm is still exponential, but in terms of number of RUs (M), which is assumed to be constrained and much lower than the number of EU devices (N). Moreover, the `3-MVC` algorithm requires affordable computational resources over scenarios with low values of RUs (i.e. $M << N$) or acceptable values of the ratio $\frac{M}{\eta}$ (e.g. large-scale scenarios), as the ones considered in Section 6.4.

6.3 The Distributed Solution for the MCC Problem

In this Section, is introduced a distributed protocol to address the MCC problem. Given the computational complexity of the centralized solutions previously discussed, let us rely on a different approach that aims at achieving the self-configuration of the RMN through local communications among RUs, and decentralized decision making. In Section 6.4 will be demonstrated the effectiveness of the distributed solution compared to the centralized one. Here, it will provided a comprehensive description of the protocol's operations, including the aspects of mesh creation and maintenance (Section 6.3.1), exploration (Section 6.3.2) and communication among the RUs (Section 6.3.3).

6.3.1 Creation and Maintenance of a RMN

In this study, each RUs is provided with autonomous mobility and frequency configuration capabilities and hence cooperation is needed to guarantee the creation of stable mesh structures. More in detail, the distributed RMN algorithm must guarantee: (*i*) the connectivity of the mesh, i.e. no partitions must occur among the RUs, (*ii*) the Quality of Service (QoS) over the links among RUs and between RUs and EU devices, (*iii*) the space/frequency exploration of the post-disaster scenario to detect isolated EU devices. Although several swarm mobility have been proposed in the related literature, none of them is able to fulfill all three requirements at the same time. For this reason, here is described an extension of the *Virtual Spring Mesh* scheme described in [140] and [130], where it is introduced channel-aware mechanisms to take into account QoS requirements on each link of the RMN. Let us assume that, at each time instant, multiple virtual forces act on each RU_i, i.e. $\vec{F_{i,1}}$,$\vec{F_{i,2}}$, ...,$\vec{F_{i,n_i}}$. Let $\vec{F(i)}$ be the sum of these virtual forces acting on RU_i, i.e.:

$$\vec{F(i)} = \sum_{j=0}^{n_i} \vec{F_{i,j}} \tag{6.25}$$

The virtual forces can be of three different types, based on the three possible categories of links:

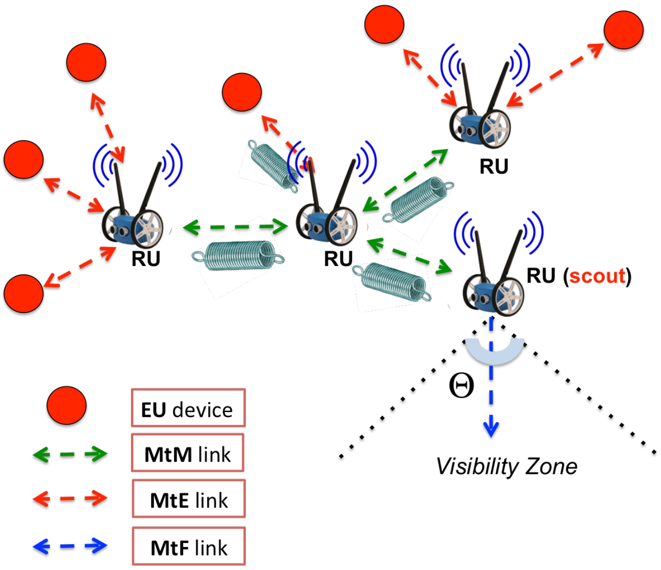


Figure 6.7: The MtM, MtE and MtF forces in a RMN deployment.

- Mesh-to-Mesh (MtM) links: that acts between two RUs of the RMN.
- Mesh-to-EU (MtE) links: that acts between a RU and an EU device.
- Mesh-to-Frontier (MtF) links: that acts on special RUs (called *scout* RUs) and are used to explore the scenario.

The MtM and MtE forces guarantee the connectivity of the RMN, while the MtF forces enable space exploration and mesh mobility. Figure 6.7 shows a (possible) deployment of the RMN, with different virtual forces applying on each link.

The details about the MtF forces will be provided in Section 6.3.2. The focus is now on the MtM and MtE forces, that are modelled as virtual springs according to well-known Hooke's law:

$$\vec{F} = -k \cdot \vec{x} \tag{6.26}$$

where k is the stifness constant and $\vec{x}$ denotes the spring displacement. In [130], the $\vec{x}$ factor is defined in terms of spatial distance between the end-points of the spring. Conversely, here this displacement is formulated as a function of the quality of the wireless link betweenthe two RUs (MtM link), or between a RU and an EU device (MtE link). More specifically, after receiving a message from RU_j or EU_j (details in Section 6.3.3), RU_i computes the Link Budget index of the link $i \leftrightarrow j$ (i.e. $LB(i,j)$) as follows:

$$LB(i,j) = Pr_j^i - RS_{thr}^i \tag{6.27}$$

Pr_j^i is the power received at RU_i, and RS_{thr}^i its receiving radio sensitivity. The LB index, also called fading margin [171], measures the communication reliability and indicates if the link is going to break. The requested link budget (LB_{req}) is used to model the Quality of Service (QoS) which must be guaranteed on each link of the RMN. Finally, the spring displacement ($\vec{x}$) is formulated as follows:

$$\vec{x} = sign() \cdot \vec{u} \cdot \delta \tag{6.28}$$

where $\vec{u}$ is the unit vector centred in RU, $sign()$ is a sign function assuming negative value whether $LB(i,j) > LB_{req}$ (i.e. force is repulsive since spring end-points are too close), and positive value whether $LB(i,j) < LB_{req}$ (i.e. force is attractive since spring end-points are too far). The factor δ is a scalar value providing the force module, based on requested and current LB on the $i - j$ link:

$$\delta = \sqrt[\alpha]{\frac{max(LB(i,j), LB_{req})}{min(LB(i,j), LB_{req})}} - 1 \tag{6.29}$$

where α is the propagation decay exponent that is dependent on the characteristics of the environment. Figure 6.8 depicts the δ values, as a function of $LB(i,j)$ and for different values of α; LB_{req} is set to 20dbm. It is easy to notice that: (*i*) δ is 0 when $LB(i,j)$ meets the LB requirements LB_{req}, (*ii*) δ increases when $LB(i,j) > LB_{req}$, reflecting the fact that the spring end-points are too close, (*iii*) δ increases more quickly when $LB(i,j) << LB_{req}$, reflecting the fact that the two nodes are moving out of their communication range and, hence, that the nodes must react quickly. The second component of Equation 6.26, i.e. the stiffness constant k, is defined separately for MtM and MtE links. If the spring displacement models the requested link quality, the k value quantifies the system responsiveness, i.e. how quickly RUs will re-act to changes of wireless conditions (e.g. due to mobility or fading effects), in order to re-establish the requested QoS value (LB_{req}).

- *MtM link.* k_{MtM} is a fixed parameter in the range [0:1] (equal to 0.5 in our experiments).

- *MtE link.* k_{MtE} is dynamically adjusted on the basis of number of EUs connected to RU_i (i.e. $n_{EU}(i)$) as follows:

$$k_{MtE} = \frac{n_{EU}(i)}{max(n_{EU}(j)) \, \forall j \in Neigh_i} \tag{6.30}$$

  We can see that the stiffness (and thus the force module) of the MtM spring is made proportional to the number of connected EUs at the current location, and compared to the actual CI of neighboring nodes. As a result, a RU connecting more EU devices at the current location will oppose more resistance to change its position than its neighbors.

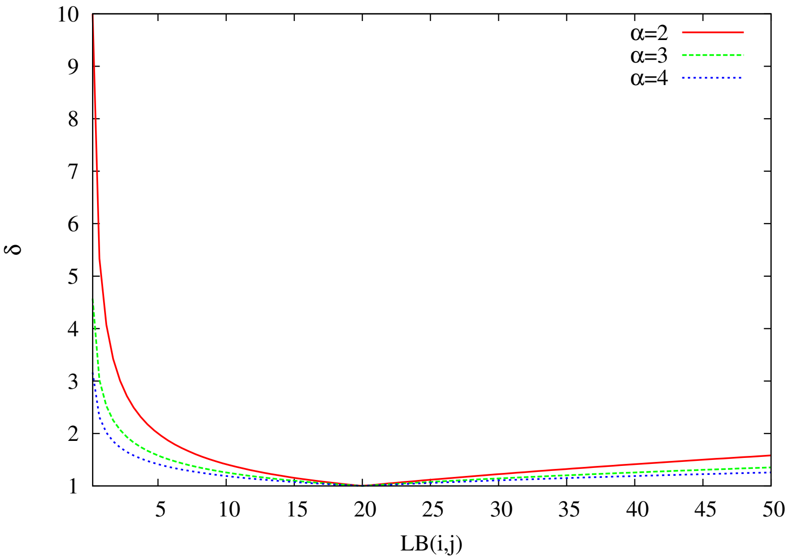


Figure 6.8: The displacement (δ) as a function of $LB(i,j)$, with LB_{req}=20 dbm.

The virtual force model enable each RU to autonomously adjust its position by computing the force $\vec{F(i)}$ periodically. However, in practice, additional mechanisms are required to avoid link breakage events caused by excessive values of $|\vec{F(i)}|$), and also to enforce mesh stabilization (i.e. convergence to stable positions). To this aim, we introduce a new metric, called *link criticality*, which is computed by each RU_i for each link $i-j$, as follows:

$$C(i,j) = 1 - \frac{min(LB(i,j), LB_{req})}{LB_{req}} \tag{6.31}$$

The value of $C(i,j)$ in Equation 6.31 is in range [0,1], where a value close to 1 indicates an high probability of link breakage. Given this index, RU_i should avoid modifying its position with respect to RU_j or to EU_j if the link in in critical condition. Similarly, let us introduce a minimum threshold on the value of $C(i,j)$, i.e. C_{thr}. If $C(i,j) > C_{thr}$, let us introduce a friction force $F\vec{R}_{i,j}$, which acts in the opposite direction than $\vec{F_{i,j}}$, and with intensity given by $|\vec{F_{i,j}}| \cdot C(i,j)$. Let S_i the set of all friction forces applying on RU_i, and $\vec{F_i}$=$\sum_{j=0}^{n} \vec{F_{i,j}}$. Let us refine Equation 6.25 by computing the resultant force as follows:

$$\vec{R(i)} = \vec{F_i} + \sum_{S_i} F\vec{R}_{i,j} \tag{6.32}$$

Figure 6.9 shows an example of how the friction force works in practice, so that RU_1 avoids breaking link with RU_3 thanks to the friction force $F_{1,3}$.

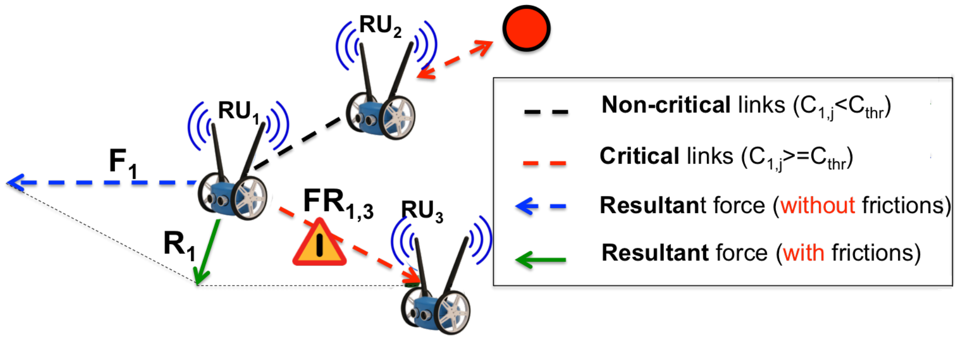

RU₂
RU₁
F₁
FR₁,₃
RU₃
R₁
Non-critical links (C₁,ⱼ<C_thr)
Critical links (C₁,ⱼ>=C_thr)
Resultant force (without frictions)
Resultant force (with frictions)

Figure 6.9: Example on friction forces acting between RUs: link $1-2$ is critical, and thus a friction force $FR_{1,2}$ is introduced. Through Equation 6.32, RU_1 computes the adjusted resultant force R_1 in order to avoid breaking link 1-2.

Every T_{DEF} period, each RU_i computes direction and intensity of $\vec{R(i)}$. If $|\vec{R(i)}|$ is higher than a threshold value R_{thr}, then RU_i moves towards the direction indicated by $\vec{R(i)}$, with constant speed equal to v. Otherwise, RU_i does not modify its current position.

6.3.2 Exploration method

The exploration of the scenario is crucial to allow the RUs detecting the isolated EU devices in order to connect them to the RMN. Two types of explorations are considered in our framework: (*i*) spatial exploration, and (*ii*) frequency exploration.

Spatial exploration is critical if uncoordinated mobility of each RU might easily lead to mesh partitions. For this reason, space explorations is delegated to special RUs, defined as *scout* nodes. First, is defined the visibility zone of RU_i with respect to vector direction $\vec{d}$ as the cone centered at RU_i with a sweep angle of θ (shown in Figure 6.7). At each beaconing interval, each RU_i verifies whether there are other RU_j in its visibility zone. In the negative case, the RU_i decides with probability ϵ whether to elect himself as *scout* node. The parameter ϵ governs the trade-off between space exploration and exploitation of already bridged EU devices, and constitutes an important parameter for the mesh formation. For a scout RU, is introduced an additional spring force, applied along direction $\vec{d}$, and modelled as a pure attractive force with intensity equal to F_{exp} (constant value).

Frequency exploration is performed by each RU. At periodic time intervals, each RU_i senses the N_{freq} frequencies through its free radio interfaces, in order to detect `HELLO` messages sent by the EU devices. To balance the sensing efforts among the RUs, and to enhance the EU discovery, each RU_i keeps statistics on the number of sensing actions performed on each frequency h, considering also neighbors' sensing actions. Let $SE_i(h)$ be the current statistic for frequency h and RU_i. At each T_B interval, RU_i senses the frequency h', which is the least sensed (and thus mostly unexplored) channel based on $SE_i(h)$, i.e.:

$$h' = argmin_{\forall h \in N_{freq}} SE_i(h) \tag{6.33}$$

Algorithm 5: Distributed mobility/radio frequency control of RU_i

```
Broadcast BEACON message structured as in Equation 6.34.
Compute the forces $\vec{F(i,j)}$ acting on each MtM/MtE/MtF link.
Compute force $\vec{F(i)}$ according to Equation 6.25.
Compute node criticality of each link $i - j$ according to Equation 6.31.
Determine resultant force $\vec{R_i}$ according to Equation 6.32
if $|\vec{R_i}| > R_{thr}$ then
    Move with speed $v$ along the direction indicated by $\vec{R_i}$.
end if
Decide frequency $h$ to sense according to Equation 6.33.
if $EU_j$ discovered on channel $h$ and $\exists k|$ frequency[k]=NULL or
frequency[k]=$h$ then
    Re-configure the radio interface to operate on channel $h$.
    Increase $n_{EU}$ by one.
end if
```

6.3.3 Communication among the RUs

Mesh creation and maintenance operations is based on periodic broadcast communication. In this way the nodes are able to (*i*) compute the LB metric at each receiving node (through Equation 6.27), and (*ii*) disseminate the information about last sensed channel, to allow distributed sensing scheduling operations (through Equation 6.33). Hence, every T_{BEACON} interval, each RU_i broadcasts a BEACON message containing the following information:

$$< n_{EU}(i), h_{SENSED} > \tag{6.34}$$

where $n_{EU}(i)$ is the number of EU devices currently connected to RU_i, and h_{SENSED} is the frequency scanned by RU_i during the last beaconing interval. Estimating the Angle-of-Arrival (AA) of the message, RU_i can also determine the direction of the virtual spring of the link $i - j$. The impact of T_{BEACON} on the convergence of the mesh creation process is investigated in Section 6.4. We also assume that isolated EU devices will periodically broadcast HELLO messages, in

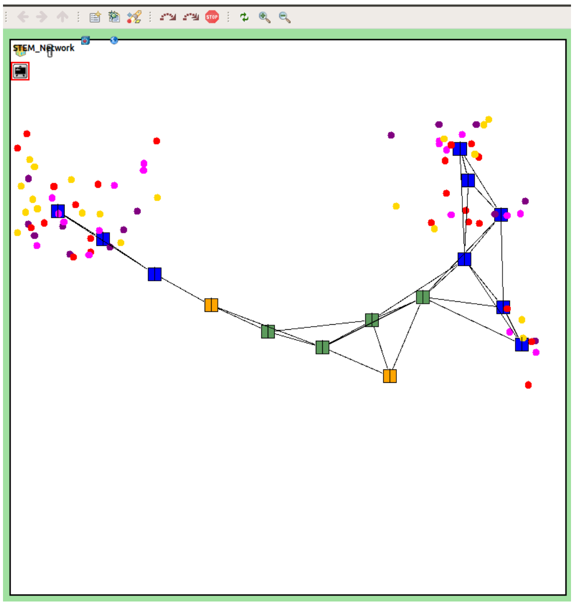


Figure 6.10: Screenshot of an OMNeT++ simulation, showing the deployment of the RMN through the MCC distributed algorithm, on a target scenario.

order to favor localization from the RUs.
When the RUs are injected in the scenario, all the N_{sdr} radios have the attribute `frequency` set to `Free`. Each time RU_i detects a `BEACON` message from EU_j on frequency h, it checks whether it has a radio k with `frequency[k]=NULL` (i.e. radio not used) or `frequency[k]=`h (i.e. radio already tuned to channel h). If one of these conditions is satisfied, then EU_j is connected to the mesh through RU_j, by setting `frequency[k]=`h. In Algorithm 5 is summarized the operations of the distributed protocol for mesh creation.

6.4 Evaluation of the Distributed MCC algorithm

In this Section, are evaluated the performance of both the Centralized MCC and Distributed MCC algorithms. To this purpose, are modeled in OMNeT++ the characteristics of a post-disaster scenario, based on the system model described in Section 6.1. N EU devices are considered uniformly distributed over an area of 1500x1500m^2. At the beginning, M RUs are injected in the center of the scenario and start organizing themselves in order to create the RMN. The movement of each RU is modelled as constant speed (equal to 3 m/s). Each RU is then provided with three radio interfaces (so $N_{sdr} = 2$), one tuned to the CCC in order to communicate with other RUs, and the other two tunable to any of the N_{freq} channel, in order to connect to the EU devices. If not specified otherwise, in the experiments will be used these values for the system parameters: N_{freq}=4, α=2, T_{BEACON}=2 seconds, $LB_{req} = 13$ dbm, RS_{thr}=-120 dbm, Pt (transmitting power)=110 mW, N_{sdr}=2. Since the goal of this study is to investigate the effectiveness of the RMN in maximally re-establishing network connectivity in a post-disaster scenario, the Connectivity Index (CI) of Equation 6.1 is referred as the main performance metric. Four different algorithms will be compared for the deployment of the RMN:

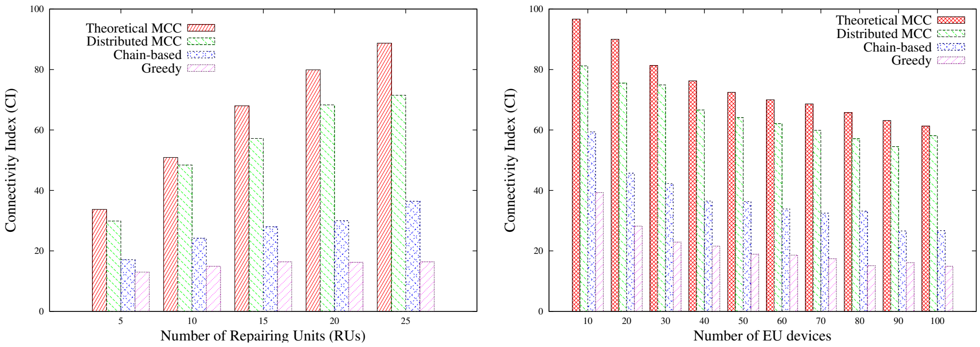
- *Centralized MCC.* Is the approximated two-step algorithm described in Section 6.2. Here, the configuration of each RU in terms of position/channels is statically decided at the simulation startup, and no communication/cooperation phase is required.

- *Distributed MCC.* This is the distributed deployment algorithm described in Section 6.3.

- *Chain-based algorithm.* This corresponds to the scheme proposed in [172], through which mobile robots self-organize into backbone structures. In the used implementation, the mobile robots are replaced with the RUs. During the simulation it is created a dynamic chain structures whose head and tail are the EU devices to be connected. Frequency allocation is performed using the same algorithm defined in Section 6.3.2 for the Distributed MCC scheme.

- *Greedy algorithm.* This corresponds to a distributed algorithm in which RUs explores the space/frequencies independently one from each other. Each time RU_i detects another EU device or another RU in its communication range, it decides with fixed probability (equal to 0.5 in our experiments) whether to stop or to keep exploring the scenario.

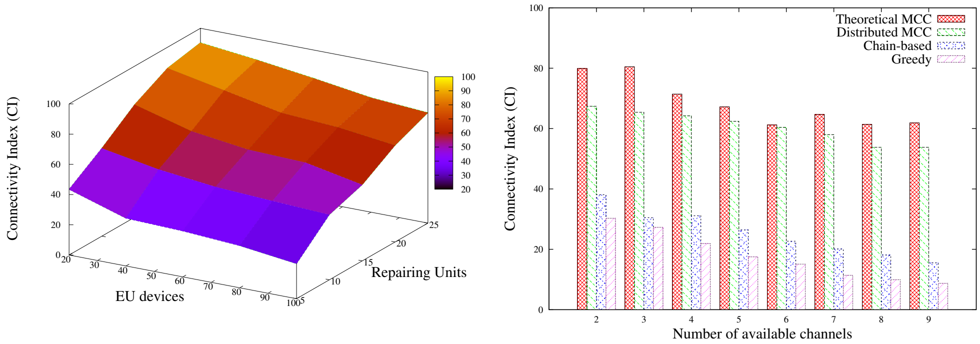
Figure 6.10 shows an OMNeT++ screenshot of the operations of the *Distributed MCC* algorithm, on an example scenario. Rectangles represent the RUs that are depicted with different colors based of their roles in the RMN: yellow rectangles are scout RUs, blue ones are RUs connected to EU devices, green are relay RUs. On the other side, circles represent the EU devices where different colors reflect the channel they are tuned to. In Section 6.4.1, it will be evaluated the CI of the four algorithms when varying the density of RUs and EU devices. In Section 6.4.2, it will be analyzed convergence over time of the swarm mobility scheme. Finally, in Section 6.4.3, it will be studied the impact of propagation conditions on system performance, by also providing preliminary results on the difference between terrestrial and aerial deployment of the RMN.



(a)

(b)

Figure 6.11: The CI for the four RMN deployment algorithms as a function of the number of RUs and of the number of EU devices are shown in Figures 6.11(a) and 6.11(b), respectively.



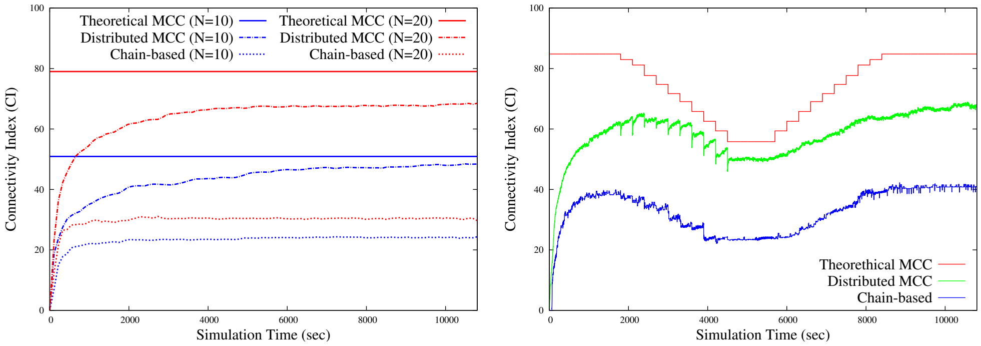
(a)

(b)

Figure 6.12: The CI of the Distributed MCC algorithm, as a function of the number of RUs and of the EU devices is shown in Figure 6.12(a). The CI for the four RMN deployment algorithms over the N_{freq} values is shown in Figure 6.12(b).

6.4.1 Connectivity Index Analysis

Figure 6.11(a) shows the CI as a function of the number of available RUs (i.e. M), depicted on the x axis. The number of EU devices is fixed to 65 (i.e. N=65); EU devices are uniformly spread out over the simulation area. As expected, the CI of all four algorithms increases with the number of RUs, since a better coverage of the scenario can be achieved; vice versa, uncovered areas might likely occur for low values of M. Moreover, Figure 6.11(a) demonstrates that the Distributed MCC algorithm can outperform the Chain-based and Greedy approaches in terms of number of EU devices connected, and can provide performance close to the Centralized MCC. We found that the system performance of the Distributed



(a)

(b)

Figure 6.13: The CI over simulation time is shown in Figure 6.13(a). The CI on a dynamic scenario with varying number of active RUs (M) over time is depicted in Figure 6.13(b)).

MCC algorithm are affected by the parameter ϵ, which determines the percentage of scout RUs, and thus the amount of exploration performed within the scenario (ϵ is fixed to 0.1 in these experiments). Adaptive tuning strategy of the ϵ parameter will be considered as future studies.

In Figure 6.11(b) is performd a similar analysis, but varying the number of EU devices while keeping constant the number of available RUs (M=15). Conversely to Figure 6.11(a), the CI decreases for all four algorithms when increasing the value on the x axis, indicating the fact that 15 RUs might not be enough to connect all EU devices of the scenario. Again, the Distributed MCC scheme provides performance close to the Centralized MCC, with a significant improvement over the Chain-based and Greedy algorithms.

Figure 6.12(a) combines previous results by showing the CI metric for the Distributed MCC scheme, as a function of the number of available RUs (in the x axis), and of the number of EU devices (in the y axis). Clearly, the CI achieves its maximum/minimum values for extreme opposite configurations of M and N (i.e. maximum for M=25, N=20, minimum for M=5, N=100). Moreover, CI is higher in configuration with $M = 25, N = 100$, than configuration with $M = 5, N = 20$, since the RMN can cover a larger area, and hence connect more EU devices.

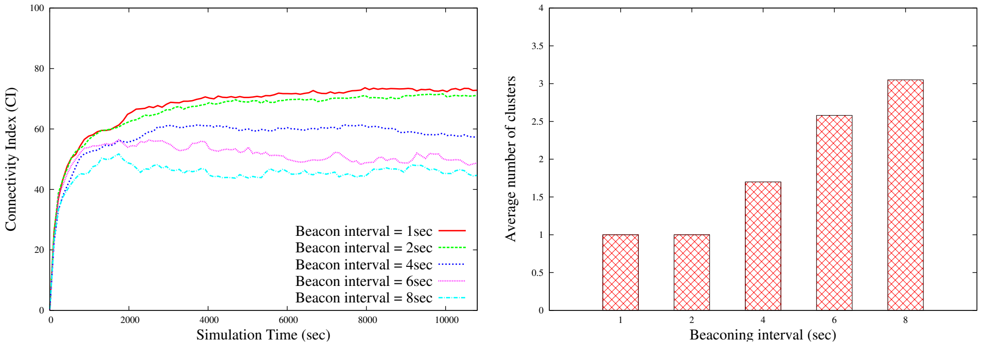
Figure 6.12(b) shows the impact of available channel resources on system performance. We consider a configuration with N=75, M=15, and we vary (on the x axis) the N_{freq} value. Based on system model introduced in Section 6.1, each EU device is tuned to a communication channel randomly decided among the available N_{freq} frequencies. Increasing this value augments the frequency diversity of the EU devices, and poses additional challenges to the creation of the RMN, since each RU has only two radio interfaces to communicate with EU devices (i.e.

N_{sdr}=2). Three important configurations can be drawn from Figure 6.12(b): (*i*) CI decreases with N_{freq} for all four schemes, since more RUs might be needed to cover the same set of EU devices, (*ii*) thanks to the spatial/frequency exploration scheme described in Section 6.3.2, the Distributed MCC algorithm experiences values of CI greater than 50% even for extreme values of N_{freq} (i.e. $N_{freq} = 9$) (*iii*) spatial density of the EU devices (shown in Figure 6.11(b)) produces an higher impact on CI performance than frequency diversity.

6.4.2 Convergence Analysis

In this Section, are provided insights on the behavior of the Distributed MCC algorithm, and is investigated the convergence over time of the swarm mobility algorithm. To this aim, in Figure 6.13(a) is depicted the CI values over simulation time, for a scenario with N=65, N_{freq}=4, and two configurations of M (equal to 10 and 20, respectively). The horizontal lines represent the CI values produced by the Centralized MCC algorithm. Here, it is omitted to draw the lines of the Greedy algorithm, since performance are much lower than the other schemes. The convergence of the Distributed MCC algorithm is reflected by the fact that the CI value increases during the initial exploration phase, till a stable value is reached at around 4000 seconds: after that, RUs do not further modify their positions. Also, it is possible to notice that the Distributed MCC provides equal performance than the Centralized MCC for M=10, and only slightly lower for M=20. In both the case, the Distributed MCC algorithm outperforms the Chain-based scheme.
Figure 6.13(b) demonstrates the adaptiveness of the proposed scheme to dynamic network environments. The CI value is depicted over simulation time, on a scenario with N=50 and N_{freq}=4. At the simulation startup, M is set to 20 RUs. From t=2000 till t=4000, is is assumed that the number of available RUs (e.g. M) is decreased by 1 every 200 seconds, till M=10 at t=4000 seconds. This behavior can be justified by assuming uncorrelated failures occurring at different RUs. From t=4000 till t=6000 seconds, the value of M is kept constant. From t=6000 to t=8000, a RU is added to the scenario every 200 seconds. In this way is simulated the fact that failures have been fixed or that new devices are being progressively injected by the safety agency to replace the damaged ones. Figure 6.13(b) shows that the Distributed algorithm dynamically adapts the RMN coverage based on current availability of RUs, without experiencing deep losses of service.

In Figures 6.14(a) and 6.14(b) is investigated the impact of the beaconing interval (i.e. T_{BEACON}) on convergence of the Distributed MCC algorithm. More specifically, Figure 6.14(a) depicts the CI metric over simulation time, for a configuration with M=20, N=65 and different values of T_{BEACON}. Smaller beaconing intervals correspond to higher CI values, since the RUs can better adjust their reciprocal positions when receiving frequent updates about the quality of

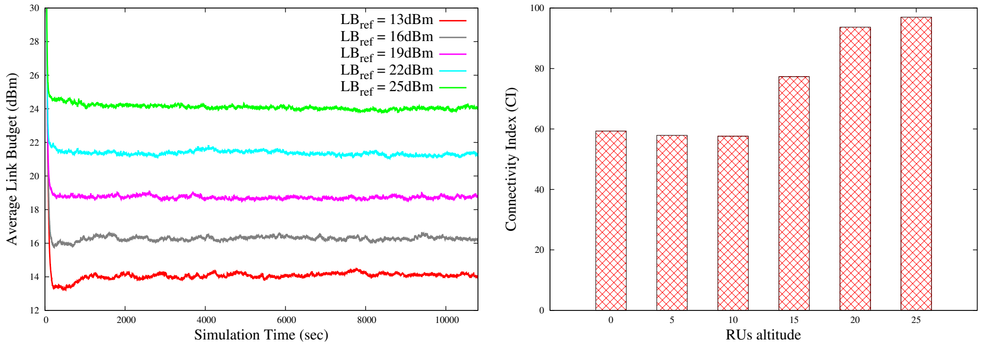


(a) (b)

Figure 6.14: The impact of T_{BEACON} parameter in terms of CI and number of clusters created among RUs is shown in Figure 6.14(a) and 6.14(b), respectively.

wireless links toward neighbor nodes. Vice versa, for T_{BEACON}=6 seconds and T_{BEACON}=8 seconds, the CI decreases significantly and presents some fluctuations, since link breakages can occur within the RMN. This is also confirmed by Figure 6.14(b), in which we depict the average number of RU clusters created during the simulation of the previous scenario. For T_{BEACON}=1 second and T_{BEACON}=2 seconds, only one cluster is created, i.e. the RMN is fully connected. For values of T_{BEACON} larger than 2 seconds, the RMN is likely partitioned, since RUs move outside their communication range before receiving next `BEACON` message, and thus they can not activate the criticality mechanisms (described in Section 6.3.1) that should prevent the breakages of RMN links. Clearly, using low beaconing intervals guarantees the stability of the RMN, but also introduce a significant communication load; for this reason, the trade-off between network overhead and convergence of the Distributed MCC algorithm should be considered on each target scenario.

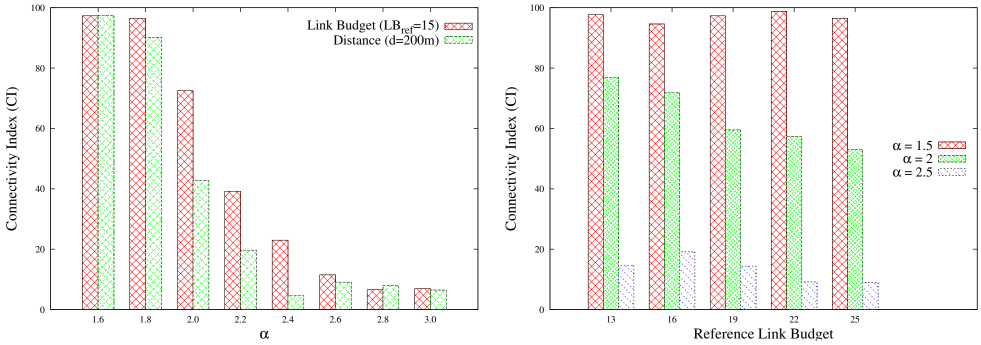
Finally, Figure 6.15(a) demonstrates the ability of Distributed MCC to meet QoS requirements during the RMN deployment; the QoS requirement is defined by the LB_{req} value, which is statically decided before network creation by safety agencies, based on characteristics of applications which must be supported by the RMN. It is considerd a scenario with N=50, M=15, T_{BEACON}=2 seconds, and it is plotted the average LB values on the MtM/MtE links of the RMN, over simulation time. The six curves represent different setting of the LB_{req} value. For each curve, there is an initial transient phase caused by the fact that all RUs are injected from the center of the scenario, at the simulation start. However, after few hundreds of simulated seconds, the RUs autonomously adjust their reciprocal positions, so that the average link budget (on the y axis) meets quite perfectly the requested LB_{req} value on each MtM/MtE link. It is worth remarking the fact



(a)

(b)

Figure 6.15: The average LB over simulation time, and for different settings of LB_{req} is shown in Figure 6.15(a). The CI for different altitude from ground deployment of the RMN. (Figure 6.15(b)).



(a)

(b)

Figure 6.16: The CI for different displacement functions (Figure 6.16(a)) and varying propagation conditions (Figure 6.16(b)).

that QoS provisioning in the Distributed MCC scheme is achieved through local communication only, and without assuming the knowledge of the current positions of the RUs.

6.4.3 Environment Analysis

Differently from previous experiments where it was assumed a free-space outdoor scenario without obstacles, in this Section are analysed the impact of propagation conditions on the performance of the Distributed MCC algorithm and is demonstrated the effectiveness of channel-aware mechanisms (used by the virtual spring model) during RMN creation. Figure 6.16(a) refers to a scenario with N=50, M=20. The CI values is depicted on the y axis; in the experiment, the exponent of the decay is varied propagation model (i.e. α) on the $x$axis. The Distributed MCC algorithm is modified by considering two alternative mechanisms to compute the displacement of the virtual spring on the MtM links:

- *Distance-based*: in this case, the spring displacement is computed as a function of spatial distance among the RUs, as proposed in [130]. We consider a reference distance value of 200 meters.

- *Channel-based*: in this case, the spring displacement is expressed as a function of the LB metric, and computed through Equation 6.27. We consider a reference LB value (LB_{req}) of 15 dbm.

For α=1.6, the reference distance value corresponds to the requested LB_{req} value. Figure 6.16(a) demonstrates the ability of channel-based mechanisms to adapt the RMN deployment to dynamic channel and propagation conditions, differently from existing distance-based metric described in [130]. Indeed, when increasing the α value, the CI of the distance-based configuration experiences a deep fall since the reference spring displacement (200 meters) might be greater than the RU communication range. Hence, frequent partitions might occur when RUs adjust their positions in order to meet the reference displacement value, according to Equation 6.26. Conversely, using the channel-based mechanism, propagation conditions are implictly taken into account by Equation 6.27: when increasing the value of α, the RUs automatically adjust their positions by getting closer one from each other, in order to meet the requested LB_{req} value on each link of the RMN. This also produces a decrease in terms of CI since the RMN will cover a smaller area; however, the channel-based mechanism greatly outperforms the distance-based in terms of average CI values, since no partitions occur.
Figure 6.16(b) depicts the CI values experienced by the Distributed MCC algorithm (channel-based mechanisms), as a function of the reference LB value (LB_{req}) on the x axis. We considered three different propagation characteristics, modeled through three different values of α. We can notice that: (*i*) higher values of LB_{req} translate into lower CI values, due to the reduced distance among neighbor RUs, and thus to the reduced coverage of the RMN, (*ii*) for the same value of LB_{req}, the CI is highly influenced by the propagation conditions.
Finally, in Figure 6.13(b) are provided preliminary insight on the impact of the altitude of the RUs on CI performance, and is elaborated on the possibility to deploy aerial RMN to re-establish connectivity among isolated EU devices. To this aim, is considers a Manhattan grid scenario of 500 meters x 500 meters, where we modeled the presence of 30 buildings of different heights. Each building has a rectangular shape and its height is randomly in range of [10:20] meters. It is assumed that EU devices are randomly placed on the ground, at indoor or outdoor

positions. The RMN is deployed by RUs mounted on Small Aerial Unmanned Vehicles (SUAVs), and moving according the Distributed MCC algorithm described so far. However, differently from previous analysis, we consider a propagation model which takes into account the attenuation caused by the presence of obstacles on the Line-of-Sight (LOS) between a wireless transmitter and receiver. More details about the propagation model can be found in [173]. Figure 6.15(b) shows the CI when we vary the altitude (h) from ground of the RMN. The configuration with h=0 corresponds to the RMN deployed on the ground, as in the analysis presented so far. The reference LB_{req} value is equal to 25 dbm for all the configurations. For $h < 10$ meters, no variations of the CI can be appreciated, since most of communications are subject to shadowing effects as for the terrestrial case. Vice versa, when $h \geq 20$ meters, all the MtM links experience LOS propagation conditions, since the RUs are located over the roofs of the buildings. This translates into an increase of the CI value, since the RUs can be more distant one from the others, and hence the RMN can cover a larger portion of the scenario. Results in Figure 6.15(b) support the idea of using aerial communication for disaster recovery operations, as suggested in [98].

6.5 Experimental test-bed

In this Section, is provided an experimental validation of the operations of the *Distributed MCC* algorithm through a small scale test bed. More specifically, the aim is to demonstrate the ability of RUs to autonomously control their positions so that the per-link QoS requirements are satisfied. In these experiments, the RUs are represented by (two) small Unmanned Ground Vehicles (UGV), each provided with wheels, an Arduino Two [1] controller, and two Wi-Fi 802.11 radio interfaces operating on the 2.4 GHz band. The mobility of each RU is controlled by the virtual spring model described in Section 6.3. The test-bed is built as follows:

- EU devices are represented by two laptops, placed at distance D one from the other, in an outdoor scenario (no obstacles in-between). Each laptop is provided with an 802.11 Wi-Fi radio.

- The RUs are placed on the straight line connecting the laptops (at random positions when the experiment starts).

- The RUs periodically transmit `BEACON` messages, in order to estimate the LB of each MtM/MtE link, and to compute the virtual spring displacement value according to Equation 6.29.

[1] http:\\www.arduino.cc

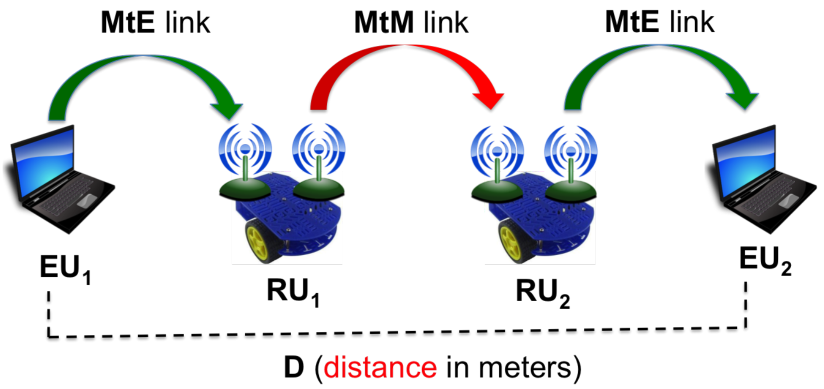


Figure 6.17: Network topology considered in the test-bed.

- The RUs must adjust their positions (moving on the straight line) in order to enable multi-hop communication between the two laptops. A radio is used to communicate on the MtM link, while the other one is used to communicate on the MtE link.

Figure 6.17 depicts the scenario used in the experiments. Two metrics for the analysis are considered:

1. *Minimum Link Budget* (LB_{min}). This is defined as the minimum LB value over all links of the RMN, i.e. LB_{min}=$min_{i,j}$ $LB(i,j)^2$. The LB_{min} metric can be seen as an indicator of the end-to-end performance of the RMN, considering the quality of the bottleneck link.

2. *Link Budget Error* (LB_{err}). This is defined as the maximum difference between the LB values experienced on the two links of RU_i, for each RU_i, i.e. LB_{err}=$max_i|LB_{i,j-1} - LB_{i,j+1}|$.

It is easy to notice that the maximum theoretical LB_{min} value is achieved when all nodes are placed at equal distance one from each other, i.e.:

$$LB_{mm} = P_{tr} - PL(\frac{d}{M+1}) - RS_{thr} \tag{6.35}$$

Here, $PL(x)$ represents the path-loss at a given distance x according to the well-known exponential propagation model [171]:

$$PL(x) = 10 \cdot \alpha \cdot log_{10}(\frac{x}{d_0}) + PL(d_0) \tag{6.36}$$

where α is the propagation decay exponent (set to 3) and d_0 is a reference distance (set to 20 meters in our experiments).

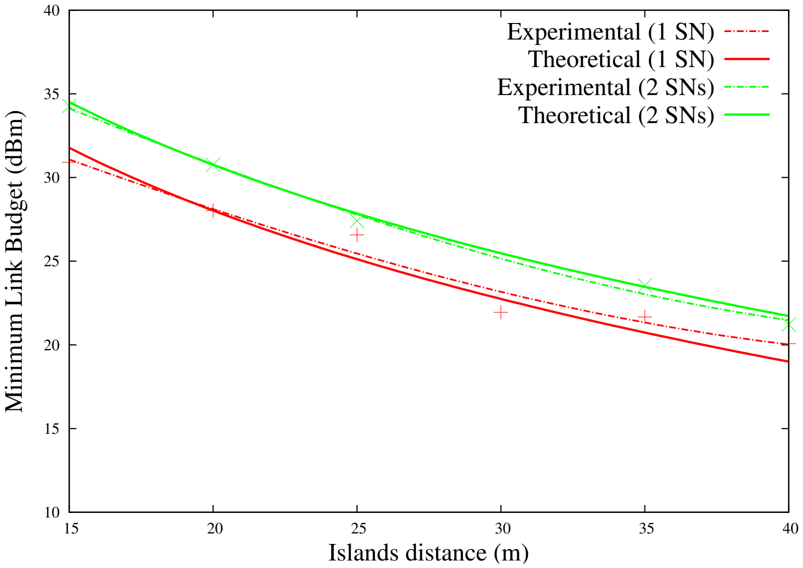
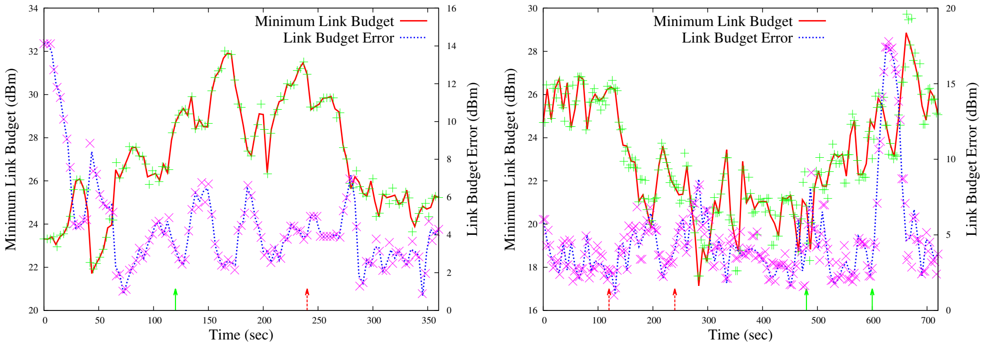


Figure 6.18: The experimental and theoretical LB_{min} values.



(a)

(b)

Figure 6.19: The values of LB_{min} and LB_{err} when dynamically varying the number of RUs and the distance D during the experiments are shown in Figures 6.19(a) and 6.19(b), respectively.

Figure 6.18 shows the experimental LB_{min} (dotted lines) and theoretical LB_{mm} (full lines), when we vary the distance between the laptops (i.e. D). The experiments are repeated by using one RU and two RUs. As expected, the LB_{min} values decrease while increasing the distance D. Moreover, the simulation results overlap quite perfectly with the theoretical values (LB_{mm}) for all the configurations of D, thus confirming the ability of the distributed control algorithm to maximize

[2] $\forall i, j \in \{EU_1, RU_1, RU_2, EU_2\}$, and $i \neq j$.

the link quality among the RUs.

Figures 6.19(a) and 6.19(b) investigate how the *Distributed MCC* algorithm can adapt the RMN deployment in dynamic environments. In Figure 6.19(a), it is considered an initial scenario with D=20, and one RU placed at a random position inbetween. As soon as the experiment starts, the RU moves toward the central position between the laptops. After 90 seconds, the LB_{min} is around 27 dBm: this perfectly matches the theoretical result depicted in Figure 6.18 (configuration with one RU). At time t=120 seconds, one more RU is added to the scenario[3]. We can see that the LB_{min} increases due to the fact that the average link length of the RMN is halved. At time t=240 seconds, one of the RU is removed from the scenario, thus restoring the original setting of the experiment[4]. As a result, LB_{min} drops to around 27dBm, as in the previous case. In the same Figure, we also plot the LB_{err} values on the x-$y2$ axes. We can notice that LB_{err} values with two RUs present larger fluctuations than the case with one RU, due to the relative movements of the RUs.

In Figure 6.19(b), the same analysis is repeated but on a slighly different dynamic environment: the number of RUs is equal to two over all the duration of the experiment, while the distance D between the laptops is varied. At time t=0 seconds, D is equal to 20 meters. We increase D to 30 meters at t=120 seconds, and to 40 meters at t=240 seconds. We do the opposite after a while, i.e. we decrease D to 30 meters at t=480 seconds, and back to 20 meters at t=600 seconds, thus restoring the original configuration. The behaviour of LB_{min} looks as expected: the LB decreases when the laptops are moved away ($D > 20$), while increases when they approach, till it becomes equal to 27 dBm for D=20 and t=700 seconds. The LB_{err} metric (on the x-$y2$ axis) is almost stable over simulation time, with small variations in correspondence of topology changes at t=120, 240 and 480 seconds. Moreover, LB_{err} values in Figure 6.19(b) are lower than Figure 6.19(a). Hence, it can be argued that varying the position of EU devices produces less impact than modifying the number of RUs. In Figure 6.19(b), a peak is observed at t=600 seconds, due to the fact that we reduced the distance D in an asymmetric way (i.e. moving closer only one laptop of 10 meters, instead of moving both of 5 meters). However, we can notice that the LB_{err} value immediately drops below the 5 dbm after few seconds. Both results of Figures 6.19(a) and 6.19(b) confirm the fact that the channel-aware mechanisms used by the Distributed MCC algorithm guarantees an adaptive deployment of the RMN in presence of dynamic scenarios, like the emergency ones.

[3] In Figure 6.19(a), this event is highlighted by the green arrow on the $x-$axis.

[4] In Figure 6.19(a), this event is highlighted by the red arrow on the $x-$axis.

Chapter 7

Energy constrained systems

In the aftermath of a large-scale emergency, the breakdown of communication infrastructure negatively impacts the actions of the first responders, i.e. the cooperation and the dissemination of information to the general population [98], as demonstrated by recent catastrophic events worldwide (e.g. the earthquake in Italy in 2012). For this reason, there is strong incentive towards the realization of backup communication systems that are able to quickly self-deploy in the first hours after an emergency situation and ensure temporary network services in the affected area. A recent report from FCC proposes the utilization of low-altitude, *Deployable Aerial Communications* (DAC) systems to support first response operations on post-disaster scenarios because of their advantages over the traditional terrestrial infrastructures [98]. DAC systems can provide more coverage than ground wireless networks as aerial links are less affected from fading and are more suited when the mobility on the roads has been compromised. Nowadays, deployment of low-altitude DAC systems is feasible thanks to the increasing availability and affordability of Small Unmanned Aerial Vehicles (SUAVs), such as multi-copter drones. However, when planning for the aerial coverage of large-scale emergency scenarios for at least the first 48 hours, coordinated mobility and energy issues must be taken into account [174]. Network mobility is known to be highly challenging in 3D environments, and only few works investigate the creation of flying self-organizing swarms, specially designed for rescue operations [175, 39, 40, 41, 42].

In this Chapter, it will be addressed how to combine the controlled mobility and the energy issues in the deployment of SUAV-based mesh networks for backup communication systems in post-disaster scenarios. More specifically, as in the previous Chapter, it will be considered an emergency scenario, where not all End-Users (EUs) devices are connected to each other, and the aerial mesh attempts to build the links between them. In this Chapter it will be provided improvements to the model described in the previous one with three new main contributions: (*i*) it is extended the distributed algorithm [145, 24] presented in the previous Chapter 6 in the context of a swarm of SUAVs, that allows them to self-organize into an aerial mesh to maximally connect the EUs on the ground. The mobility scheme is based on the Virtual Spring force model [130] and introduces channel-aware metrics in order to guarantee a minimum link quality on the air-to-ground and air-to-air links. (*ii*) It is modelled a realistic 3D urban environment in the simulation tool, with shadowing effect caused by buildings, and it is investigated the benefits provided by an aerial mesh deployment compared to a terrestrial one (for example through wheeled mobile robots), in terms of coverage and link stability with different altitudes. (*iii*) It is investigated the approaches to maximize the lifetime of the aerial mesh by considering a scenario where SUAVs can recharge their batteries through contact with the ground recharging stations. A distributed scheduling algorithm is proposed that provide the coverage of the disaster area by the SUAVs while maximizing the network lifetime.

7.1 System Model

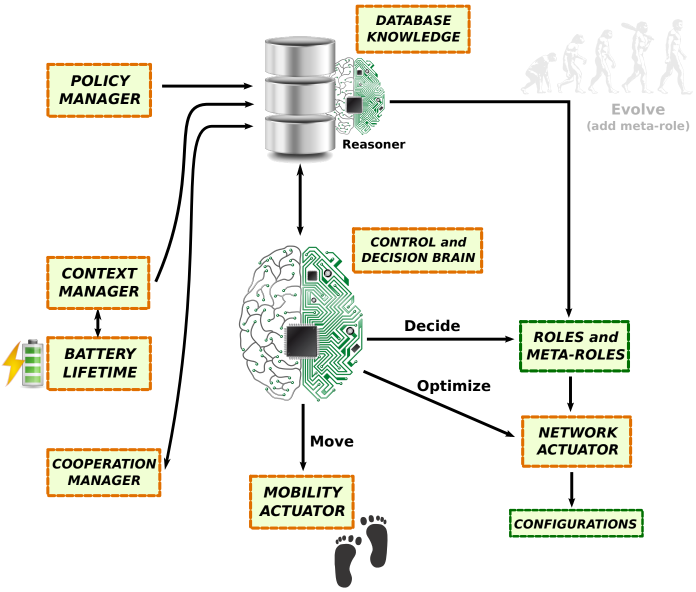


Figure 7.1: The used subset of the Stem-Node architecture with the Battery-Lifetime module

Let us consider an urban, post-disaster scenario shown in Figure 7.2, with buildings of different heights, and N End-User (EU) devices located in outdoor

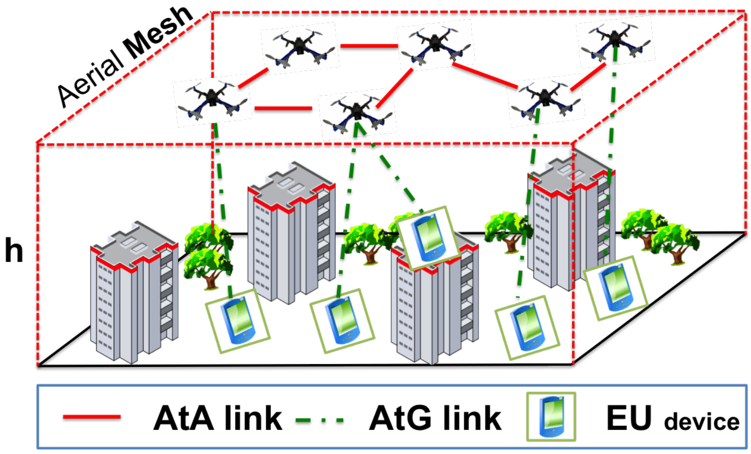


Figure 7.2: The emergency scenario and the aerial mesh deployment [173].

or indoor position. In the aftermath of the emergency, isolated partitions of EUs might emerge, who must be eventually connected through additional Repairing Units (RUs). Each RU is constituted by a SUAV (multicopter), flying at an initial altitude of h meters, and equipped with a wireless radio transceiver to communicate with other RUs (air-to-air link) and to EU devices (air-to-ground link). Each RU is battery-powered and consumes energy for flying and for radio communication. Here, it is used the same sub-set of the STEM-Node architecture used in Chapter 6 with the *Context Manager* module equipped with *Battery Lifetime* module that adds the battery-level awareness of the RU.

For the recharging procedure, let us assume the presence of fixed ground recharging stations. This study does not rely on specific wireless technologies, and for ease of exposition, let us assume that IEEE 802.11n wireless transceivers are used. The objective of this Chapter is to investigate energy-efficient distributed deployment strategies, so that RUs are able to self-organize into an aerial mesh structure while maximizing the EU devices connected to the mesh (shown in Figure 1). Let us recall the Connectivity Index (CI) defined in 6.1. More formally, let's say that two EU devices u_1 and u_2 are interconnected at time t whether there exists a path inside the aerial mesh from u_1 to u_2 passing through RUs. Let $E(t)$ the set of interconnected EU devices at time t. Let us use the Connectivity Index at time t as $CI(t) = \frac{|E(t)|}{N}$. In practice, the CI metric quantifies the ability of the aerial mesh to re-establish the connectivity on the post-disaster scenarios.

7.2 Distributed Swarm Mobility

In the method that will be described, distributed mobility of RUs is subject to three rules: (i) it must preserve the network connectivity of the aerial mesh, i.e. no

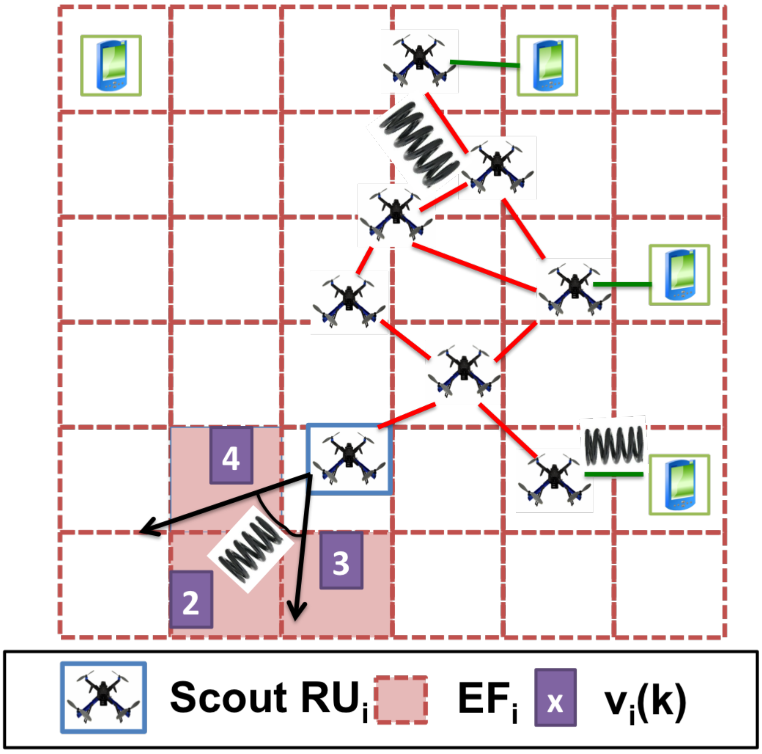


Figure 7.3: The aerial mesh with virtual springs and scout RNs [173].

partitions should occur, (*ii*) it must satisfy Quality of Service (QoS) over the mesh-to-mesh (MtM) and over the mesh-to-EU links (MtE), (*iii*) it must guarantee the CI metric (discussed earlier). Due to the challenges posed by 3D mobility [176, 177], here will be considered a simplified scenario in which all RUs move at the same known and fixed altitude (h) reducing the network deployment to a 2D case. However, in Section 7.5, it will be investigated the impact of the h parameter on system performance.

The space is divided into a grid of GxG cells (Figure 7.3), and each RU is aware of its current cell through a GPS interface.
The distributed swarm mobility algorithm relies on the Virtual Spring Mesh scheme described in [130] and on the extension decribed in Chapter 6 for the distributed creation and maintenance of dynamic (terrestrial) backbones.
The method that we already saw before, associates with each wireless link a virtual spring force characterized by a natural length l_0 and a stiffness constant k and acting according to the Hooke's law:

$$\vec{F} = -k \cdot (\vec{x} - l_0) \tag{7.1}$$

where $\vec{x}$ denotes the spring displacement. As described in Section 6.3, in this study will be considered three types of virtual spring forces: *MtM*, *MtE* and *MtF*. Differently from the one defined in Section 6.3, the end-points of the MtF virtual spring are the RU_i and the center of a cell in the 2D grid space. This force will be better explained in Section 7.2.1.

Also in this scenario, it is assumed that each disconnected EU device periodically transmits a HELLO message, with its position and identifier in order to enable its localization from the RUs. Similarly, while flying, each RU_i broadcasts a BEACON message on the Common Control Channel (CCC) in the 2.4 GhZ band every T_f intervals containing its id, position, the number of EU devices currently connected to (n^i_{EU}), and the set of neighbor RUs of RU_i ($Neigh_i$).
Conversely, here we use the same definition of the Link Budget (LB) in Equation 6.27 and the spring displacement $\delta = (\vec{x} - l_0)$ in Equation 6.29.

The propagation decay exponent α used in our scenario is equal to 2. Based on its connection with other neighboring RUs, and discovered EU devices, multiple virtual forces $\vec{F_0}$, $\vec{F_1}$ ... $\vec{F_n}$ act on each RU. Every T_{DEC} intervals, each RU_i computes the resultant force $\vec{F}$ defined as $\vec{F} = \sum_{i=0}^{n} \vec{F_i}$, and moves in the direction indicated by $\vec{R}$, with constant speed. To avoid fluctuations we define two additional mobility mechanisms: (*i*) a minimum threshold R_{thr} is introduced, so that an RU will change its position only if $|\vec{F}| > F_{thr}$, and (*ii*) before moving on the direction indicated by $\vec{F}$, each RU verifies through the CI metric whether any breakage of MtM link will occur, and if so, it does not update its position.

7.2.1 Exploration method

This phase allows the RUs to locate the EU isolated devices and connect them to the mesh. While a too weak exploration might translate into suboptimal CI performance, a too strong mobility of RUs, dedicated to the exploration, may lead to the partitioning of the aerial mesh. To address both the connectivity and performance issues, our mobility scheme delegates the exploration phase to a dynamically selected set of special nodes, called Scout RUs, which are placed on the edges of the aerial mesh. A virtual spring force (with length equal to 0, and thus attractive only) is used to drive RUs towards less explored cells of the grid. Differently form the method explain in Section 6.3.2, here the exploration is made by mean of an exploration grid that is used to avoid to explore areas already visited. More specifically, the exploration procedure involves four steps:

- *Scout selection.* Every T_{SCOUT} seconds, each RU_i checks its position compared to its neighbour RUs. In case RU_i does not detect any other RU in its visibility zone (defined as a cone with sweep angle of θ centred in RU_i), then it self-elect as scout node, with probability equal to p_{SCOUT}.

- *Direction selection.* Each RU_i keeps statistics about the number of times it has visited each cell j of the scenario (i.e. $v_i(j)$). The cell value $v_i(j)$ is incremented by 1 each time the RU stays in the cell j for a minimum duration interval (equal to 10 second in the experiments). Let us consider the exploration frontier of RU_i (EF_i) as the set of cells located inside the intersection of the square centered on the current cell, and of side equal to $2 \cdot hop + 1$, and the visibility zone of RU_i. In Figure 7.3, it is shown an example of the EF zone with hop=1 and θ=30. In practice, the parameter hop defines the horizon of the exploration phase, i.e. how far RU_i is looking at when deciding its next position. Then, RU_i selects the cell j that has been less visited in EF_i, i.e. $j = argmin\ v_i(k)|\forall k \in EF_i$.

- *Force computation.* Once the next cell j has been selected, a virtual MtF spring is defined between RU_i and the center of cell j. The spring displacement is fixed and equal to a reference value LB_{min}. On the other side, the stiffness constant k_{MtF} is adjusted on the basis of the amount of exploration performed on cell j, i.e.:

$$k_{MtF} = \left(1 - \frac{v_i(j)}{v_i^{Max}}\right)^{v_i^{min}+1} \tag{7.2}$$

where v_i^{Max} and v_i^{min} are respectively the maximum and minimum cell values for RU_i, considering all cells of the scenario, i.e. $v_i^{Max} = max(v_i(j)), \forall j \in G \times G$

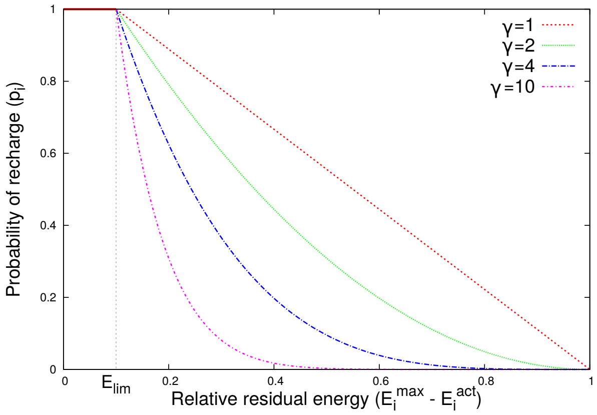


Figure 7.4: The p_i^{sr} function for different values of γ_i.

and $v_i^{min} = min(v_i(j)), \forall j \in G \times G$. Through Equation 7.2, is expressed the intuitive concept that RU_i is more attracted by those cells that has been explored less than the average and that the exploration force decreases when the discovery ratio of the full scenario (i.e. v^{min}) increases.

7.3 Distributed Charging Scheduling

Now will be described the distributed algorithm to let RUs autonomously decide when to go to recharge. The algorithm must satisfy the following requirements: (*i*) to attempt to preserve the CI by giving precedence to RUs whose departure will not cause the partitioning of the aerial mesh, and (*ii*) to account for the recharging need of each RU based on its residual battery energy. To fulfil these requirements a probabilistic approach is proposed, in which each RU_i is assigned a probability p_i of going to recharge, defined as:

$$p_i(E_i^{act}, E_i^{max}) = \left(\frac{E_i^{max} - E_i^{act}}{E_i^{max}} \right)^{\gamma_i} \tag{7.3}$$

where E_i^{max} and E_i^{act} are, respectively, the maximum and the actual residual battery capacity of RU_i. The exponent $\gamma_i \geq 1$ modifies the probability p_i by reflecting the cost incurred by the departure of RU_i, as discussed below. To avoid the case of a RU having insufficient energy to reach the nearest charging station, we introduce a threshold value E^{lim} on the residual energy, below which the RU *must* go to recharge. Hence, the probability p_i is modified as follows:

$$p_i^{sr} = \begin{cases} p_i(E_i^{act} - E^{lim}, E_i^{max} - E^{lim}) & \text{if } E_i^{act} > E^{lim} \\ 1 & \text{otherwise} \end{cases} \tag{7.4}$$

In Figure 7.4 we plot the p_i^{sr} values, for different configuration of γ_i. It is easy to notice that p_i^{sr} decreases with the current state of charge of RU_i (E_i^{act}), and that for the same value of E_i^{act}, the probability is lower for higher values of γ_i. This latter is defined as follows:

$$\gamma_i = \gamma_i^{critical} + \gamma_i^{EU} + 1 \tag{7.5}$$

The value of $\gamma_i^{critical}$ reflects the connectivity degree of RU_i, and is defined as follows:

$$\gamma_i^{critical} = \gamma^{maxC} \cdot (1 - e^{-\#numclusters}) \tag{7.6}$$

Here, γ^{maxC} bounds the maximum value of $\gamma_i^{critical}$ (equal to 12 in the experiments) and $\#numclusters$ is the estimation of the number of the potential clusters that might be formed if the RU_i disconnects from the mesh. This value is computed by determining the rank of the matrix A^{Neigh_i}, where A is the adjacency matrix of RU_i. The $\#numclusters$ value takes into account only the 2-hop neighbors of RU_i, and thus, it provides a local approximation of disconnected components. The second component of Equation 7.5 (γ_i^{EU}), is specific to the MtE link, and reflects the actual EU devices currently interconnected to the node RU_i:

$$\gamma_i^{EU} = \begin{cases} \gamma^{maxEU} \cdot k_{MtE} & \text{if } n_{EU}^i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{7.7}$$

where γ^{maxEU} bounds the maximum value of γ_i^{EU} (equal to 3 in the experiments) and k_{MtE} is the stiffness constant of the spring for MtE link defined by Equation 6.30. Basically, through (7.6) and (7.7), the probability p_i^{sr} is discounted by considering: (*i*) the potential clusters that might occur, and (*ii*) the isolated EU devices that might be left.

Every $T_{recharge}$ seconds, each RU_i decides with probability p_i^{sr} whether to go to recharge or not. We assume a linear charging model, i.e. the charging time CT_i is computed as: $CT_i = CT^{r\_max} \cdot \left(\frac{E_i^{max} - E_i^{act}}{E_i^{max}}\right)$, where CT^{r_max} is the time required by a full battery recharge. After charging is completed, the RU resumes its operation.

7.4 3D Scenario Modelling

In this Section, we detail how the 3D scenario has been modeled through the OMNeT++ simulation tool. The scenarion is populated by rectangular-shape buildings with varying dimensions on the 3^{rd} axes (the height). Although in the evaluation (Section 7.5) are considered authomatic generated scenarios, realistic scenarios can also be modeled through the proposed tool, by importing the

XML maps provided by *OpenStreetMaps* [178] with building information. Modeling the wireless propagation effects in 3D environments is highly-challenging due to the complications of creating accurate ray-tracing models in complex scenarios. Hence it will be used a simplified 3D propagation model that takes into account the attenuation effect caused by buildings on the line of sight from the transmitter to the receiver. Although this model cannot capture the complex reflection/diffraction phenomena, its suitability to guarantee good approximation for large-scale network simulations has been demonstrated in [179]. More specifically, the algorithm works in three steps:

- First, it is considered the *Line-of-Sight* (LOS), i.e. the straight line between the sender and the receiver (which can be two RUs or a RU and an EU device).
- Then, it is determined all the points in which the straight line collides with an obstacle.
- Finally, it is applied an attenuation factor to the received signal, based on the the length of the indoor path and on the number of intersected outdoor walls.

The pathloss (in dB) is modeled as $PL[d] = \alpha \cdot 10 \cdot log_{10}(d(i,j)) + \beta$, where $d(i,j)$ is the 3D distance between node i to node j, α is the propagation exponent (fixed to 2 in our case, i.e. a free-space model is considered) and β is a zero-mean Gaussian distributed random variable with standard deviation σ (in dB). By computing the intersection points between the LOS and the faces of a building b, we derive the length d'_b, in meters, in which the signal travels indoor. The signal attenuation is modelled as follows:

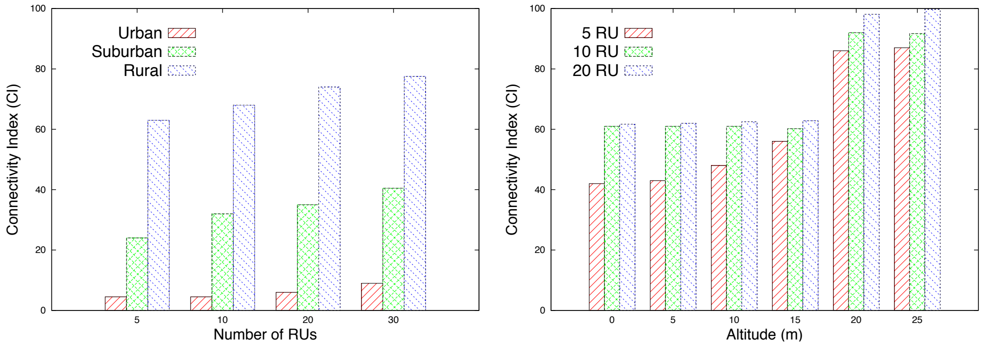
$$S_b = 2 \cdot k + d'_b \cdot \eta \tag{7.8}$$

where k is the attenuation factor due to the outside walls, and η is the attenuation related to indoor propagation through dry walls, furniture etc. The parameters k and η are set to 20dB and 1dB/mt, based on literature surveys. Finally, given B the set of buildings of the scenario, it is computed the power of received signal as $P_{rx} = P_{tx} - PL[d] - \sum_{b=0}^{B} S_b$.

In order to further reduce the computation, is considered the cell-grid world depicted in Figure 7.3, and hence is reduced B to $B' \subseteq B$, i.e. the set of buildings that are on the cells intersected by the LOS between nodes i and j.

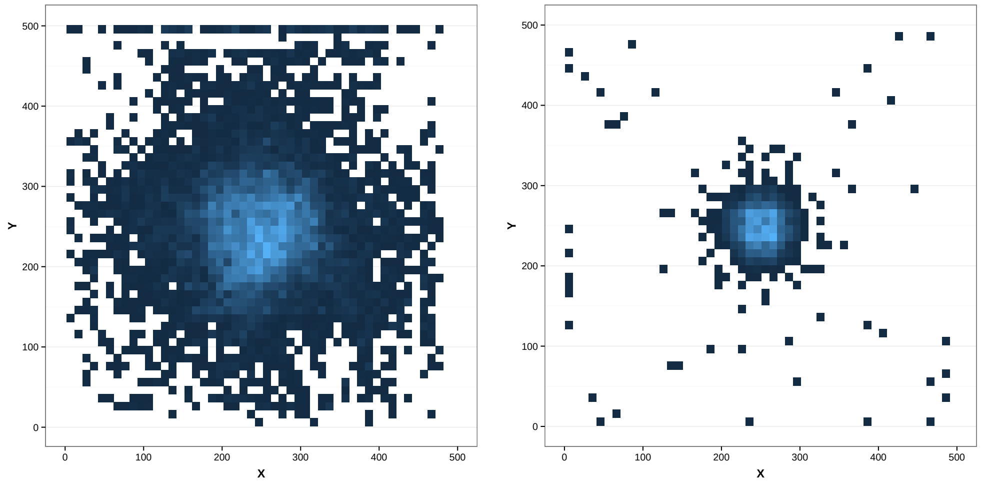
7.5 Charging scheduling performance

In this Section are evaluated the performance of the distributed mobility scheme (Section 7.2) and of the charging scheduling algorithm (Section 7.3) inside 3D



(a) (b)

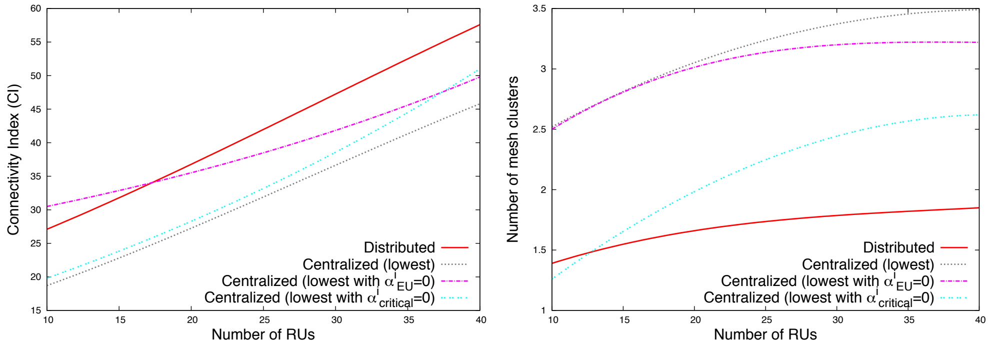
Figure 7.5: The CI metric as a function of the number of RUs and of the altitude h is shown in Figure 7.5(a) and 7.5(b), respectively.



(a) (b)

Figure 7.6: The average positioning of the RUs for $h = 25$ is shown in Figure 7.6(a). The average positioning of the RUs for $h = 0$ is shown in Figure 7.6(b).

scenarios modelled according the propagation model described in Section 7.4. It is considered a scenario of 500m x 500m, with buildings of height 30m placed at random cells of a Manhattan grid scenario. 50 EU devices are randomly distributed at indoor/outdoor locations. In Figures 7.5(a) and 7.5(b) is depicted the CI metric (Section III) when varying the number of RUs and the altitude from ground. More specifically, in Figure 7.5(a) is plotted the CI values against the number of RUs, for three increasing building density configurations, namely Rural, Suburban, and Urban. Clearly, increasing the number of RUs has a positive effect regardless of the scenario considered, since it translates into the possibility



(a) (b)

Figure 7.7: The average CI metric and the number of originated clusters for the four scheduling charging algorithms are depicted in Figures 7.7(a) and 7.7(b), respectively.

to enlarge the exploration and coverage range of the aerial mesh network. Figure 7.5(a) demonstrates the fact that the attenuation caused by buildings might have a significant impact on the quality of MtM links, since RUs are forced to stay closer to guarantee the request QoS expressed in terms of minimum Link Budget LB_{req} (Equation 6.27). Also, frequent partitions may occur in the aerial mesh, caused by mobility of RUs. In Figure 7.5(b) is further investigated this issue, by varying the altitude h from ground of the aerial mesh. It is considered a modified version of the Suburban scenario (medium building density) with building heights uniformly chosen in the range [10, 20] m. This result clearly demonstrates that the CI is affected by the altitude h more than the number of RUs. Indeed, CI value significantly increases when $h \geq$ 15m, i.e. when RUs are, on average, just over the buildings clutter height. The case with h=0 corresponds to the case in which mobile nodes move on the ground, and thus the mobility algorithm described Section 7.2 is implemented by robot equipped with wheels. Thus, aerial communication performs much better than ground communication, and the CI with 5 RUs deployed at 20 m is much higher than the CI with 30 RUs on the ground (h=0).

In Figures 7.6(a) and 7.6(b), is plottd the coverage area with varying heights of 0m and 25m, when using the same number of RUs (i.e. 15). All the RUs are injected at the center of the scenario at the start of the simulation. The blue color gradients depict the probability that RUs will stop at a given position[1]. We see that RUs moving in the air are able to cover much wider area than terrestrial robots, while guaranteeing the same quality over the mesh link.

[1] Lighter colors indicate more frequent areas, darker colors indicate less frequent area, white color indicates areas with probability lower than 0.01.

In Figures 7.7(a) and 7.7(b) is evaluated the effectiveness of the distributed charging scheme in guaranteeing a persistent coverage over the emergency area through four scheduling approaches: (*i*) Distributed, which corresponds to our solution described in Section 7.3, (*ii*) Centralized (lowest), which corresponds to a centralized algorithm in which at each $T_{recharge}$ interval the RU_i with lowest residual energy E_i^{act} is selected for recharging, (*ii*) Centralized (lowest with $\gamma_i^{EU} = 0$) which works as the previous, but it chooses the lowest residual energy RU_i with $\gamma_i^{EU} = 0$, i.e. not connected to EU devices, (*ii*) Centralized (lowest with $\gamma_i^{critical} = 0$) which works as the previous, but it chooses the lowest residual energy RU_i with $\gamma_i^{critical} = 0$, i.e. not originating mesh partitions after its departure. Figures 7.7(b) shows the average CI for the four algorithms when varying the number of available RUs. This result demonstrates that selecting RUs on the basis of the energy factor only, like the Centralized (lowest)method, can cause frequent partitions within the mesh network. This is also confirmed by Figure 7.7(b) where is depicted the average number of isolated clusters that are originated during the simulation. The Centralized (lowest with $\gamma_i^{EU} = 0$) scheme attempts to maximize the number of connected EU devices on the MtE link, and for this reason, it outperforms the other two centralized schemes in terms of CI (Figure 7.7(b)). Vice versa, the Centralized (lowest with $\gamma_i^{critical} = 0$) scheme attempts to preserve connectivity among RUs on the MtM links, and for this reason it creates a reduced number of clusters compared to the other two centralized algorithms (Figure 7.7(b)). Our distributed algorithm takes into account both the connectivity of the aerial mesh and to EU ground devices through the two components of γ_i (Equations 7.5 and 7.7), to give the best performance both in terms of CI and of number of cluster partitions.

Chapter 8

TVWS frequencies for rescue missions

The proliferation of new pervasive mobile services and applications has increased the demand of spectrum bandwidth for high speed connectivity. For this reason and given the scarcity of radio spectrum resources, the research focus has shifted on how to find better way to utilize the radio frequencies already allocated. One of the most interesting solution is represented by *TV White Spaces* (TVWS) frequencies that are left free or are underutilized by the TV broadcasters after the worldwide digital TV switch-over [180].
Due to the unique characteristics of the TV channels' frequencies (i.e. VHF/UHF), like the good signal propagation in shadowed scenario and the low transmitting power consumption, the TV White Spaces are becoming an attractive technology for emergency operations [5, 181, 4, 182, 183].

Several national regulator, like FCC [6] and Ofcom [7], have already issued specifications and requirements to enable unlicensed devices to operate in the TVWS. According to these specifications, geolocation database (GLDB) is the reference technique to protect the operations of the licensed TV transmitters, also referred to as Primary Users (PUs) [6] [7]. Each time a secondary device would like to access the TV channels opportunistically, it must first query the GLDB in order to retrieve the list of TVWS available at its location. Clearly, the viability of this technique strongly depends on the accuracy of the TV channel availability estimation, which is typically performed on the basis of extensive propagation modelling, transmitters data and regulators information [180]. However, recent studies indicate that, depending on the level of the environment complexity, the channels availability estimation can be performed well only with a significant amount of auxiliary information and fine-tuning [184]. This is the case of TVWS estimation at the TV cell edge, where the TV signal is low, and hence the GLDB might fail in accurately identifying the contour region. At the same time, it is worth noting that most of GLDBs work by assuming a bi-dimensional scenario modeling, i.e. the environment is divided into cells of fixed size, and the TV availability is computed on the basis of signal reception at the rooftop level. While this approach maximally protects the operations of licensed users, it might likely lead to sub-optimal re-utilization of TV frequency resources in complex scenarios, where there might be spectrum opportunities also at different heights for the same cell. For these reasons, 3D TVWS spectrum database and sharing models are being progressively proposed and investigated [185].

In this Chapter, it is proposed three preliminary yet important contributions on the practical utilization of UASVs for TVWS deployment scenarios. First, the design of a distributed mobility algorithm to enable the UASVs to explore an unknown scenario in terms of space and frequency. The swarm mobility algorithm takes into account channel-aware metrics, like sensing correlation distance, in order to maximize the detection accuracy while considering the (limited) flying autonomy of each device. Second, it is described how sensing reports are collected and aggregated in order to build a shadowing map of the scenario, dynamically adapting the spatial resolution of the spectrum database. Third, the performance of proposed technique is evaluated through OMNeT++ simulator, and demonstrated its effectiveness in terms of sensing coverage and accuracy.

8.1 Enhancing TVWS Database with unmanned aerial vehicles

In this Section, it is introduced the method of enhancing the accuracy and spatial resolution of existing GLDBs thanks to the utilization of a fleet of flying sensing units, also called *Unmanned Aerial Scanning Vehicles* (UASVs).

Again, in this scenario will be used the STEM-Node as reference architecture. In this case the UASVs have only one role, i.e. the *scanning* one and hence the *Roles* decision module is not present, as depicted in Figure 8.1. A key role is performed by the *Network Actuator* that is in charge of scanning the TVWS frequencies in order to be aware of the frequencies occupancy. The UASVs, in fact, are able to autonomously explore an unknown environment by sensing the TV radio frequencies, to self-configure its own current location and scanning channel and to share results in order to build a 3D shadowing model. The *Cooperation Manager* and the *Database Knowledge* have the task of spreading the scan information among the UAVs and of analysing the scan results in order to estimate the shadowing map. The results of the cooperative sensing phase from the UASVs can then be used for two goals: (*i*) fine tuning of the existing GLDB propagation model over specific map regions (e.g. TV cells' edge) based on the fading con-

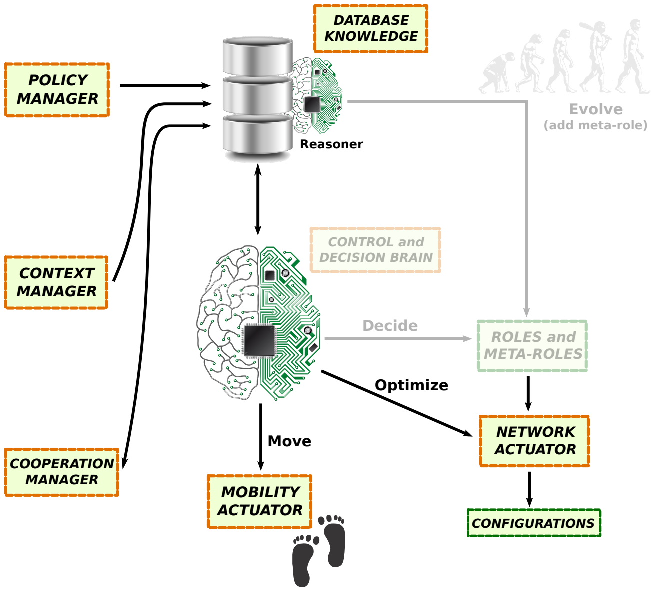


Figure 8.1: The Stem-Node architecture used for designing the UASV

ditions estimated by the UASVs or (*ii*) population of a 3D TVWS database for a fine-grained opportunistic frequency reuse, i.e. taking also in consideration the height of the radio transmissions.

8.1.1 Environment model

Let us define a grid scenario made by $C = L \times H$ cells of dimension d_C. K is the set of TV channels to scan. It is assumed a log-normal shadowing as reference path loss model for the transmission over TVWS, i.e. [186]:

$$PL_\alpha(d)\,[dB] = \overline{PL}_\alpha(d) + \mathcal{X}_\sigma \tag{8.1}$$

$$\overline{PL}_\alpha(d) = \overline{PL}(d_0) + 10 \cdot \alpha \cdot log\left(\frac{d}{d_0}\right) \tag{8.2}$$

where $\overline{PL}_\alpha(d)$ is the log-distance path loss with α as path-loss exponent and $\mathcal{X}_\sigma$ is a zero-mean Gaussian distributed random variable (in dB) with σ as standard deviation (in dB).

The operation of spectrum scanning in a log-normal shadowing path-loss may lead to correlated shadowing scans. This will degrade the performance of those scan reports for points that experience similar shadowing effect due to their proximity. The spatial correlation coefficient $\rho_{i,k}$ between two different radio scans i, k is defined as follow [186]:

$$\rho_{i,k} = e^{-d_{i,k}/D} \tag{8.3}$$

where $d_{i,k}$ is the distance between the two scans i, k and D is called *decorrelation distance*. From empirical measurement [187] the decorrelation distance can be set to $D = 8m$ for urban environment and $D = 500m$ for suburban environment.

Using the log-normal shadowing as path-loss model, a free space environment will have $\alpha = 2$, and an urban environment will have $\alpha = 5$ [186]. With these preconditions, let us introduce a parametric definition for the decorrelation distance D in function of the α at coordinate (x, y), i.e.:

$$D_\alpha[m] = \begin{cases} 500, & \text{if } \alpha \leq 2 \\ 8, & \text{if } \alpha \geq 5 \\ ((500 - 8) * e^{-2*(\alpha-2)}) + 8, & \text{otherwise} \end{cases} \tag{8.4}$$

Using the Equation 8.4 it is modelled the exponential decay of the decorrelation distance related to the environment shadowing. Each point (x, y) of the scenario has its own characteristic α and hence it will be used interchangeably D_α and $D_{(x,y)}$.

8.1.2 Scenario modelling

Let T be the set of transmitting towers on the TV radio channels inside the scenario. For each tower $t_i \in T$ let us assume to know the position p_{t_i}, the transmitting TV channel k_{t_i} and the transmitting power Ptx_{t_i} (these information are of public domain). $T_{k_i} \subseteq T$ is the sub-set of TV towers transmitting on channel $k_i \in K$.

Let us define the set U, of cardinality $|U| = N_{UASV}$, as the fleet of UASVs that is in charge of exploring the scenario, scanning the TV channels and reporting the occupancy status of all the channels $k_i \in K$. Each UASV $u_i \in U$, while exploring the scenario, generates point reports that are formed as a pair $\langle(x, y), report\rangle$ where $1 \leq x \leq (d_C \cdot L)$, $1 \leq y \leq (d_C \cdot H)$ and $report \in \{0, 1\}$. Each UASV is equipped with two radio interfaces. One radio is an IEEE 802.11-compliant device used to communicate with the others UASVs, while the other radio is a *Software Defined Radio* (SDR) that is able to opportunely tune its listening frequency on each $k_i \in K$ of the TV band.

8.1.3 Problem formulation

In order to populate the 3D TVWS database, the aim of this Chapter is to find $\mathcal{F} : K, C \rightarrow \{0, 1\}$ such that for each channel $k_i \in K$ and for each cell $C_{n,m}$, for $1 \leq n \leq L$ and $1 \leq m \leq H$, the function $\mathcal{F}(k_i, C_{n,m})$ returns the occupancy of the channel k_i into the cell $C_{n,m}$.

To better analyse the proposed function, now will be defined an analytical frequency occupancy solution that will be used next as a comparison model. Given the path-loss model in Equation 8.1, let $\mathcal{F}_a(k_i, C_{n,m}) : K, C \rightarrow \{0, 1\}$ be the analytical function that calculates if $C_{n,m}$ is covered by any transmitting station over channel k_i:

$$\mathcal{F}_a(k_i, C_{n,m}) = \begin{cases} 1, & \text{if } \sum_{\substack{\forall (x,y) \in C_{n,m} \\ \forall t_i \in T_{k_i}}} Prx((x,y), t_i) > \lambda \\ 0, & \text{otherwise} \end{cases} \tag{8.5}$$

$$Prx((x,y), t_i) = Ptx_{t_i} - (\overline{PL}_\alpha(d(p_t, (x,y))) - \beta\sigma_{(x,y)}) \tag{8.6}$$

where $d(p_1, p_2)$ is the distance between the two points p_1 and p_2, λ is the signal power threshold (in dBm) to mark the channel as busy, $\sigma_{(x,y)}$ is the standard deviation that specify the path-loss model at the point (x, y) and β is a system parameter that indicates the conservative behaviour of the calculated $Prx((x, y), t_i)$. The β parameter defines the coverage radius of the transmitting towers: for $\beta = 0$ the probability that the signal of the transmitting tower t_i reaches a point outside the coverage area defined by $Prx((x, y), t_i) > \lambda$ is 0.5, for $\beta = 1$ the probability is 0.16, for $\beta = 3$ the probability is 0.001, and so on (*three-sigma rule*). Increasing the value of β yields to a conservative behaviour of the analytical model, since it would increase the radius of the transmitting tower; however this strategy could be too much conservative reducing drastically the reuse of the TVWS channels.

8.2 Scan Algorithm

In this Section, it is described the exploring strategy adopted by the UASVs to scan the TVWS channels over the target scenario. Each UASV is provided with an autonomous mobility system and frequency configuration capabilities. To optimize the scanning procedure, cooperation among the UASVs is needed to avoid replicated actions, which might originate when exploring part of the scenario already covered by others UASVs. At the same time, this cooperation mechanism must be carefully designed to be energy-efficient in order to avoid a useless drain of the battery power.

8.2.1 Field Force Based Movement

To fulfil these requirements, a field-based self-coordination method [188] is adopted, that is used for both exploring the scenario (*attractive field*) guiding the spectrum scanning activities and avoiding UASVs self-collisions (*repulsive field*).

Each UASV $u_i \in U$ maintains a list of point-reports, that is indicated as $P_{u_i} = \{p_{u_i,1}, p_{u_i,2}, \ldots\}$, where $p_{u_i,j} = \langle (x,y), report\rangle_i^j$ for $1 \leq j \leq |P_{u_i}|$. While exploring the scenario, each UASV u_i broadcasts a beacon message with its own P_{u_i} every t_{coop} seconds. Whenever an UASV u_j receives a packet P_{u_i} from UASV u_i, it updates its report list $P_{u_j} = P_{u_j} \cup P_{u_i}$. As a results, all the point-reports are gradually disseminated within the swarm. The mobility control algorithm of each UASV is based on the point field-force model. More specifically, let us assume that multiple field forces can apply on each UASV u_i, i.e. $\vec{F}_{i,1}, \vec{F}_{i,2}, \ldots, \vec{F}_{i,f_i}$. Let $\vec{F}_i = \sum_{j=1}^{f_i} \vec{F}_{i,j}$ be the total force acting on the u_i. The force $\vec{F}_i$ will define the acceleration vector for the u_i. There are three possible types of field forces:

- $\vec{F}_i^R(p_{u_i,j})$ - **Repulsive forces** from each point-report $p_{u_i,j}$: these forces are used to prevent the UASVs u_i from moving on areas already scanned and is defined as follow:

  $$\vec{F}_i^R(p_{u_i,j}) = \vec{u} \cdot w_R \cdot e^{-(\xi_{D_{(x,y)}} \cdot d(u_i, p_{u_i,j})^2)} \tag{8.7}$$

  where $\vec{u}$ is the unit vector which has the direction from $p_{u_i,j}$ towards u_i; w_R is a system parameter that defines the maximum magnitude of the repulsive force; $\xi_{D_{(x,y)}}$ is a parameter depending on the value of the decorrelation distance $D_{(x,y)}$ (where (x,y) are the coordinates of the point report $p_{u_i,j}$) and is defined as follow:

  $$\xi_{D_{(x,y)}} = \frac{ln(1/\epsilon)}{D_{(x,y)}^2} \tag{8.8}$$

  where ϵ is a system parameter. The $\xi_{D_{(x,y)}}$ parameter is designed in such a way that it can control the slope of the bell-shape field-force function defined by Equation 8.7, so that at a distance of $D_{(x,y)}$ the force is equal to $\epsilon \cdot w_R$ (see Figure 8.2).

- $\vec{F}_i^U(u_j)$ - **Repulsive forces** from the others UASVs: this force will keep the UASVs away from each other avoiding both collisions and shadowing correlation in the scanning procedure. Since the UASVs are moving around the scenario, this force is modelled like in Equation 8.7 but it is also time dependent. Let Δt_{u_j} be the time elapsed from the last message reception from UASV u_j:

  $$\vec{F}_i^U(u_j) = \vec{u} \cdot w_R \cdot e^{-(\xi_{D_{(x,y)}} \cdot d(u_i,u_j)^2 + \Delta t_{u_j} \cdot \tau)} \tag{8.9}$$

  where τ is a system parameter and define the time-decay factor.

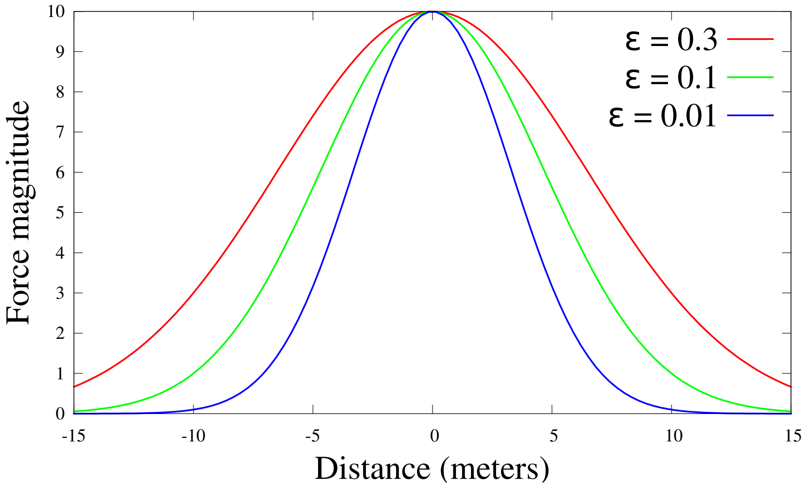


Figure 8.2: Effects of the parameter ϵ on the magnitude of the force defined in Equation 8.7 with $w_R = 10$ and $D_{(x,y)} = 10m$

- $\vec{F}_i^E(x,y)$ - **Attractive force** towards the exploring point: in order to explore the TV frequency usage, each UASV must move around the scenario. The field force is defined as follow:

$$\vec{F}_i^E(x,y) = \vec{u} \cdot w_E \cdot \left(1 - e^{-(\xi_{D_{(x,y)}} \cdot d(u_i,(x_e,y_e))^2)}\right) \tag{8.10}$$

  where w_E is a system parameter that defines the maximum magnitude of the force. The point (x_e, y_e) is calculated by Algorithm 6 where the UASV u_i chooses n_R random points in the scenario and selects the point p_r that minimize the average of the magnitude of the total forces acting on the path from (x,y) to (x_r, y_r).

Algorithm 6: Calculate the exploring point (x_e, y_e)

```
Ensure: UASV $u_i$ has coordinate $(x,y)$
  $P \leftarrow \{p_1, p_2, \ldots, p_{n_R}\}$ random points $[p_r : (x_r, y_r)]$
  for $\forall p_r \in P$ do
    $\overline{\vec{F}_i^R}(p_r) \leftarrow$ average magnitude of $p_{u_i,MAX}$ on the path that goes from $(x,y)$ to $(x_r, y_r)$
  end for
  $p_{min} \leftarrow argmin_{p_r \in P} \overline{\vec{F}_i^R}(p_r)$
  return $(x_{min}, y_{min})$
```

In Figure 8.3 is shown an example scenario where is depicted the generated field forces that act on a UASV.

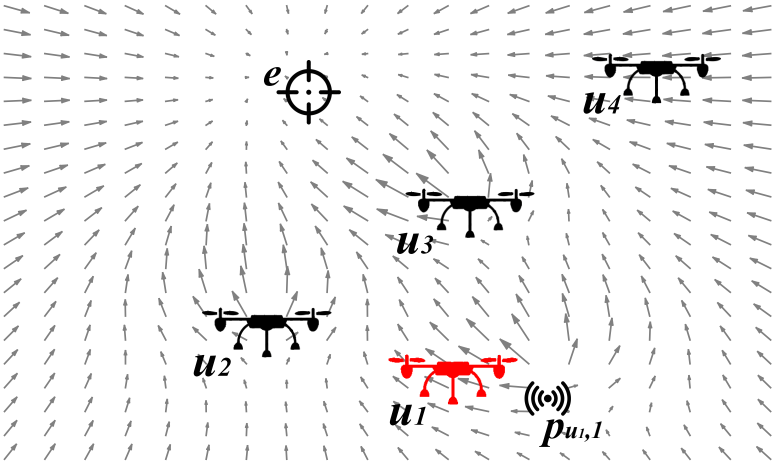


Figure 8.3: Example of the field forces felt by the UASV u_1 drawn in red. The scan point $p_{u_1,1}$ generates a force F_i^R; the point e is an attraction point that generates F_1^E; u_2, u_3 and u_4 generate the repulsive force F_1^U (the u_4's force is almost 0 due to the time-decay factor)

8.2.2 Creating the scanning reports

While moving around the scenario, the UASVs must scan the TVWS frequencies in order to find out the current channel availability for the secondary users. As explained in Section 8.1, the quality of a scan reports depends on both the number of sensing samples and the spatial correlation among the samples. To address this issue, is used a probabilistic method that enables the UASVs to increase the chance of executing spectrum scanning operations over points-decorrelated regions, while limiting the exploration phase over points-correlated regions.

Every t_{scan} seconds each UASV u_i performs with probability P_{scan} a TV channel scan. Let us assume that this operation must be performed while being in a static position because each channel scan-report must be associated to specific geographic coordinates. Therefore every time an UASV wants to start the scan procedure it must stop moving. Moreover, at each scan operation the UASV will sense all the channels $k_i \in K$. The required time is defined as T_K and includes: the time of scanning that is typically in the order of few μs when using energy detector techniques [189] and the time needed to stabilize the positioning system. For an UASV u_i, the probability P_{scan} is defined as follow:

$$P_{scan}(u_i) = 1 - \frac{|\vec{F}_i^R(p_{u_i,MAX})|}{w_R} \tag{8.11}$$

$$p_{u_i,MAX} = argmax_{\forall p_{u_i,j} \in P_{u_i}} |\vec{F}_i^R(p_{u_i,j})| \tag{8.12}$$

The probability $P_{scan}(u_i)$ is be zero over the scan point $p_{u_i,MAX}$ and $1-\epsilon$ at distance $D_{(x,y)}$, where (x,y) are the coordinates of $p_{u_i,MAX}$. In this way, the UASV will have higher probability to scan the spectrum at decorrelated locations of the same area.

8.2.3 Creating the shadowing map

In our model, the decorrelation distance $D_{(x,y)}$ depends on the value of α (propagation decay) that characterizes the point (x,y) (see Section 8.1.1). However, the values of α are unknown when the UASVs start exploring the scenario. To overcome this problem, the UASVs will estimate the current signal propagation and shadowing conditions relaying on the known scenario data, like the positions of the TV towers and their transmitting power. As preliminary step before building the shadowing map, each UASV u_i uses a default value α^0, as initialization value, such that $\alpha_{(x,y)} = \alpha^0$, $\forall(x,y)$. Every time the point-reports set P_{u_i} is updated with a new scan $p_{u_i,j}$, the shadowing map is updated accordingly.
There could be two cases based on the report of the scan: busy or free. The update procedure is described in the Algorithm 7, where the action radius of the scan point $p_{u_i,j}$ is defined by the value of $D_{(x_j,y_j)}$. For clarity reasons, we omitted the sensed channel $k \in K$ in Algorithm 7. The Equations in Algorithm 7 are derived from the path-loss model described in Section 8.1.1. If the scan report is *free* (line 5 in Algorithm 7) then the worst-case assumption is used, i.e. the power received at the UASV u_i is assumed as being just below the sensing threshold λ.

Algorithm 7: Update of the shadow map for the UASV u_i

```
Ensure: At the beginning $\alpha_{(x,y)} = \alpha^0$ and $c_{(x,y)} = 0$
Require: Scan point $p_{u_i,j} = \langle (x_j,y_j), report_j \rangle_i^j$ available
1: for $\forall(x_u,y_u)$ such that $d((x_j,y_j),(x_u,y_u)) \leq D_{(x_j,y_j)}$ do
2:     if $report_j$ is $busy$ then
3:         $\alpha_j = \frac{Ptx_t - \overline{PL}(d_0) - Prx_{u_i}}{10 \cdot log\left(\frac{d}{d_0}\right)}$
4:     else
5:         $\alpha_j = \frac{Ptx_t - \overline{PL}(d_0) - \lambda}{10 \cdot log\left(\frac{d}{d_0}\right)}$
6:     end if
7:     $c_{(x_u,y_u)} = c_{(x_u,y_u)} + 1$
8:     $\alpha_{(x_u,y_u)} = \alpha_{(x_u,y_u)} + \frac{\alpha_j - \alpha_{(x_u,y_u)}}{c_{(x_u,y_u)}}$
9: end for
```

It is easy to notice in the pseudo-code above that the shadowing map is always kept update while scanning and it will improve in accuracy with the number of performed sensing actions.

8.2.4 Combining the final reports

After that sensing operations have been completed, all the UASVs return to the base station and all the scan reports of each UASV are elaborated by a *Central Unit* (CU). Let L_{rep} be the set of all the reports made by the UASVs. Each scan report is associated to a specific cell of the scenario, and the final decisions are performed by the CU based on the aggregate set $S(C_{n,m}) : \{\langle (x,y), report \rangle | (x,y) \in C_{n,m}\} \subseteq L_{rep}$. However, it is worth noting that the cell size (i.e. d_C) used during exploration is a system parameter, which might not correspond to the characteristic of the scenario in terms of shadowing values and propagation conditions. For this reason, it is introduced a more accurate scheme for scenario partitioning to identify clusters of cells that can be considered belonging to the same shadow region, i.e. associated to similar propagation conditions. Algorithm 8 illustrates the clustering procedure. Here, are introduced two more system parameters, i.e. α_{DIFF} and D_{MUX}. The first defines the maximum difference of the α values within each cluster, while the second is a parameter defining the maximum size of the cluster.

Algorithm 8: Calculate the clusters set CL

```
$CL \leftarrow \emptyset$
for $\forall C_{(n,m)}$ do
  if $\forall CL_i \in CL \Rightarrow C_{(n,m)} \notin CL_i$ then
    $CL_{new} \leftarrow \{C_{(n,m)}\}$
    for $\forall C_{(x,y)}$ contiguous to any $C_{(w,z)} \in CL_{new}$ do
      if $(abs(\overline{\alpha}_{(n,m)} - \overline{\alpha}_{(x,y)}) \leq \alpha_{DIFF})$ AND
      $(\overline{d}(C_{(n,m)}, C_{(x,y)}) \leq (D_{\overline{\alpha}_{(n,m)}} \cdot D_{MUX}))$ then
        $CL_{new} \leftarrow CL_{new} \cup \{C_{(x,y)}\}$
      end if
    end for
    $CL \leftarrow CL \cup \{CL_{new}\}$
  end if
end for
return $CL$
```

In Algorithm 8, $\overline{\alpha}_{(a,b)}$ is defined as the mean value on the α parameter inside the cell $C_{(a,b)}$ and $\overline{d}(C_i, C_j)$ is the distance between two cells. At the end of the clustering procedure the set $CL : \{CL_1, CL_2, \ldots\}$ is generated. Each cluster CL_i is formed by a subset of contiguous scenario cells. Let $\mathcal{G} : C_{n,m} \rightarrow CL$ be the function that returns the cluster to which a cell belongs, for each cell. The function $S(C_{n,m})$ is then refined as follow: $S(C_{n,m}) : \{\langle (x,y), report\rangle | (x,y) \in \mathcal{G}(C_{n,m})\} \subseteq L_{rep}$ that defines the set of reports belonging to any cell in the cluster $\mathcal{G}(C_{n,m})$. Finally we compute the function $\mathcal{F}(k_i, C_{n,m})$ as follow:

$$\mathcal{F}(k_i, C_{n,m}) = \begin{cases} 1, & \text{if } Busy(k_i, S(C_{n,m})) \geq KoN \\ 0, & \text{otherwise} \end{cases} \tag{8.13}$$

where $Busy(k_i, \cdot)$ is a function that counts the number of busy reports over the channel k_i and KoN (*K-out-of-N*[1]) is a system parameter that indicates the aggregation method: if $KoN = 1$ then the cell is considered busy if there is at least one busy report (*OR*) while if $KoN = |S(C_{n,m})|$ then the cell is considered busy if all the scan report are busy (*AND*).

8.3 Performance Evaluation

The metric used to evaluate the performance are the following:

- **Coverage Index (CI)** - this metric is defined as the ratio of cells having at least one scan report.

$$CI = \frac{\sum_{1\leq n\leq N, 1\leq m\leq M} \overline{S}(C_{n,m})}{L \cdot H} \tag{8.14}$$

  where $\overline{S}(C_{n,m})$ is 1 if $|S(C_{n,m})| \geq 1$ and 0 otherwise.

- **Quality Index (QI)** - this metric is defined as the quality of the final report, for each cell, considering both the number of scan points and the correlation between the scans. These two properties are in logarithmic relation [189] [187]. The QI metric is defined as follow:

$$QI(C_{n,m}) = \frac{ln(D_{\overline{\alpha}_{(n,m)}}) \cdot ln(|S(C_{n,m})|)}{2} + 1 \tag{8.15}$$

  If $\overline{S}(C_{n,m}) = 0$ then $QI(C_{n,m}) = 0$.

- **Accuracy Index (AI)** - this metric is defined as the ratio of cells for which the final scan report is equal to the analytical model, namely:

$$AI = \frac{\sum_{\substack{1\leq n\leq L \\ 1\leq m\leq H \\ k_i \in K}} \mathcal{F}(k_i, C_{n,m}) \stackrel{?}{=} \mathcal{F}_a(k_i, C_{n,m})}{L \cdot H \cdot |K|} \tag{8.16}$$

[1] *KoN* is a rule used in hard-decision combining methods for *collaborative spectrum sensing* [189]

Table 8.1: Default parameters value

Name	**Value**	**Name**	**Value**	**Name**	**Value**
N_{USV}	15	ϵ	0.05	KoN	1 (OR)
β	1	t_{coop}	$5s$	t_{scan}	$3s$
w_R	10	w_E	20	τ	0.003
T_K	$1s$	$\|K\|$	1 (CH 21)	λ	$-90dBm$
α^0	2	n_R	20	α_{DIFF}	0.5
D_{MUX}	2	v_{MAX}	$5^m/_s$	map size	$800m \times 800m$

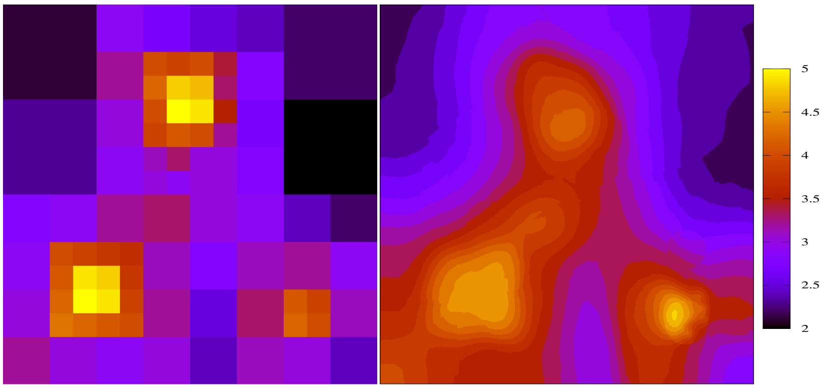
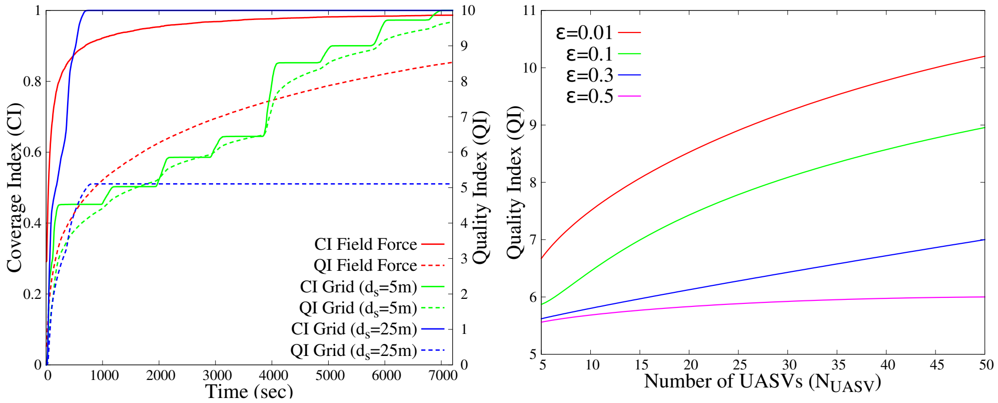


Figure 8.4: The original map (on the left) and the estimated one (on the right)

where the binary operator $\stackrel{?}{=}$ returns 1 if the two operands are equal and 0 otherwise.

The OMNeT++ simulator is used as simulation tool. If not otherwise specified, the default values of the system parameters are listed in Table 8.1. The parameter v_{MAX} defines the maximum velocity of each UASV. Since here it is considered a small case study scenario of $800m \times 800m$ with a single transmitter, it is also assumed a low transmitting power values for the TV stations, i.e. $Ptx_t = 20dBm$. The scenario is shaped as a square map where the TV transmitter is positioned on an corner while the UASVs perform the reconnaissance over the whole scenario. In this study, and is assumed a 2D scenario, i.e.: is considered a fixed flight altitude for the UASVs and we set this value to the height from the ground of the receiver antennas, so that the system will sense the TV frequencies at the receivers' level. The analysis of 3D scenarios are left as future work.

In Figure 8.4 is shown the ability of the fleet of UASVs to estimate the actual propagation conditions, reflected by the shadowing map. It can be seen that the main differences, between the real shadow map (left square) and the esti-



(a)

(b)

Figure 8.5: The convergence analysis of the CI and the QI of the field force method in comparison with the static scan method is depicted in Figure 8.5(a). The QI over N_{UASV} and ϵ is depicted in Figure 8.5(b).

mated map (right square), occur in locations where the UASVs were not able to detect the TV station transmissions, i.e. where the channel was reported as free. In this case, as described in Section 8.2.3, the UASVs assume that the value of α is just below the transmission threshold. In Figure 8.5(a) we analyse the convergence of the CI and the QI metrics over time. For this scenario we used $N_{UASV} = 25$ and $\epsilon = 0.01$. To better understand the advantage of using the proposed dynamic and adaptive method, we compared it with a static scan method. This latter method works as follow: the scenario is split in N_{UASV} equal parts $\Delta = \{\delta_i | 1 \leq i \leq N_{UASV}\}$ where each UASV u_i is in charge of scan the part δ_i. Each UASV explores with a snake method with a scans inter-distance of d_s. It can be noticed in Figure 8.5(a) that the static method with $d_s = 25m$ reach very quickly the total coverage but with a low QI. When $d_s = 5m$, instead, the CI grows very slowly but reaching at the end an higher QI. Looking now at the field force method, the CI metric quickly converges to 1, since the swarm of UASVs is able to explore the whole map through the attractive force. Also the QI metric increases over time as a consequence of the fact that more uncorrelated samples are gathered by the UASVs. In conclusion, it can be stated that the field force method is able to make a fast scan of the scenario while dynamically adapt the inter-distance scanning spacing depending on the actual characteristics of the environment.

In Sections 8.2.1 and 8.2.2 is defined ϵ as a critical parameter for scanning and exploring the scenario. In Figure 8.5(b), it is demonstrated that the tuning of ϵ has strong influence on the QI value. For example, for $\epsilon = 0.5$, the QI does not improve even increasing the number of UASVs; this is because higher values of

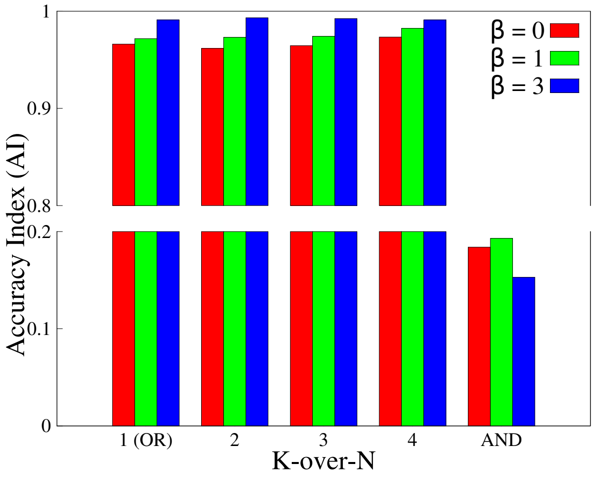


Figure 8.6: The AI over KoN and β

ϵ correspond to lower probability to execute the scan procedure. Finally, the AI metric is analysed over the *K-over-N* (*KoN*) parameter. For low value of KoN the accuracy index is nearly equal to 1, but when using the *AND* method, i.e. $KoN = |S(C_{n,m})|$, the performances fall down drastically. This fact can be explaining considering the average number of scans in a cell cluster $C_{n,m}$ ($|S(C_{n,m})|$); in our simulations this value is ≈ 200 and this implies that the probability that all scan reports are *busy* is very low. Clearly, the *AND* approach is not feasible because it will consider almost every cell as *free*. From the Figure 8.6 we can also notice the impact of the β parameter. Greater values of β corresponds to a larger coverage of the TV towers. In this case the conservative approach of increasing the value of β espouse the conservative approach of low value of KoN, while goes in opposite direction of the AND method.

Part III

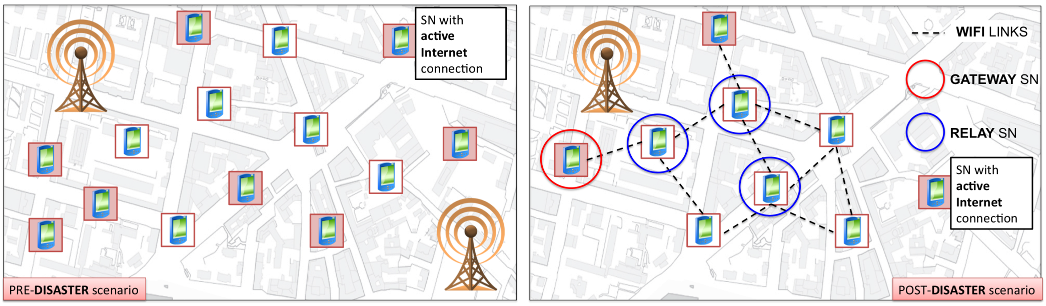
Spontaneous networks

Chapter 9

Spontaneous networks for disaster recovery

Current advances in telecommunications and recent trends in mobile services suggest that the future of wireless devices will be characterized by the keyword multi: multi-interface, multi-service and multi-reconfigurable. On the one hand, the variety of wireless technologies available on the market is opening the door to a plethora of heterogeneous devices provided with multiple radio interfaces, and connected to the Internet through several different access technologies (e.g. Wi-Fi, Wi-Max, 3G/4G, etc) [190]. On the other hand, novel mobile applications (e.g. location-based services and mobile social networks), will coexist with the traditional Internet-based services, determining a great range of possible Quality of Service (QoS) requirements which must be supported by the network providers. The way of managing the network complexity while guaranteeing the integration and interconnection of heterogeneous devices is challenging and object of several recent studies [191, 192]. However, the presence of different devices and wireless access technologies might not always constitute a limitation, rather a potential to exploit on several scenarios like the emergency ones. Recent catastrophic events (e.g. the Japanese tsunamis or the Katrina hurricane in US) demonstrated worldwide the fragility of fixed terrestrial communication infrastructures [5], as well as the need for more flexible and interoperable network architectures to support reliable communication among rescue teams and survivors [135].

Given the pervasive penetration of end-users devices (e.g. smartphones), spontaneous networks constitute promising solutions to implement emergency communication systems in which heterogeneous devices share their resources (e.g. Internet connection, energy power), in order to increase the network coverage, and thus the probability to reach survivors and to coordinate with rescue teams [193, 194, 195, 196, 101]. In addition, as the end-user devices embed sensors such as accelerometer, gyroscope, GPS, etc., they can enable the localization of



(a) (b)

Figure 9.1: The pre-disaster scenario is shown in Figure 9.1(a). The post-disaster scenario with the STEM-Net is shown in Figure 9.1(b) [199].

the end-users by also providing useful information about the context and the nature of the emergency [197, 198]. These spontaneous networks can guarantee emergency communication in post-disaster scenarios where the original infrastructure has been partially damaged by the occurrence of unpredictable or catastrophic events [199]. However, the heterogeneity of devices and wireless access technologies poses important challenges on the network deployment and management. Furthermore, there is the opportunity in the future to integrate these spontaneous networks with the unmanned vehicles deployed by the recovery team.

In this Chapter, is proposed the STEM-Net architecture (introduced in Chapter 5) as a viable network model to handle the devices' heterogeneity and to enable spontaneous networking functionalities in post-disaster scenarios. In *STEM-Net* the wireless devices (the SNs) are able to adapt their transmitting configurations, cover different roles (e.g. router, bridge, etc) according to the system needs and evolve their functionalities through cooperation with other nodes. Here, is provides a proof-of-concept of the principles of nodes' mutation and evolution, by discussing how heterogeneous end-user devices provided with SN capabilities can dynamically self-organize into multi-hop networks, and share the Internet access by switching among three roles: stub, transit and gateway SNs. A bio-inspired distributed gateway selection mode is defined to allow each SN device to autonomously decide its current role, based on the system needs and on the individual hardware characteristics and resources (e.g. residual energy or queue occupation).

9.1 Scenario Model

The flexibility of the Stem Node (SN) architecture described in Chapter 5 allow its implementation on several different environments and for different purposes. In this Chapter, hence, the SN architecture is instantiated for a specific deployment, i.e. spontaneous emergency communication networks. The focus is restricted to a specific class of devices, i.e. mobile end-users devices (smartphones, tablets) that are usually considered only the end-points of the communication infrastructures. In contrast, the goal of this study is to demonstrate how these devices, augmented by the reconfiguration capabilities offered by the SN architecture, can self-organize to create a multi-hop autonomous wireless networks. Let us consider the pre-emergency scenario depicted in Figure 9.1(a) where the heterogeneous end-user devices can be provided with different capabilities in terms of radio equipments, battery energy and computational power. All these devices are equipped with a WiFi radio transceiver and they can communicate on a Common Control Channel (CCC) on the 2.4 GHz ISM band. Furthermore, some of the end-user devices are also provided with a mobile Internet connection (through the 3G/4G cellular infrastructure). Let us assume that, at a specific time instant, an unforeseen event occurs within the scenario, so that all the devices need to access the Internet in order to transmit emergency data. An example of such event might be a natural disaster (e.g. earthquake). In this scenario parts of the infrastructure may be damaged or not available (e.g. due to the excessive network load) and the survivors, with their own smart end-device, need to transmit geo-location data in order to help the relief operations. In this Chapter is investigated how to dynamically deploy a multi-hop STEM-Net (Figure 9.1(b)), so that the isolated devices can still access the Internet through a multi-hop communication network made by all the cooperative devices.

9.2 Stem Nodes (SNs) implementation

For this study, let us suppose again that each device implements the SN architecture depicted in Figure 5.1. Due to the intrinsic user-constrained mobility of the devices, the Mobility Actuator component is not present. Again, the Evolution module is not present. The main architecture shrinks in the architecture depicted in Figure 9.2.

9.2.1 Roles Definition

Three different SN roles (i.e. M=3) will be used:

- *Gateway* (r_1): this is an end-user device with an active Internet connection provided by a 3G/4G radio interface. Moreover, it has software capabilities to share its connection with other devices.

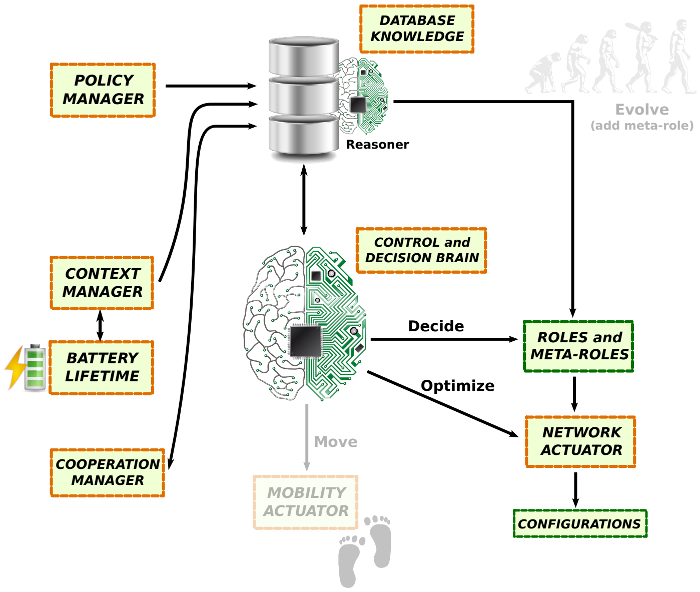


Figure 9.2: The used subset of the Stem-Node architecture for the spontaneous network creation

- *Relay* (r_2): this is an end-user device with routing capabilities and is capable of forwarding the incoming traffic towards the closest SN gateway. It supports an enhanced version of the AODV [200] routing protocol described in Section 9.3.2.

- *Stub* (r_3): this is a device which is not involved in the network management process (i.e. it does not serve as gateway or relay). However, it might generate traffic flows.

Based on its own hardware characteristics, each SN has an initial set of roles R it is able to assume and specific capabilities to evolve its network functionalities in the future, i.e. to expand the set R. In this study, however, will not be described the evolutionary methods that the SNs use to exchange software modules. In fact, it will be assumed that the routing module (indispensable to assume the relay role) and the *Tethering* module (indispensable to assume the gateway role) are already installed on the end-user devices.
In this work will be analysed the decision algorithm where each SN can dynamically mutate its role in order to implement the system needs. Details on how module distribution and installation can be performed in a mesh-phone environment are discussed in [195][148].

9.2.2 Policy Manager (PM)

The PM component is in charge of guaranteeing the satisfaction of the QoS requirements for the goal functions defined by the specific communication scenario. In the context of emergency communications, one important goal is to guarantee satisfactory bandwidth for the STEM-Net, so that safety applications can be supported adequately. At the same time, the spontaneous network is composed of battery-powered devices and hence network lifetime should be taken into account during the deployment. For this reason, are considered two metrics that reflect the status of each SN i in terms of resource utilization:

- The *Energy factor* ($E(i)$). This is defined as $\frac{E_{act}(i)}{E_{tot}(i)}$, where E_{act} denotes the actual energy level (in mW) of SN i, while E_{tot} is the energy in a full charge battery, which is device-dependant. Although the energy ratio might not be comparable when considering SNs that are also provided with different battery capacities (i.e. different values of $E_{tot}(i)$), the metric is formulated in this way since it is assumed that all SNs will participate to the emergency network deployment: hence, the aim is to discharge the SNs at the same rate, as described in detail in Section 9.3.

- The *Congestion factor* ($C(i)$). This is a measure of the network congestion, and it is expressed as the occupation ratio of the queue at IFQ layer. For double-interface devices, we distinguish between $C_{Wifi}(i)$, i.e. the queue occupation ratio on the WiFi link, and $C_{cell}(i)$, i.e. the queue occupation ratio on the 3G/4G link. Several metrics could be used to reflect the congestion of a SN; however, this formulation of $C(i)$ is chosen because: (*i*) higher queue occupation might translate into higher packet drop probability. This is crucial since each packet in this emergency scenario might convey crucial information; (*ii*) the metric value is computed locally on each SN without introducing additional network overhead.

9.2.3 Cooperation Manager (CM)

In STEM-Net, the role selection is performed locally at each SN, on the basis of the system-wide requirements that must be guaranteed (e.g. network throughput and energy lifetime maximization). To bring out a collective intelligent behaviour, some information must be shared by the SNs through the *CM* and then maintained locally in the DataBase Knowledge module (*DBK*). More in detail, each SN i maintains a Gateway List (GL_i), that includes the list of SN gateways up to h hops of distance. Moreover, each SN i will periodically share the following information with its neighbours:

$$< E(i), C_{Wifi}(i), C_{cell}(i), D(i), role(i) > \tag{9.1}$$

where the first three factors are respectively the energy and congestion metric defined in Section 9.2.2, $D(i)$ denotes the average distance of SN i from the gateways in its GL_i, and $role(i)$ denotes the current role of SN i, according to the list provided in Section 9.2.1. This information is appended to `HELLO` messages which are periodically sent by each SN.
Once receiving an `HELLO` message from SN j, the SN i extracts the data and stores them in the Neighbour List (NL_i) in the DBK.

As explained before, STEM-Net foresees the utilization of two software modules:

- *Routing module*. A SN can participate to the routing process and can work as relay in a multi-hop path.

- *3G/4G Tether module*. A double-interface SN can share its Internet connection with other devices, at a single or multi hops distance.

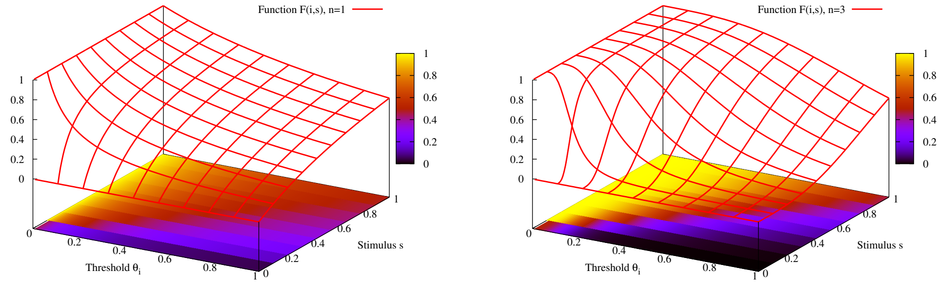
It is assumed in this study that all devices will collaborate and share their resources to guarantee system-wide connectivity, although this might not occur in presence of selfish behaviours.

9.2.4 Decision and Control Brain

The DCB component is in charge of deciding the current role of each SN, based on its own status and on the system policy defined in Section 9.2.2. The selection of the role is performed in order to: (*i*) maximize the STEM-Net throughput and (*ii*) pro-long the STEM-Net lifetime, considering the additional energy consumption required to serve as gateway. Similarly, once a Stub SN needs to communicate data to a gateway, a distributed election scheme is integrated within the AODV routing protocol [200] so that SNs can mutate to relay SN. The approach is based on the division of labour model that is inspired by the task allocation process observed in several insect colonies. In the original problem formulation [201], a set of individuals is given a set of tasks to be performed. To execute this specific task is associated to it a numeric stimulus s, which quantifies its importance for the colony. Moreover, each individual i is characterized by a threshold θ_i that defines its propensity to respond to the stimulus and thus to perform the task. A response function $F(i, s)$ is then defined as the probability of individual i to perform the task, given a threshold θ_i. In [201], the function $F(i, s)$ is computed as follows:

$$F(i, s) = \frac{s^n}{s^n + \theta_i^n} \tag{9.2}$$

where n is a steepness factor. In Figures 9.3(a) and 9.3(b) is depicted the $F(i, s)$ values as a function of the threshold intensity (x-axis) and of the stimulus intensity



(a)

(b)

Figure 9.3: The shape of $F(i, s)$ as a function of the threshold intensity (x axis) and of the stimulus intensity (y axis), for n=1 (Figure 9.3(a)), and n=3 (Figure 9.3(b)).

(y axis). For $s \gg \theta_i$ the individual i has an high response function to the stimulus and hence it will likely perform the task, while it will likely ignore it for $s \ll \theta_i$. In the proposed formulation, the stimulus measures the need for a specific role (r_1 or r_2) in the network, while the threshold measures the ability of an individual SN to assume that role.

9.3 The STEM-Net Creation

In this Section, is described how the STEM-Net is deployed through the self-configuration of the SNs, on the basis of the stimulus-response model previously introduced. First, is detailed the distributed scheme through which the SNs provided with the gateway capabilities can decide to assume the role r_1 (in Section 9.3.1). Then, is presented the analogous scheme which allows the SNs provided with the relay capabilities to participate to the routing process. Table 9.1 shows the list of parameters and variables used by the STEM-Net deployment algorithms.

9.3.1 Distributed Gateways Election

In a practical STEM-Net deployment for emergency communications, the number of available Internet gateways constitutes a trade-off between the system throughput and the energy lifetime. Intuitively, the higher is the number of gateways, the lower is the average length of the multi-hop routes between a stub SN and a gateway, and thus the higher the network performance. At the same

Table 9.1: Abbreviations and Notations

$E(i)$	energy level ($0 \leq E(i) \leq 1$) of SN i.
$C_{cell}(i)$	queue occupation level ($0 \leq C_{cell}(i) \leq 1$) over the cellular link of SN i.
$C_{wifi}(i)$	queue occupation level ($0 \leq C_{wifi}(i) \leq 1$) over the WiFi link of SN i.
$D(i)$	average distance from SN i to gateway SNs.
GL_i	gateway list of SN i.
NL_i	neighbour list of SN i.
α	weight ($0 \leq \alpha \leq 1$) of energy requirements.
β	weight ($0 \leq \beta \leq 1$) of throughput requirements.
T_{sm}	timer for sending a STIMULUS message.
T_e	election phase duration.
$SM_{gw}(i)$	status metric of SN i (gateway).
$TM_{gw}(i)$	threshold metric of SN i (stub).
$RF_{gw}(i,j)$	probability of SN i to respond to a STIMULUS from SN j.
n	stepness factor of the RF function (equal to 3 in our formulation).
$Del(i)$	defer timer of SN i before sending a GATEWAY message.
δ_e	seconds remaining till the end of the current election phase ($0 \leq \delta_e \leq T_e$).
$GC(i)$	gateway cost function of SN i.
N	number of newly elected gateways during current election phase.
$SM_{rel}(i)$	status metric of SN i, source node of the path.
$TM_{rel}^{(j,k)}(i)$	threshold metric of SN i, receiving a REQUEST message for the path $j - k$.
$RF_{rel}^{(j,k)}(i)$	probability of SN i to re-broadcast a REQUEST message for the path $j - k$.
T_{req}	time limit threshold for receiving a REPLY message.
γ_{req}	fixed increase of the $SM_{rel}(i)$ value, at each routing creation attempt.
n_{retry}	number of route creation attempts.
p_{start}	probability of a SN to self-elect as gateway at the network startup.
T_{route}	time validity of each route.

time, turning each SN with hardware capabilities into a gateway might significantly impact the network lifetime, since mobile end-user devices are usually provided with limited energy resources. Several centralized gateway selection schemes have been proposed in the literature of wireless networks [202]. Differently from the existing schemes, the STEM-Net solution attempts to capture the trade-off through a distributed scheme which implements the stimulus-response model at each local node in order to favour the emergence of the optimal system-wide behaviour. In this case, the task corresponds to the action of self-electing as SN gateway, the stimulus corresponds to the system-wide need of gateways, and the threshold θ_i quantifies the ability of SN i to serve as gateway. Thus, it is defined the status metric of a SN gateway i (i.e. $SM_{gw}(i)$) as follows:

$$SM_{gw}(i) = \alpha \cdot (1 - E(i)) + \beta \cdot C_{cell}(i) \tag{9.3}$$

where $E(i)$ and $C_{cell}(i)$ are, respectively, the energy and congestion metrics defined in Section 9.2.2. The α and β weights (with α+β=1) balance the system requirements, i.e. throughput optimization versus battery life maximization. It is easy to notice that $SM_{gw}(i)$ increase if (*i*) gateway SN i has low energy (and thus must be replaced) or (*ii*) gateway SN i is experiencing network congestion (and thus additional gateways should be elected). Every T_{sm}+r time intervals, each SN gateway i broadcasts its $SM(i)$, $E(i)$ and $C(i)$ values on a `STIMULUS` message, where r is a random generated delay (with $r < T_{sm}$) which has been introduced to avoid nodes' synchronization. The `STIMULUS` message is re-broadcasted for h_s hops (in the experiments h_s=2). On receiving the `STIMULUS` message, an election phase is started for T_e seconds, during which other SNs can self-elect as gateways. To this purpose, each SN i is associated to a threshold metric (i.e. $TM_{gw}(i)$) which denotes its current ability to cover the gateway role and it is defined as follows:

$$TM_{gw}(i) = \alpha \cdot \frac{E(i)}{E_{Max}} + \beta \cdot \frac{D(i)}{D_{Max}} \tag{9.4}$$

where $D(i)$ (i.e. the average distance from gateway SNs in the GL_i) has been introduced in order to avoid electing SNs too close each other, since they might likely experience mutual interference. Vice versa, if SNs are distant enough, the risk of mutual interference is reduced and multiple traffic flows having as destinations different gateways can proceed in parallel, leading to an effective re-utilization of channel resources over the scenario. The E_{Max} and D_{Max} factors represent the maximum metric values among all the neighbours of SN i, i.e.:

$$E_{Max} = max\{E(j)|\forall j \in NL_i\} \tag{9.5}$$

$$D_{Max} = max\{D(j)|\forall j \in NL_i\} \tag{9.6}$$

In Equation 9.4, the α and β values balance the system requirements as in Equation 9.3. It is easy to see that $TM_{gw}(i)$ is in range]0:1], where a value close to 1 indicates the fact that SN i is more suitable to work as gateway SN than its neighbours. Finally, it is defined the response function of SN i to a stimulus coming from a SN j as follows (n=3):

$$RF_{gw}(i,j) = \frac{SM_{gw}(j)^3}{SM_{gw}(j)^3 + (1 - TM_{gw}(i))^3} \tag{9.7}$$

When SN i receives a `STIMULUS` message from SN j it will self-elects as gateway with probability $RF_{gw}(i,j)$ and, if this is the case, it will send a `GATEWAY` message, which includes the values of $E(i)$ and $C(i)$. The message is then rebroadcasted within the STEM-Net in order to reach all the SNs, which will update their GL structure accordingly. Here, are considered two different election policies:

- *Single Election*. Only one SN can respond to a `STIMULUS` message coming from SN j. This is realized through a contention-based filter as in [202]. Each SN willing to elect as gateway computes a random delay $Del(i)$ before sending the `GATEWAY` message, which is proportional to the value of the response function $RF_{gw}(i,j)$:

  $$Del(i) = (1 - RF_{gw}(i,j)) \cdot T_e \tag{9.8}$$

  If, while deferring, a SN overhears a `GATEWAY` message from another SN, it will abort its candidature. Since the defer time depends from $RF_{gw}(i,j)$, SNs with higher response values experience also higher probability to win the election and thus to become a gateway.

- *Multiple Election*. With this method, multiple SNs can respond positively to a `STIMULUS` message, based on the intensity of the stimulus. As in the previous case, each SN computes the stimulus function through Equation 9.7 and decides whether to send a `GATEWAY` message or not. However, differently from the previous case, a SN (e.g. SN i) does not cancel its transmission attempt when it overhears a `GATEWAY` message from SN k which responds to the `STIMULUS` from SN j. Instead, SN i recomputes the intensity of the stimulus (i.e. of the status metric from SN j) as follows:

  $$SM_{gw}(i,j) = \alpha \cdot (1 - E(k)) + \beta \cdot \frac{C_{cell}(j)}{N+1} \tag{9.9}$$

  Here N represents the number of newly elected gateways that has sent a `GATEWAY` message during the current election phase. The idea behind

Equation 9.9 is that each new gateway will contribute in reducing the contention currently experienced by SN j, assuming an equal share of traffic loads among the available gateways (β component of Equation 9.9). Similarly, the α component of the status metric is modified based on the energy level of the newly elected node SN k. Then, SN i recomputes the response function $RF_{gw}(i,j)$ through Equation 9.7, and with probability $RF_{gw}(i,j)$ it computes the defer timer as before:

$$Del(i) = (1 - RF_{gw}(i,j)) \cdot \delta_e \tag{9.10}$$

where δ_e are the seconds remaining till the end of the current election phase.

The first policy induces a gradual variation of the network topology, since during each election phase at most one new gateway is elected. The second policy attempts to reduce the converge time by scaling the number of gateways to the actual system needs, expressed through the α and β values. Both of these policies are evaluated in Section 9.4.

At the end of the election phase, the SN j that produced the `STIMULUS` message must decide whether to stay as gateway or whether to leave the role. This is performed in a straightforward way by comparing the benefits of having an additional gateway in the STEM-Net against the risk of consuming all the residual energy of SN j. It is measured such trade-off through the Gateway Cost ($GC(j)$) function, defined as follows:

$$GC(j) = \beta \cdot \left(\frac{C_{cell}(j)}{N} - \frac{C_{cell}(j)}{N+1} \right) - \alpha \cdot (1 - E(j)) \tag{9.11}$$

where the first addendum estimates the network congestion reduction if SN j keeps the gateway role, while the second addendum quantifies the energy risk of SN j (i.e. the possibility that SN will run out of energy). If $N = 0$, than the first addendum is set equal to 0. It is easy to see that $GC(j)$ can assume values in range [-1:1], where a positive value indicates that the gain of having one more gateway in the network will overcomes the energy constraints of SN j. If $GC(j) > 0$, then SN j decides to stay as gateway with probability p ($p >$ 0.5, equal to 0.8 in the experiments), otherwise it decides with the same probability to leave the role. In case of decision to leave, the SN j broadcasts a `LEAVE` message, that announces the fact that SN j is not reachable anymore as network gateway. As for the `GATEWAY` messages, also `LEAVE` messages must be re-broadcasted network-wide. The main operations of the distributed gateway election scheme are summarized by Algorithm 9.

Algorithm 9: Distributed Gateway Election scheme

```
Operations performed by SN $i$, with $role(i)$=gateway (every $T_{sm} + r$ intervals).
    Compute $SM_{gw}(i)$ through Equation 9.3.
    Send STIMULUS message by including $SM_{gw}(i)$, $C_{cell}(i)$ and $E(i)$ values.
    Wait for $T_e$ time intervals (election phase).
    Compute gateway cost function $GC(i)$ and random value $rand$.
    if ($GC(i) \leq 0$ and $p \leq rand$) or ($GC(i) > 0$ and $p > rand$) then
        Send LEAVE message and set $role(i)$=stub.
    end if

Operations performed by SN $i$, with $role(i)$=stub (during the election phase)
On receiving a STIMULUS message from SN $j$:
    Compute response function $RF_{gw}(i,j)$ through Equations 9.4 and 9.7.
    Compute random value $rand$.
    if $RF_{gw}(i,j) \geq rand$ then
        Compute defer time $Del(i)$ through Equation 9.8.
        Set $transmit$=true and start Defer Timer (DF).
    end if
On receiving a GATEWAY message from $SNk$:
    if Single Election mode is used then
        Cancel transmission attempt by setting $transmit$=false
    else
        Increase $N$ (number of newly elected gateways).
        Recompute $SM_{gw}(i,j)$ through Equation 9.9.
        Recompute defer time $Del(i)$ through Equation 9.10.
    end if
On the expire event of the Defer Timer (DF):
    if $transmit$ == true then
        Send GATEWAY message with $E(i)$ value.
    end if
```

9.3.2 Distributed Relays Election and Routing Process

In STEM-Net, each time a SN has to communicate emergency data, it needs a path toward a SN gateway. To this purpose, let us assume that each SN (e.g. SN j) keeps in the *Gateway List* (GL_j) the information about the gateway addresses and the distance from each of them, in terms of number of hops. The GL_j is dynamically updated by SN j each time it receives a `GATEWAY` or `LEAVE` message, announcing respectively the adding/removal of a gateway SN. Although several gateway selection policies might be implemented and used, is assumed without loss of generality that SN j will always choose to send its data toward the closest gateway in its GL_j. On average, this choice translates into a fair load balance among the available gateways, assuming an uniform distribution of the SNs within the network scenario. As for the gateways selection, the relays selection for a specific source-destination path determines the end-to-end throughput and it has an impact on the energy lifetime of the SNs along that path. Again, this trade-off is captured through a distributed relay election scheme, which is based on the stimulus-response mechanism of Section 9.3.1 with different definition of the SM, TM and RF functions (described in the following). The mechanism is then integrated into the popular AODV routing protocol [200]. More specifically, when SN j needs a routing path toward gateway SN k, it broadcasts a `REQUEST` message, including the selected destination (e.g SN k) and a status metric value $SM_{rel}(j)$. In this formulation, $SM_{rel}(j)$ is expressed by the current queue occupation level of SN j on the WiFi interface, i.e. $C_{wifi}(j)$, and implicitly reflects the urgency of the communication. Differently from the basic AODV scheme, not all the SNs receiving the `REQUEST` will re-broadcast it. Instead, each SN (e.g. SN i) receiving the `REQUEST` message computes a threshold metric (i.e. $TM_{rel}^{(j,k)}(i)$) which denotes its capability to participate to the routing path between SN i and SN k. It is considered a formulation of $TM_{rel}^{(j,k)}(i)$ as follows:

$$TM_{rel}^{(j,k)}(i) = \alpha \cdot \frac{E(i)}{E_{Max}} + \beta \cdot \left(1 - \left(\frac{D(i,k)}{2 \cdot D_{Max}(k)} + \frac{C_{wifi}(i)}{2 \cdot C_{Max}}\right)\right) \tag{9.12}$$

Here, the first addendum (i.e. $\frac{E(i)}{E_{Max}}$) denotes the energy level of SN i, compared to its neighbours, while the second addendum denotes the network performance of SN i in case it will be included in the routing path, considering the actual queue utilization on the WiFi interface (i.e. C_{wifi}), and the distance to the destination (i.e. $D(i,k)$). The factor C_{Max} represents the maximum metric values among all the neighbours of SN i, i.e.: $C_{Max} = max\{C_{wifi}(j) | \forall j \in NL_i\}$. It is easy to see that $TM_{rel}^{(j,k)}(i)$ assumes values in range [0:1], where a value close to 1 indicates that SN i might be a good candidate as network relay for the path $j-k$, compared to its neighbourhood. The α and β factors express the trade-off between network throughput and energy consumption of each SN, on the basis of the STEM-Net deployment requirements. Then, SN i computes the response function $RF_{rel}(i, j)$:

$$RF_{rel}^{(j,k)}(i) = \frac{SM_{rel}(j)^3}{SM_{rel}(j)^3 + (1 - TM_{rel}^{(j,k)}(i))^3} \tag{9.13}$$

With probability $RF_{rel}^{(j,k)}(i)$, SN i broadcasts the `REQUEST` message, otherwise it discards it. When the message reaches SN j, it sends back a `REPLY` message, like in the classical AODV protocol. It might happen that no nodes forward the `REQUEST` message, and thus the routing process fails in finding the path. In order to deal with these events, it is assumed that the source node (SN j) will keep track of the time the first `REQUEST` message is issued. In case no `REPLY` message is received within a time interval T_{REQ}, it starts a new routing process, by broadcasting a new `REQUEST` message and by increasing the intensity of the stimulus:

$$SM_{rel}(j) = C_{wifi}(j) + \gamma_{rel} \cdot n_{retry} \tag{9.14}$$

where γ_{rel} is a fixed constant (equal to 0.2 in the experiments), and n_{retry} is the number of request retries performed by SN j. The value of $SM_{rel}(j)$ is normalized to 1 in case the right part of Equation 9.14 produces a value higher than 1.

9.3.3 Network Bootstrap and Maintenance

The routing process described so far assumes that, before starting a communication, a sender SN has always at least one gateway address in its GL. However, at STEM-Net startup, no stimulus is available, hance no `GATEWAY` messages are generated. For this reason, a bootstrap mechanism is required to decide the initial presence of the SN gateways. This problem is solved by letting each SN provided with the GW capabilities to self-elect as gateway with a fixed probability p_{start} (equal to 0.5 in our experiments). The initial number of gateways does not affect the future network deployment, since the number of gateways is progressively adjusted on the basis of the current stimulus, defined by Equation 9.3. Similarly, in order to deal with the events in which no gateway is available, it is assumed that each SN provided equipped with the hardware capabilities will self-elect as gateway in case it does not overhear a `STIMULUS` or `GATEWAY` message for a maximum of $3 \cdot T_e$ time intervals. All messages used by STEM-Nen (i.e. `STIMULUS`, `GATEWAY` and `LEAVE`) include a sequence number to prevent out-of-order or duplicate processing.

In STEM-Net, dynamic route maintenance is required since the network topology can dynamically change as a consequence of the users' mobility, the variation of the time response functions at each SN, and the gateway election process. To address these latter fact, each time SN i leaves the gateway role and issues a `LEAVE` message, all the routes destined to SN i are invalidated, and re-established through the algorithm defined in Section 9.3.2. To address the general problem, a maximum time validity is associated to each route (T_{route}); when this timer expires, the route is invalidated, and SNs must issue a new routing request to build a new path.

9.4 Role selection evaluation

In this Section, is evaluated the performance of the distributed role selection algorithm for spontaneous network deployment in emergency scenarios. More specifically, in OMNeT++ is modelled a scenario of *800mx800m*, with N smartphones randomly placed on it. Each device is assumed to have a limited battery capacity, initially chosen randomly in the range of *[1500:2000] mA/h*. End-user mobility is not considered in our analysis. All the mobile devices are provided with a Wi-Fi interface, but only one third of them (i.e. $\frac{N}{3}$) is able to access the Internet through a 3G/4G Internet connection, and thus can assume the gateway role (r_1). At a specific time instant, an emergency event occurs in the network, so that all the smartphones need to transmit data to Internet, directly (in case they serve as gateway), or indirectly (i.e. sending data to an active gateway through the multi-hop ad-hoc network). Let us assume that each device implements the SN architecture of Figure 5.1, and that modules' exchanges and software upgrades have already been performed according to the dissemination scheme presented in [148]. This means that the role-set R of each SN will not change during the simulation, however each SN can dynamically mutate its current role, choosing among the $|R|$ possible alternatives. Readers interested in understanding the phases of modules exhange and upgrade can refer to [148]. Here, the focus is on the evaluation of distributed network deployment strategies. Unless otherwise stated, the values of simulation parameters are listed in Table 9.2. The following metrics are studied in the analysis:

- *Network Throughput (NT)*: this is defined as the amount of KB/s which are forwarded outside the STEM-Net by gateway SNs on the 3G/4G cellular links.

- *Delivery Ratio (DR)*: this is defined as the ratio of packets which are correctly forwarded outside the STEM-Net, over the total load generated by the SNs.

- *Network Delay (ND)*: this is defined as the average delay experienced by packets to flow from source nodes (i.e. stub SNs) to gateway SNs.

- *Energy Lifetime (EL)*: this is computed as the time interval (in hours) from the network startup till the instant the first SN runs out of power.

Table 9.2: Simulation Parameters

Network size	800mx800m
Number of SNs	{20,30,40,50,60,70,80,90,100}
Traffic load of each SN	[100-1000] KB/s
Battery capacity	[1500-2000] mA/h
WiFi rate	54 Mb/s
3G/4G rate	1 Mb/s
Transmit power	7 dBm
Transmit radius	200 meters
α (Equation 9.3)	0.5
β (Equation 9.3)	0.5
T_{sm} (stimulus beaconing)	5s
h (stimulus propagation hops)	3
T_{route} (time validity of each route)	10s
Election policy	Multiple gateway

- *Gateway ratio (GR)*: this is computed as the average number of SN gateways concurrently active inside the network, over the total number of SNs (i.e. N).

The following four different schemes of gateway selection are implemented in the CDB of each SN:

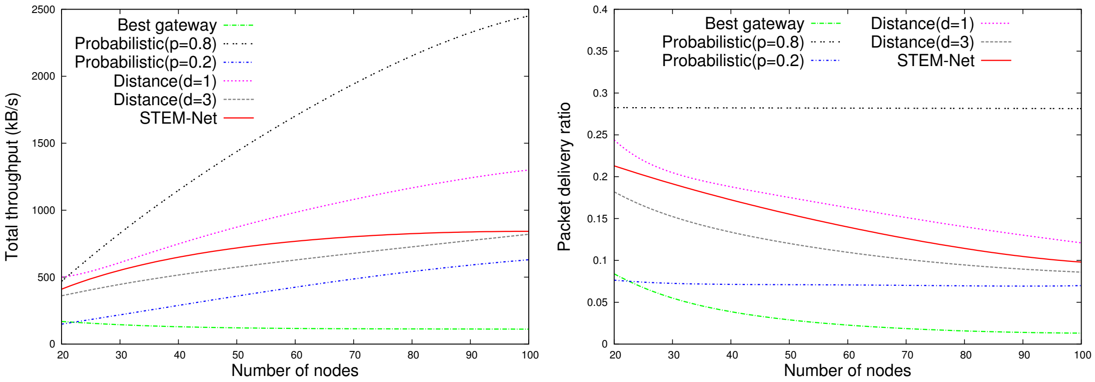
- *STEM-Net*: in this case, each SN provided with an active 3G/4G connection decides whether to elect as a gateway or not according to the bio-inspired approach described in Section 9.2.4.

- *Probabilistic-based*: in this case, each SN decides with a fixed probability (i.e. p) to self-elect as gateway. Are evaluated two configurations for p=0.8 and p=0.2.

- *Distance-based*: in this case, each SN self-elects as gateway in case it does not detect any other gateway at a distance equal or lower than a threshold (i.e. d). Are evaluated two configurations for d=1 and d=3.

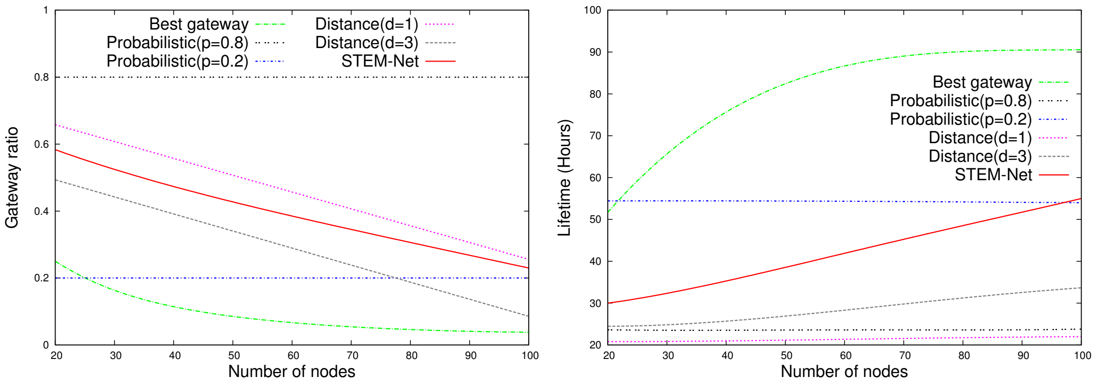
- *Best Gateway*: in this case, a single SN gateway is elected in the network, by choosing the one having the maximum residual energy, in a centralized way.

In Probabilistic-based, Distance-based and Best-gateway schemes, distributed gateway elections are performed every T_{sm} time intervals, and AODV is used for



(a) (b)

Figure 9.4: The NT and DR metrics as a function of the number of SNs are shown in Figures 9.4(a) and 9.4(b), respectively.



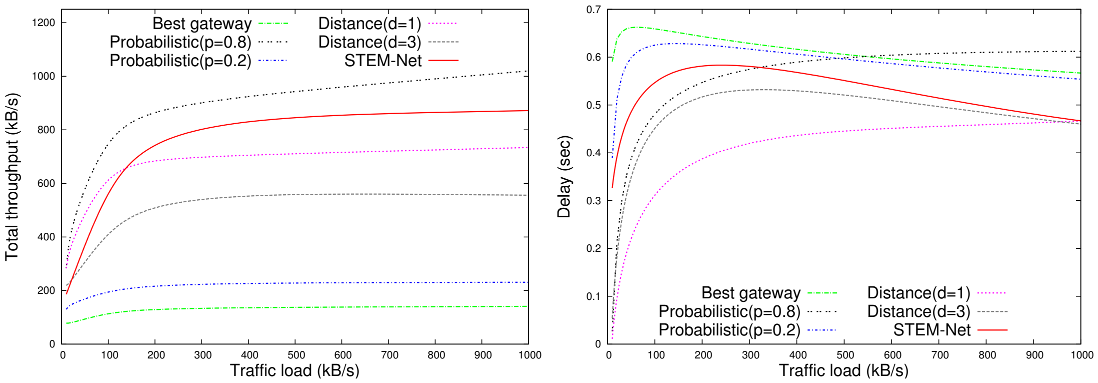
(a) (b)

Figure 9.5: The GR and EL metrics as a function of the number of SNs are shown in Figures 9.5(a) and 9.5(b), respectively.

routing. In STEM-Net, routing is performed according to the distributed relay selection scheme described at Section 9.3.2. In Section 9.5, is shown the system performance when varying the number of SNs. In Section 9.5.1, is repeated the same study when varying the load produced by each SN. Finally, in Section 9.5.2, is investigated the performance of the four schemes in a dynamic network scenario, and is further characterized the impact of design requirements (expressed by the α and β parameters) on the STEM-Net deployment.

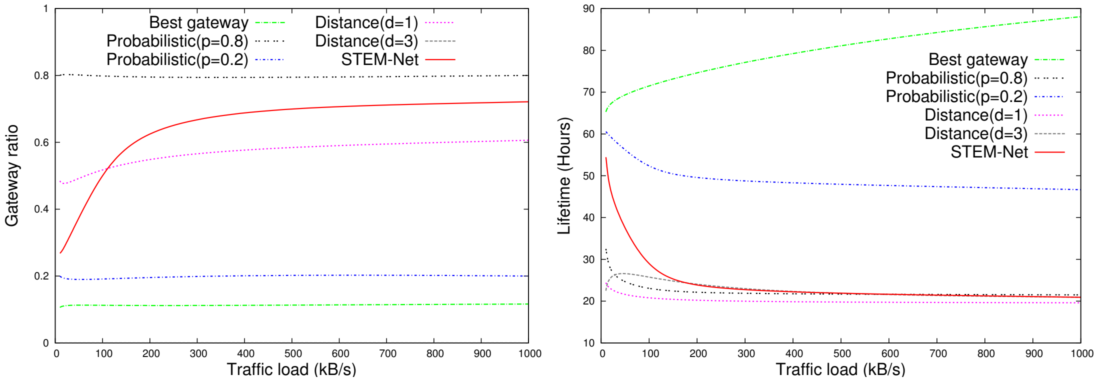
9.5 Connection Analysis

Figures 9.4(a), 9.4(b), 9.5(a) and 9.5(b) shows the NT, DR, GR and EL metrics described above as a function of the number of SNs (N). Each SN produces a constant load of *100 KB/s*. For the Probabilistic schemes, the higher is the network density, the higher is the number of gateways, and hence the lower are average path length and the packet drop probability due to channel interference or to queue overflow at intermediate hops. This is reflected by Figure 9.4(a), which shows that the Probabilistic (p=0.8) scheme guarantees the highest throughput for increasing values of N. The performance of STEM-Net is lower than Probabilistic (p=0.8) and Distance (d=1), since the number of gateways is dynamically adjusted based on network congestion conditions and not on the number of SNs composing the emergency network; however, STEM-Net still outperforms the other three schemes. In contrast, the Best Gateway scheme produces the lowest throughput, due to the fact that all the outbound traffic is directed toward a single node. Figure 9.4(b) confirms the previous analysis in terms of DR metric. Figure 9.5(a) depicts the GR metric, which is computed by considering the ratio between the actual number of SN gateways, over the maximum number of electable gateways (i.e. $\frac{N}{3}$). As expected, the GR value is constant for the Probabilistic scheme, and equal to the p parameter. Viceversa, for the Distance-based scheme, the GR values decrease with N, due to the fact that SN gateways are more likely located at a distance lower than h hops. Also the GR values of STEM-Net show a decreasing trend. This can be justified by considering two aspects of STEM-Net: (*i*) the response function of Equation 9.4 takes into account the average distance among SNs, and prevents electing gateways too close one from each other, specially on high dense networks; (*ii*) the stimulus function of Equation 9.3 takes into account the network congestion level, which in the experiments setting does not increase linearly with N, since each SN produces a moderate traffic load. The ability of STEM-Net to self-adapt to variable system loads is made more evident in Section 9.5.1. Finally, Figure 9.5(b) depicts the Network Lifetime (NL) metric as a function of the number of SNs. The situation is the opposite of Figure 9.5(a). The Probabilistic-based scheme (p=0.8) produces the worst performance, since it involves the largest number of active gateways, which will faster consume their batteries compared to the other schemes. Viceversa, approaches with lower GR (Figure 9.5(a)) maximize the lifetime, and the Best Gateway provides the highest performance, since at each election phase only one SN consumes its battery as gateway, and the SN with the highest residual energy is selected. Figure 9.5(b) confirms that STEM-Net is able to prolong lifetime even in dense network scenarios, as energy issues are accounted in the selection process of both gateway and relay SNs (Equations 9.4 and 9.12). Moreover, combining Figure 9.4(a) and 9.5(b), it becomes evident that STEM-Net is able to guarantee the best-tradeoff between



(a) (b)

Figure 9.6: The NT and ND metrics as a function of the load produced by each SN are shown in Figures 9.6(a) and 9.6(b), respectively.



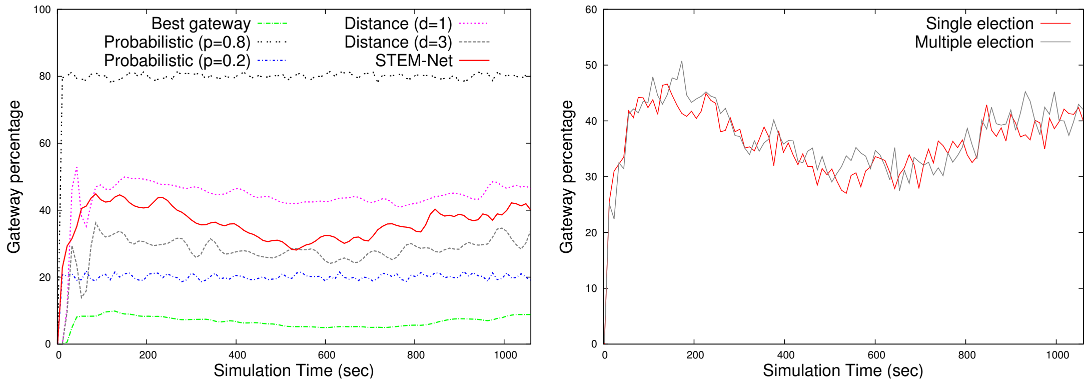
(a) (b)

Figure 9.7: The GR and EL metrics as a function of the load produced by each SN are shown in Figures 9.7(a) and 9.7(b), respectively.

throughput and energy issues, in accordance with the network requirements reflected by the values of α and β (both equal to 0.5). Altering these coefficients, an unbalanced network deployment privileging one of the two issues can be dynamically guaranteed by STEM-Net, as demonstrated in Section 9.5.2.

9.5.1 Load Analysis

Figures 9.6(a), 9.6(b), 9.7(a) and 9.7(b) show the NT, ND, GR and EL metrics described above as a function of the load (in KB/s) produced by each SN. The configuration considered is with 30 SNs in the network (N=30). Figure 9.6(a) confirms that STEM-Net guarantees an end-to-end throughput close to the Prob-



(a)

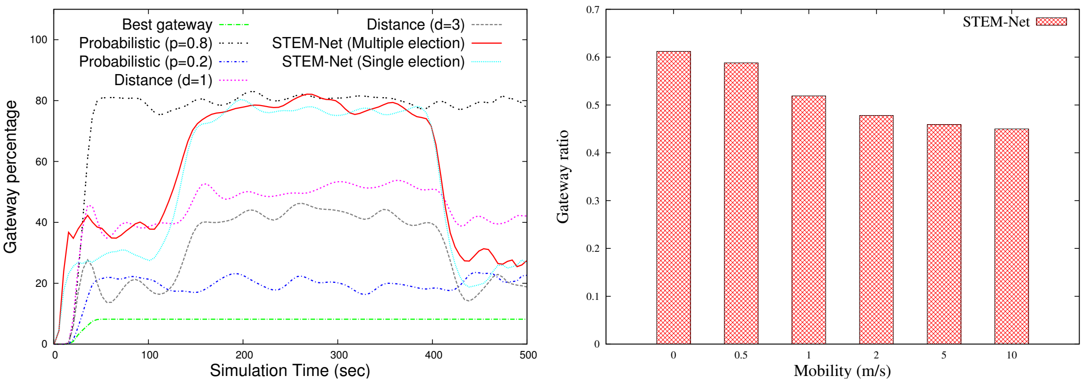
(b)

Figure 9.8: The GR value in a dynamic environment varying the number of SNs (in Figure 9.8(a)) and supporting single or multiple gateway election (in Figure 9.8(b)).

abilistic scheme (p=0.8), and higher than the other schemes. At the same time, the ND of STEM-Net gradually decreases while increasing the traffic load originated by the SNs (Figure 9.6(b)), and is much lower than Probabilistic scheme (p=0.8). This can be justified since STEM-Net adapts the network deployment by increasing the number of SN gateways, and hence reducing the average path length. This is also confirmed by Figure 9.7(a), which depicts the GR metrics. When the load increases, also the response function used by SNs to self-elect as a gateway increases, and this leads to higher values of the GR metric for STEM-Net. Viceversa, the GR values are almost constant for all the other schemes, in which the number of gateways is decided only on the basis of the number of SNs (fixed in this analysis), and not on the congestion level. Figure 9.7(b) completes the analysis by showing the EL metric. Again, schemes relaying on less SN gateways (i.e. Best gateway and Probabilistic) guarantee higher lifetime. However, considering both 9.6(a) and 9.7(b), STEM-Net guarantees the best trade-off between throughput and energy issues.

9.5.2 Dynamic and Parameter Characterization Analysis

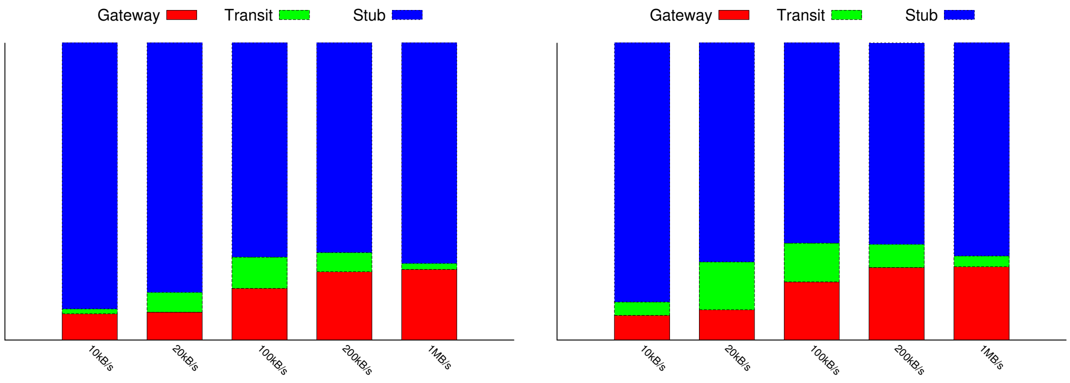
In Figures 9.8(a) and 9.8(b) is further illustrated the ability of STEM-Net to adapt the network deployment to dynamic load and network conditions. Specifically, is considered a network environment in which the number of SNs (i.e. N) is varied during simulation time (t). As before, each SN produces a constant load of 50 KB/s. At the simulation start, there are 40 active SNs in the scenario (i.e. N=40). From instant 100, is added one SN every 20 seconds, till N=60. From



(a) (b)

Figure 9.9: The GR value in a dynamic environment varying the load produced by each SN is shown in Figure 9.9(a). The GR of STEM-NET as a function of the SN mobility is shown in Figure 9.9(b).

t=600s, one SN leaves the network every 20 seconds till the initial network configuration (i.e. N=40) is reached at t=1000. Figure 9.8(a) depicts the GR metric over simulated time for the four schemes introduced earlier. In STEM-Net, the number of gateways is almost constant during all the simulation, due to fact that the load introduced by incoming SNs is low, and does not trigger the need of additional gateways. As a result, the GR metric shows the inverse behaviour than N (i.e. it decreases when N increases and viceversa). In Figure 9.8(b), is refined the previous analysis, by showing the GR metric for two different configurations of STEM-Net, when single or multiple gateway election mechanisms (Section 9.3.1) are used. The multiple election mechanism produces slightly higher values of the GR metric, since more than one SN may respond to the same STIMULUS message on the basis of the probabilistic filter of Equation 9.7, and thus may self-elect as gateways. However Figure 9.8(b) shows that such difference is minimal, since, regardless of the election mechanism used, the increasing number of SNs does not trigger the need of additional gateways, at least from the point of view of congestion (while active SNs with low energy values are still replaced, due to the $E(i)$ component of Equation 9.3). In Figure 9.9(a) is depicted the GR metric for a second dynamic network environment, in which the overall number of SN is fixed (and equal to 40), while the load produced is varied by each SN during the simulation time. More specifically, at simulation start, each SN produces a constant load equal to 10 Kb/s, which is increased to 1 Mb/s from t=50 till t=200, and again reduced to 10 Kb/s for $t > 400$. The dynamic variable load models a configuration in which the SNs switch to a different (safety-related) application, or modify the amount of data they produce for first responders (e.g. multime-



(a)

(b)

Figure 9.10: The STEM-NET role distribution is shown in Figures 9.10(a) (routing protocol of Section 9.3.2 is used) and 9.10(b) (basic AODV is used).

dia traffic). From Figure 9.9(a), it is possible to observe that: (*i*) for $t > 100$, the GR value increases in STEM-Net as a consequence of the additional traffic load, which determines higher values of the C_{cell} component of Equation 9.3, and in turn triggers the creation of new gateways, (*ii*) when the load decreases, the number of gateways in STEM-Net is automatically scaled down, (*iii*) the Single and Multiple Election mechanisms produce on average the same number of SN gateways under the same traffic load, however the Multiple Election scheme guarantees a faster reaction and protocol adaptation to the network dynamics. In the analysis performed so far, it is assumed a stationary deployment of the emergency network, i.e. without considering the impact of SN mobility. Figure 9.9(b) depicts the GR metric in a configuration with 30 SNs (i.e. N=30), moving according to the `MassMobility` model of OMNeT++. The maximum speed (in m/s) of the SN is varied on the x-axis. The GR metric decreases when increasing the average speed of the SNs. This is due to the fact that each SN can overhear more gateways in its neighborhood, and add them to the local GL data structure described in Section 9.2.3, leading to a possible decrease of the response function of Equation 9.4. An additional mechanism is required to refresh the GL structure, so that each SN can detect when a gateway is moving out of the threshold distance (h hops).

Figures 9.10(a) and 9.10(b) illustrate how the role selection is performed in STEM-Nwt, considering the three roles foreseen by the architecture. Is considered a configuration with 30 SNs, and the traffic load is varied produced by each SN (on the x-axis of both Figures). Are considered two different configurations of the STEM-NET architecture. Figure 9.10(a) refers to the full configuration of STEM-Net considered so far, while Figure 9.10(b) considers the case in which the stimulus-response mechanism is disabled for routing (Section 9.3.2), and the clas-

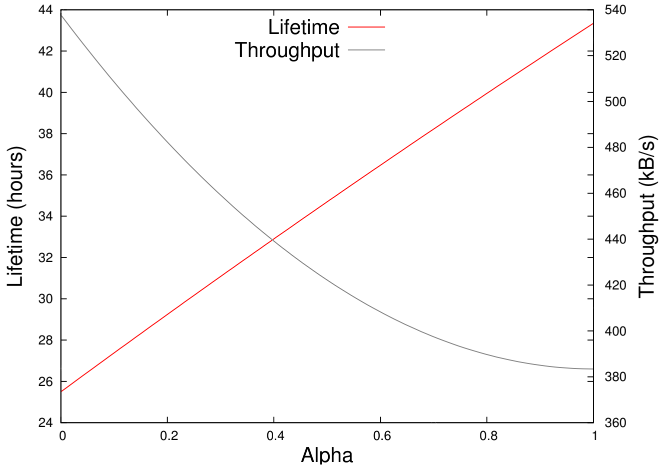


Figure 9.11: The impact of α and β on system performance is shown in Figure 9.11.

sical AODV is used instead. Both the Figures show these trends: *(i)* the number of SN gateways increase with the network load according to the motivations already provided before, and *(ii)* the number of SN relays is intrinsically dependant on the number of gateways. More specifically, the number of SN relays increases when the load is below the 100 KB/s, since multi-hop routes are required to reach the additional gateways created. However, the more gateways are determined, the less is the average length of the route, and this explains the reduction of relays for high load values (e.g. 1 Mb/s). Moreover, comparing Figures 9.10(a) and 9.10(b), it is evident that the leftmost configuration creates less relay SNs. This is reasonable, since when using the stimulus-response mechanism (Section 9.3.2), only a fraction of the potential receiver of an AODV `REQUEST` message will forward it, and more specifically only the nodes with energy and congestion values below the threshold.

Finally, Figure 9.11 clarifies the role of the α and β parameters on the STEM-Net deployment. It is considered a configuration with 40 active SNs, each transmitting 50 KB/s. On the x-axis, is varied the α value (β value is simply 1-α). On the same Figure, is plotted two lines corresponding to the goal functions of the stimulus-response model, i.e. network lifetime (on the x-y axes) and network throughput (on the x-y2 axes). The trade-off between the goal functions is made evident by the fact the throughput maximization is achieved with α=0, which is also the configuration providing the lowest lifetime values. The correct setting of α and β is outside the scope of the paper, and it is generally considered an input parameter, provided by the scenario administrator (if any) or by the user applications. However, the previous analysis shows that -based on the desired goal function- the STEM-Net self-deploy in order to maximize the requested metric (NL or NT).

Part IV

Tools for design and performance analysis

Chapter 10

Simulators for networks and controlled mobility systems

We can state that simulation allows the analysis of a system's capabilities, capacities and behaviors without requiring the construction of or experimentation with the real system [203]. This statement is of remarkable concern when the system under analysis deals with numerous and complex components whose cost is non-negligible. One possible example of expensive equipment consists in networks of Unmanned Aerial Vehicles. To make an experiments (to analyse for example a network protocol) worth and meaningful, in fact, we should consider many flying vehicles at the same time, which would make the operations costly and would introduce safety problems. for this reason the simulative approach is more viable and useful before any kind of prototyping development phase.

In the specific context of communication protocol design and control system definition in a UAVs fleet, the current literature presents many existing solutions that could be used to separately simulate a communication network and a formation control of UAVs. However, when the level of detail needs to be mixed, as for a packet-level simulation regarding the networking part and an accurate simulation of control surface with significant flying models for the control part, the number of existing simulative tools that can be used decreases significantly. The literature for simulation systems that can run at the same time networking and control methods with high level of details is scarce. From a networking simulation standpoint there is, in fact, the need to have a node movements rendering that is the most accurate possible. In the field of UAV's wireless communication, the technology available is dependent on radio transmissions and thus subject to interference figures strictly dependent on relative movements of each communication pairs.

For these reasons, in this Chapter is introduced a novel approach into the simulation of networked flying robots that integrates two existing solutions and leverage both their own capabilities and the validation already carried out upon them. More in detail, will be devised an architecture capable of integrating the *FL-AIR* simulator [204] with a mainstream network simulator, NS-3 [205]. It is leveraged the capability of *FL-AIR* and *NS-3* to: (i) be able to simulate real-time systems and (ii) be deployed on top of the networking interfaces of real Linux systems. In this way, the two components of the proposed architecture can exchange data and be able to use each others' strong-points to simulate a wide spectrum of complex scenarios in the context of UAVs fleets. In this Chapter is proposed a simulation architecture and its related software and is proved the system capabilities to render both networked control systems [206, 207] and distributed networked control systems [208].

10.1 Actual simulation tools

In this section are mainly listed separately network simulators and control simulators. This is because the literature of both network and control system simulator is wide but, concerning only integrated solutions, it is indeed scarce. For each part are showed the most interesting existing solutions. In this way are made clear the choices that are made concerning the proposed system's components and the faced design challenges.

10.1.1 Flying Drones Simulators

Two main possibilities arise from the literature in the context of UAV simulators. The first involves pilot-training software suites like [209], [210], [211], which are *classical* flight simulators that implement realistic flying models. The number of instances of this kind of software is increasing quickly, thanks also to the new discipline of *drone racing* and its necessity to train pilots. While they can provide a realistic flight model and good dynamic behavior, the mentioned simulators are not capable to simulate an autonomous flight as they are tailored to human interaction. Moreover, they are often distributed as closed source software, thus it is quite impossible to adapt them to simulate autonomous fleets of UAVs performing specific missions.

In the field of robotics, instead, open source and customizable specific UAV simulators are easier to find. The two most popular are Gazebo [212] and Morse [213]. Both are well documented and can be integrated with robotics middleware like the Robot Operating System (ROS) [214]. Unfortunately, ROS is not oriented towards real time simulation and is not well suited for developing low level control laws for UAVs. At the same time, Gazebo is capable of interfacing towards popular flight controllers like APM [215] or Pixhawk [216], usually not available on Morse. Both are capable to simulate fleets.

Despite the advantages of Gazebo over others simulators, this study will focused on *FL-AIR*, a simulator developed at the Heudiasyc laboratory (see Section 10.2.1 for more details). The main advantage of this simulator lies in its extreme lightness in terms of resource utilization, which make it perfect for scalable environment. As it will be described in Section 10.2.3, *FL-AIR* is tailored to low-level control of UAVs and it also perfectly integrates real robots. Another advantage of using this simulator is the easiness of the transition from simulation programs to real experiments and prototypes.

10.1.2 Network Simulators

The network simulation software can be classified in two large categories: commercial or open-source software. Some of the most important commercial software suites that have been thoroughly exploited in scientific literature are the *Qual-Net* [217] and OPNET suites [218]. For evident reasons in this work will be preferred the use of *open source* simulation solutions. In this perspective, historical works on network simulation are represented by the projects of GloMoSim [219], NS-2 [220] and JiST/SWANS [221], all of them no longer under active development. Actively maintained project are instead the OMNeT++ [222], wsnet [223] and NS-3 [224] software suites. For the aim of this work, the most suitable network simulator needs to fulfil the following characteristics: (i) a strong scientific literature background; (ii) the possibility of configuring protocols at all levels of ISO/OSI protocol stack; (iii) the capability to simulate at any level of detail, specifically at the packet-level and (iv) the ability to operate in real-time interfacing directly with the operating system real devices.

According to these needs the NS-3 network simulator is chosen among the others since it fulfil all the described requests.

10.1.3 Others coupled simulators

In literature are presents others attempts of building bidirectionally coupled simulators. However, these attempts are not regarding the simulation of flight drones. In the road vehicular mobility, for example, there is the integration of the Simulation of Urban MObility (*SUMO*) simulator with both NS-3, through the iTETRIS Control System *(iCS)* [225], and OMNeT++ through the *Veins framework* [226]. These simulators are specifically designed for traffic simulator and analysis of V2X communications. The proposed solution of integration between the two simulators, NS-3 and *FL-AIR*, differs from these two cited vehicular simulator because we want to avoid the adding of others external modules (like iCS) that would increase the complexity of the general system. Another bidirectionally coupled simulator is RoboNetSim [227]. The architecture of this simulator is similar to the one proposed in this Chapter. In fact, it propose the integration between a network simulator (NS-3) and a robot simulator (ARGoS [228]). Differently from ARGoS, FL-AIR has the attractive characteristic of an easy portability with real devices. With this property, it is possible to generate, with the same implemented code, both the simulator program and the executable program to execute directly on the real device.

10.2 Architecture definition

For the proposed coupled simulator proposal we leverage the virtual interfaces that the Network Simulator 3 (NS-3) is capable of manage, that make possible for it to interface directly with the operating system and software-simulates network devices in real-time. The coupled simulator architecture is capable to use these interfaces to connect *FL-AIR*'s UAVs. In this way, all the data traffic exchanged among the autonomous robots passes through NS3 working in real-time mode.

10.2.1 FL-AIR

As stated before, is used a simulation framework called *FL-AIR*, currently being developed at [229]. This tool has been created to ease integration and testing of researchers' algorithms. It is composed of different autonomous applications running on Linux systems. One of them is the *World Simulator*. Figure 10.2 summarizes *FL-AIR*'s internal functioning in the case of two simulated UAVs. Each different composing application is represented by a block and is connected to a *Ground Control Station* through UDT sockets [230]. The *Ground Control Station* is *FL-AIR*'s monitoring and multiplexing block. Its purpose is both to display information and implement commands coming from other blocks.

The *FL-AIR* simulator relies heavily on socket communication to implement the data excanging among its components and, most important of all, among the UAVs, being them simulated or real. The *World Simulator* subcomponent is, in fact, a specific program which manages the dynamics of the UAV and provides the implementation of virtual sensors and actuators. It can also contain an interface to an already deployed *Optitrack* system [231]. Since the communication occurs between two different block via socket communication, these programs can be either UAVs simulated as processes in the same machine or application running on real robots, relaying commands and sensed data.

FL-AIR implements an hardware abstraction layer by means of interface classes to each kind of sensors and actuators; thus the only difference between a simu-

lation program and a real drone is the use of a specific implementation of the interface (virtual or real one). UAV's state machine and securities check are also done by specific classes. The excanging of information between the UAVs and the World Simulator is made via shared memory. The shared memory is intended as the one provided by operating system standard libraries. In this way the *World Simulator* can compute the UAV state of through its discretized dynamical model [232]. The model output is used to update UAV status in the virtual 3D world: it handles collisions and keeps track of positions that are then sent back to the simulated UAVs control models. It is to note that multiple UAVs can be simulated using this scheme.
The £d environment is provided by the Irrlicht 3D engine [233] (see Figure 10.1) for visualization and also to simulate sensors that need interaction with the environment (cameras, lidars, ultrasonic sensors, etc). The *World Simulator* is also compatible with Xenomai [234] (a real time extension to the Linux kernel), which allows to have strict bounds on the dynamic model outputs and strict constraints on the synchronization among simulator modules. For the results shown in this Chapter, the display part of the simulator operates in non real-time mode for ease of prototyping.

In *FL-AIR*, the UAVs (being either real or simulated) communicate directly using UDP sockets. This means that all the traffic generated by the very UAV's network interfaces (simulated or real) is routed through the computer hosting the *Ground Control Station* and, optionally, the *World Simulator*. Furthermore, *FL-AIR* supports network remote controls to manually pilot the drones.

10.2.2 Network Simulator 3

NS-3 [224] is an open-source project of a discrete-event network simulator in which the simulation core and models are implemented in C++. The main objective of NS-3 is to provide models of how packet data networks operate and perform. Furthermore, it allows its users to perform studies that are difficult or even not possible to implement in real systems, thanks to its capability to be arbitrarily extended.

As said before, NS-3 can easily create and manage virtual devices in its host machine. The NS-3 network simulator has the ability to work in the so called *real-time/emulation* mode. In this mode, the simulator can exchange packets in real-time (the best possible it can do) with the outside world through virtual network interfaces. Hence, it can generate traffic towards real networks and drive a simulated network with packets coming from real world. As the UAVs communication is done in *FL-AIR* by using sockets, it is easy to route the packets through NS-3 virtual interface in place of the local ones. To allow this routing a set of tools from the Linux Kernel is used: *Linux containers* (*LXC*), *TAP devices* and

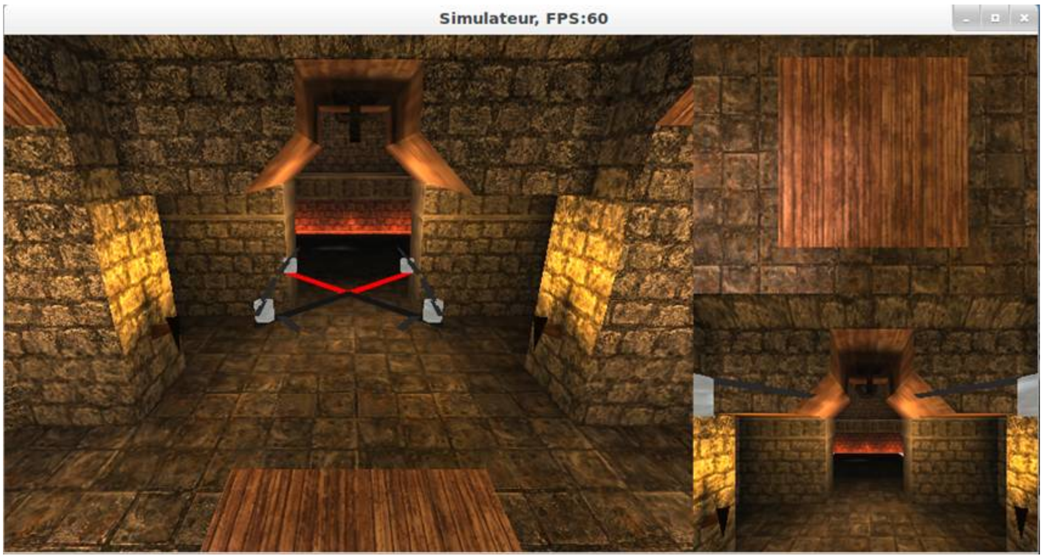


Figure 10.1: Overview of *Flair* 3D environment. Two virtual cameras are displayed on the right.

network bridges.

- *LXC (Linux Container)* [235] is an operating-system-level virtualization method for running isolated Linux systems (containers) on a control machine. The main characteristic of LXC is the ability to create a lightweight virtual machine with an environment as close as possible to a standard Linux installation but without the need for a separate kernel.
  Despite the appearance of increasing the resource burden of the architecture, the use of LXCs provide multiple advantages: (i) ease of installation, (ii) complete isolation of the application that will run into the containers and (iii) possibility to freely configure network devices. The containers will host the UAV application: the sockets the UAV application will create will be also automatically hooked by the *LXC* and become available to be routed.

- *TAP devices* are virtual network devices. These devices are different from the real network device due to the fact that the packets that travel through a TAP device are managed by user-space software.
  As will be explained later, the TAP devices will be managed entirely by the *Tap Bridge Module* of NS-3.

- A *network bridge* is a device that is used to join two or more network segments. A bridge behaves transparently like a virtual level-2 network *switch* where both real devices and virtual ones can be connected to it.

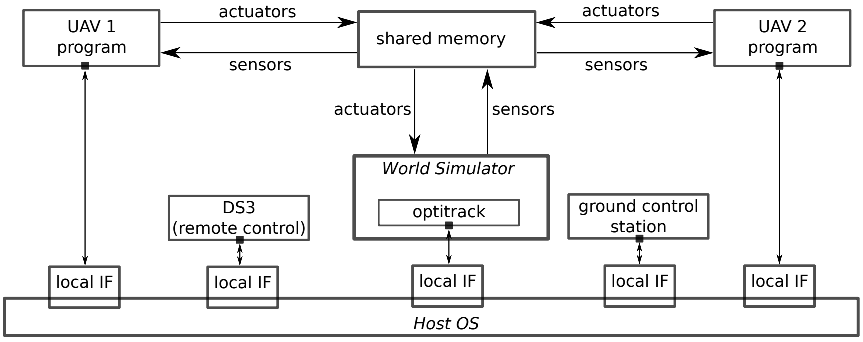


Figure 10.2: Overview of *Flair*.

10.2.3 Interworking

Now will be described the architecture that enable the connection between the *Flair* and the NS-3 simulator.

The proposed architecture is called *CUSCUS* (CommUnicationS-Control distribUted Simulator). *Flair* and NS-3 will be run in parallel within *CUSCUS* and set up according to the scenario to simulate.

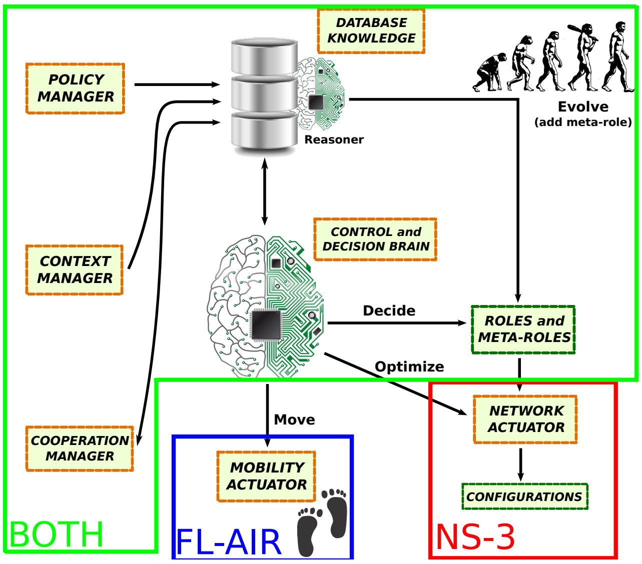


Figure 10.3: The Stem-Node architecture implemented in CUSCUS

Looking at the main architecture of a STEM-Node (see Figure 10.3 we can notice how the different modules of a STEM-Node will be implemented in CUSCUS. The *Network Actuator* is implemented in NS-3 because it will deal with the communication issues; the *Mobility Actuator* is implemented in FL-AIR since it is able to model very accurate mobility control models. Finally, the other modules can be implemented in both the simulators, depending on the characteristics of the simulated scenario.

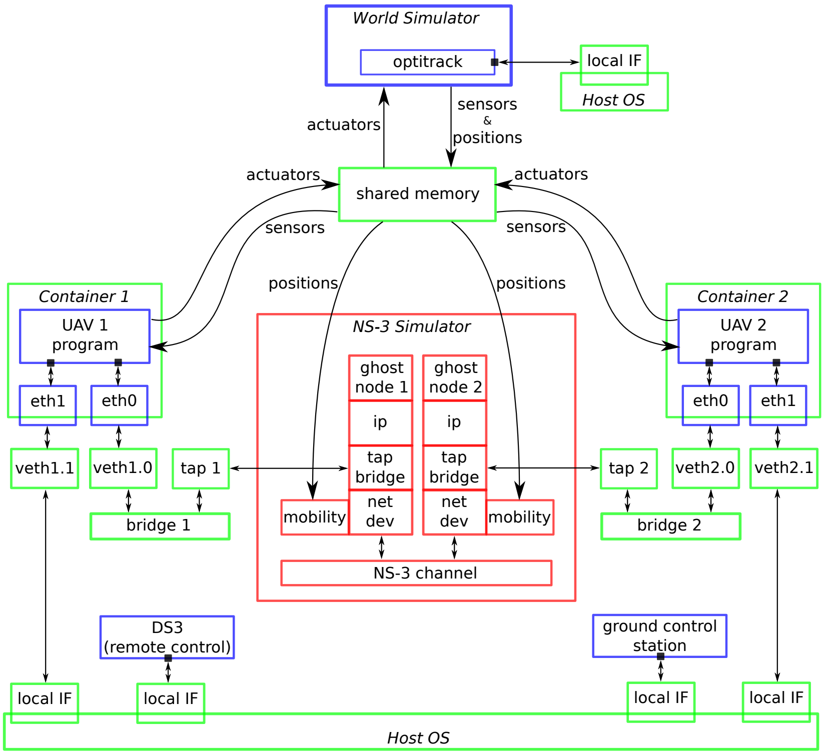


Figure 10.4: Overview of *CUSCUS* architecture. The *red* indicates the NS-3 simulator, the *blue* indicates the *FL-AIR* modules and the *green* parts indicate the host pc

As we can see from the main CUSCUS architecture pictured in Figure 10.4, each *LXC* will envelope a *FL-AIR*'s UAV application and intercept all the generated traffic. The *network bridge* connects the *LXC* and the *TAP* device. Finally the *TAP* device is the link between the host machine and the NS-3 module.

After creating *CUSCUS* as capable to intercept the traffic coming to and from *FL-AIR*, is exploited the *Tap Bridge* Module [236] on NS-3. This module is capable of integrating the *real* UAV host application into the NS-3 network host via the *TAP* interface. These are created on the host running NS-3 and all the traffic that is generated from and directed to the TAP device is intercepted by the

NS-3 module that will transparently connect the application with the simulated network. More specifically, there are three basic operating modes of this NS-3 module: *ConfigureLocal*, *UseLocal* and *UseBridge*. Their basic functionality are essentially identical but they differ on the way they create and configure the bridge devices. In this work is used the *UseLocal* mode that has the characteristic of using an existing TAP device previously created and configured by the user, in order to be more flexible in connecting the *NS-3* simulator with *FL-AIR*.

In Figure 10.4 is depicted the main architecture of the *CUSCUS* framework. Here we can easily notice the connection between the two simulators made by the *TAP-bridge-LXC* chain. The architecture's main features are:

- Each UAV control application i is loaded in its own container (*Container* i in Figure 10.4). In this way the application will run as if it is loaded on a real UAV. Since these applications and the *World Simulator* run on the same real host machine, they can still exchange data through the shared memory.

- Each simulated UAV has two components: once into the LXC container, that describes the control strategies defined in the UAV application and the other in NS-3 (as *ghost node* i) that describes the UAV's network behavior.

- In order to let NS-3 be aware of each UAV position and orientation, the *World Simulator* will update constantly the position and orientation of each UAV in the shared memory of Figure 10.4. In this way, the mobility model implemented in NS-£ can maintain update the node position and orientation just reading from the shared memory.

In conclusion, a closed loop is maintained: while the UAVs move in *Flair*, their position is relayed to NS-3 and the communication channel model is modified accordingly. The packets created by the UAVs will travel through the *TAP-bridge-LXC* chain toward NS-3.
In the architecture depicted in Figure 10.4 we can notice that the *Ground Control Station* and the *remote controller* modules are placed outside NS-3. Moreover, a second network interface (*eth1*) is added to each LXC to connect these modules (Ground Control Station and remote controller) with the UAV application without passing through NS-3. This architectural choice is only for simplify the analysis made in Section 10.3. These modules can indeed be easily put inside NS-3 to simulate also the communication issues between the UAVs and the remote controllers. However, this is an example of the *CUSCUS* architecture flexibility, that allows the user to completely customize its own environment.

Looking at the *CUSCUS* main architecture, it can be noticed the architectural design of the system that results from the introduction of a *TAP-bridge-LXC* chain for each UAV control application. Despite the apparent architectural complexity, it is important to point out that the delays added to the network packets are negligible (as we will see in the Section 10.3) as these last are directly managed by the host operating system, that will fast route them through the chain.

10.3 CUSCUS performance

To evaluate the feasibility to deploy *CUSCUS* will be performed a set of experiments to test and measure some benchmarks. For the experiments is used a Dell XPS 8500 workstation with an Intel® Core™i7-3770 CPU @ 3.40 GHz and 16 GB of RAM memory. The entire *CUSCUS*, comprised of NS-3 and *FL-AIR* runs on this *host* machine. The results have been averaged over 20 simulation runs.

In the experiments are analysed the scalability of the CUSCUS simulator with respect of the number of simulated UAVs and the impact of the delay overhead introduced by the proposed architecture. The FL-AIR UAV application used is a formation control algorithm that enable the creation of a circle of UAVs that move arount the center of the scenario. The distributed control system employed on each UAV takes its own position and orientation from a simulated *Optitrack* system and tries to maintain the same distance between all the nodes while they circle. As the number of nodes increases, the UAVs will arrange themselves at the vertexes of a polygon with an increasing number of sides. The UAVs keep a fixed altitude of 10 m for the whole simulation.

On the NS-3 module, each UAV is equipped by a single IEEE 802.11 interface and hence is used a Ricean fading model, as for UAVs the signal on the LOS path is much stronger than the one on indirect paths. It is assumed no packet fragmentation and that the nodes always stay in each others' connection range.

In the experiments the robots sample their position and broadcast it periodically, each T_b seconds. For the analysis are defined two kind of delays: the *architectural* delay and the *network* delay. The former is the overhead introduced by CUSCUS that the packets experience when they are routed through the architectural components; the latter is due instead to the simulated network behavior.

A first index that can be used to evaluate the performances of *CUSCUS* is composed by the physical resource used by the *host* during the simulation. In Figure 10.5 are depicted the percentages of CPU and RAM usage increasing the number of nodes. As we can see, the trend of the resource used by *CUSCUS* is linear in respect of the simulated UAVs number. This result shows the *scalability* of *CUSCUS*, which is able to exploit the available resources efficiently.

The second test is done in order to analyse the architecture, i.e. the *architectural* delay introduced by *CUSCUS* in the *TAP-bridge-LXC* chain (see Section 10.2.3). This measure is important as it can estimate the time delays introduced

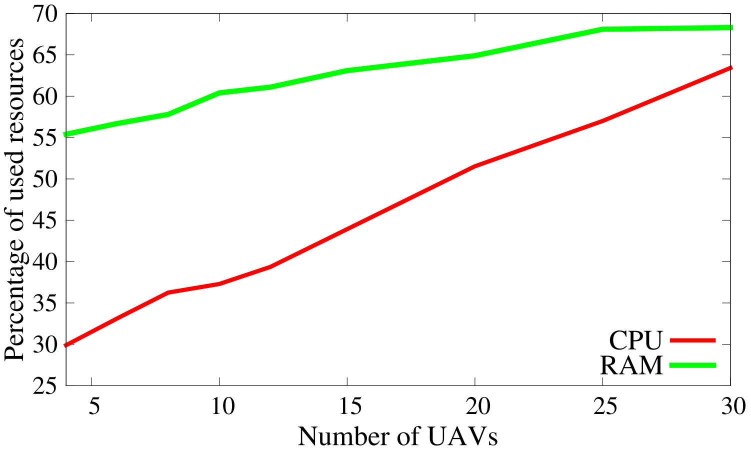


Figure 10.5: Resource Usage at the *host* varying the number of UAVs.

by the *CUSCUS* structure. In Figure 10.6 is shown the delay [μs] the packets experience in the path from *Flair* to NS-3. The broadcast time T_b is set to $50ms$ for this experiment. We can notice from the Figure 10.6 that the *architectural* delay introduced is constant and it stands around $70\mu s$. This result shows that the delay introduced solely by *CUSCUS* is negligible and stable with respect to the UAVs number, i.e. the traffic generated by the simulation scenario does not influence the variation of the *architectural* delay.

The last experiment shows the *suitability* of the CUSCUS architecture for the study and analysis of distributed networked control system. For the experiment are executed a series of tests, modifying the number of UAVs and the T_b value in order to study the impact of using a simulated communication network for the exchange of control system's messages. The scenario used is a modified version of the precedent one: the nodes will not retrieve the positional information of the other UAVs directly from the Optitrack system. They will instead use the one their neighbours have broadcasted through the simulated NS-3 channel. In this way are measured the error introduced in the formation control inside *Flair* due to the *network* delays.

In Figure 10.7 we can see a screenshot of the CUSCUS simulator in execution. In this experiment four drones are simulated in order to test the formation control algorithm. In the same Figure, is shown also the packet exchanging between the drones (the Linux terminal in the bottom-right part of Figure 10.7). In this particular screenshot it can be notices the send and the receive process executed in *Drone_0* and *Drone_1*. The *terminal* prints are executed directly on the containers where the UAV programs are running. The data packets, that are broadcasted by each drone, travel via the *eth0* interface of actual container (see Figure 10.4), pass

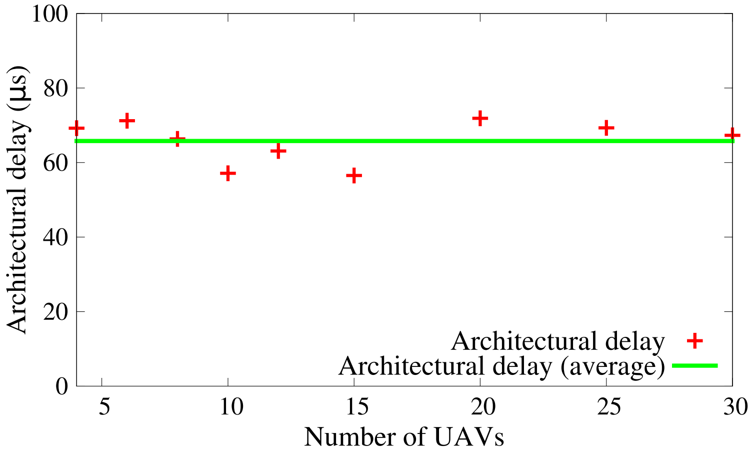
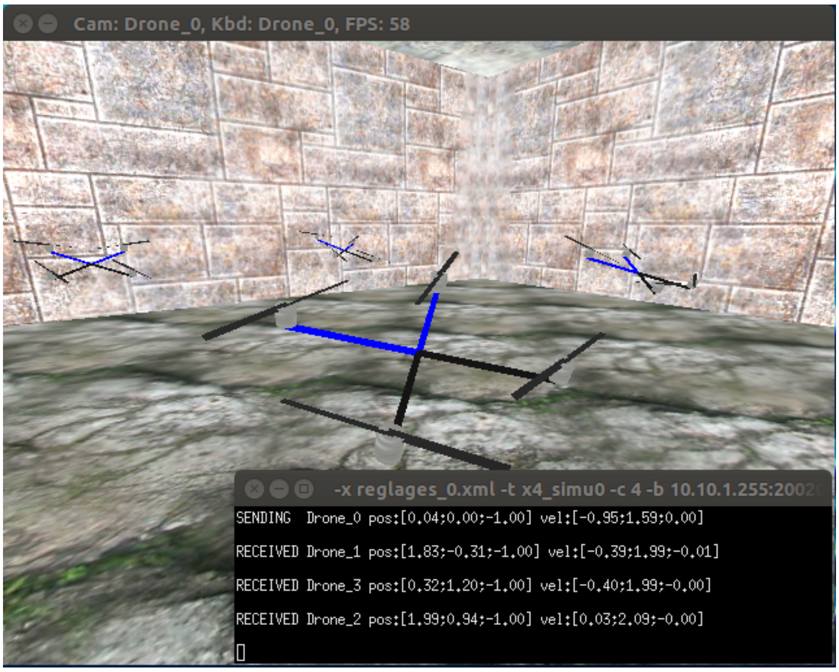


Figure 10.6: Architectural delay of CUSCUS varying the number of UAVs.


Cam: Drone_0, Kbd: Drone_0, FPS: 58
-x reglages_0.xml -t x4_simu0 -c 4 -b 10.10.1.255:2002

```
SENDING  Drone_0 pos:[0.04;0.00;-1.00] vel:[-0.95;1.59;0.00]
RECEIVED Drone_1 pos:[1.83;-0.31;-1.00] vel:[-0.39;1.99;-0.01]
RECEIVED Drone_3 pos:[0.32;1.20;-1.00] vel:[-0.40;1.99;-0.00]
RECEIVED Drone_2 pos:[1.99;0.94;-1.00] vel:[0.03;2.09;-0.00]
```

Figure 10.7: Screenshot of a formation control algorithm simulated in CUSCUS

through the simulated network created by NS-3, and finally reach the destination drone via, again, the *eth0* interface.

Let us define *error* $[m]$ as the distance between the *actual position* of the simulated UAVs and their *reference position* in the formation. The *reference position* is the position that nodes should be at and is used by the control algorithm to guide the UAV. Figure 10.8 shows this *error*, as the result of these tests. In the case where the UAVs get the position of the neighbours UAVs from the Optitrack system ($T_b = 0ms$ in the figures), the *error* is reduced to minimum and it is due only to the control algorithm itself. Instead, when the simulated UAVs take the information from their neighbours, there is a *delay* introduced by the network that destabilize the control algorithm.
From these experiments it can be inferred the following conclusions: (i) the T_b value increase strongly impacts the formation error, whereas (ii) the number of UAVs does not affect this error. The former conclusion comes from the fact that with big T_b, the control algorithm is running a distributed control system with more and more outdated information about the neighboring nodes' positions. The latter comes from the fact that, in this particular case, communications issues such as packets collisions and, hence, packet retransmissions, have negligible impact on the wireless communication network. It is to remark that the collision probability is very small since the exchanged packets are also very small. They are composed by only the sender node's position and orientation.

In conclusion it can be stated that the *CUSCUS* framework is able to execute reliable distributed networked control system simulations by keeping time overheads constant when the number of simulated UAVs varies. Hence, it enables the user to study and analyze practical instances of this kind of control systems.

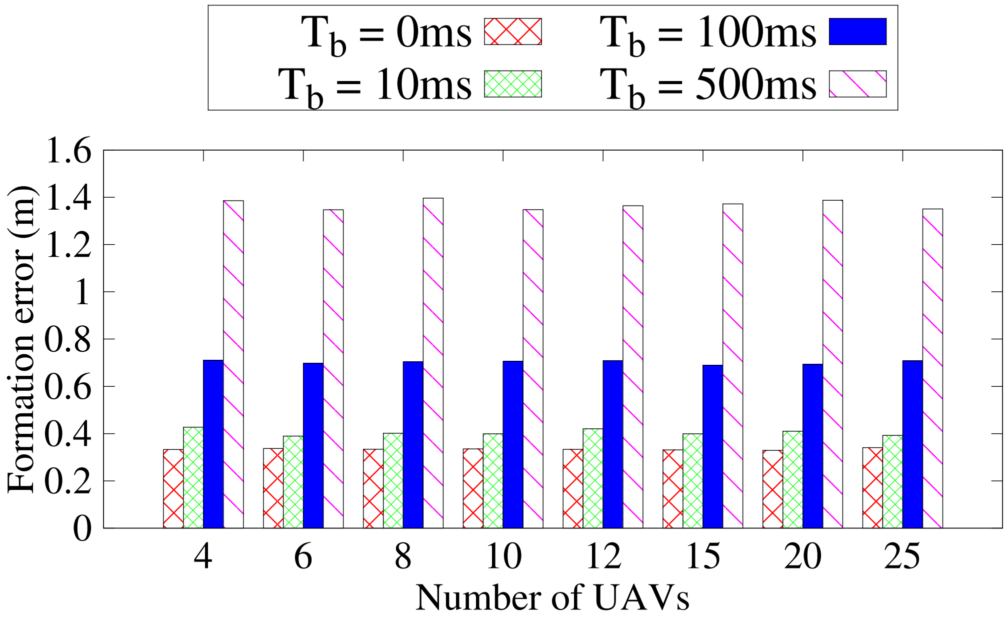


Figure 10.8: Formation Error varying the number of simulated UAVs and T_b.

Part V

Conclusions

Chapter 11

Conclusions

The actual deployed technologies for public safety networks are not able to face the issues that arise in the aftermath of a natural or human disaster. In fact, the lack of communication infrastructures or the damage that these latter experience during the disaster, make the first recovery operations difficult to be fulfilled by the rescue teams.

11.1 Thesis results

In this thesis it is presented a new vision for next-generation public safety systems based on autonomous vehicles and spontaneous communication networks.

In the first part of the thesis the concept of staminal network (*STEM-Net*) is proposed, where the so called *STEM-Node* is able to modify its properties in order to adapt to the environment requests. These nodes are able to cooperate each other in order to implement system-wide behaviours and are capable of self-configure at multiple layers of the protocols stack on the basis of their hardware configurations assuming multiple network roles. Furthermore, a STEM-Node has autonomous mobility capability to modify and control the node's location. These special nodes can increase their capabilities by *evolution*: in this way a node is able to add new modules to its own possible functions set, so that possible newer issues can be managed by the node. The characteristics of *evolvability*, *adaptability*, *reliability* and *resilience* bring multiple advantages in the use of the STEM-Node architecture compared with more traditional approaches.
Given the ability of self-controlled mobility, the deployment of a swarm of unmanned vehicles brings new opportunities in the rescue operations for emergency scenarios. To deal with this powerful capability of self-movement, the vehicles must be supported by efficient mobility control systems that are able to move the vehicle in the best possible way. This is crucial for the *Unmanned Aerial*

Vehicles (UAVs), where the motion control systems play a very important role for the mission outcome. In fact, the mobility controller and the communication network links are strictly correlated. For this reason in this thesis is deeply analysed the literature about the *mobility-aware communication protocols* and the *communication-aware mobility controllers*. Then is proposed a new architectural stack layering, called *Networked Behaviours* layers to help the researcher in developing new systems that will be able to deal with both the communication and the mobility control issues.

The last post-disaster emergency events occurred around the worlds have highlighted that in these scenarios the available network communication infrastructure is unable to manage the effort needed in these situations. For this reason in this thesis is proposed an efficient distributed algorithm to be deployed on autonomous unmanned vehicles that is able to re-enstablish the network connectivity. This technique will give more opportunities to the rescue teams for cooperating and managing the emergency situation. The swarm of unmanned vehicles have two main functions: i) maintaining the network connectivity among the vehicles inside the mesh network and ii) help the search and rescue operations by looking for survivors in the disaster area by giving them the possibility of communication. With this system, the rescue teams will be able to have constant connectivity during the operations and to have direct contact with the survivors.

The proposed algorithm is derived from a nature-inspired technique using virtual spring forces that enable the vehicles to maintain the right inter-vehicle distance to both avoid network link disconnection and allow the scenario exploration with the aim of searching for survivors. To evaluate and validate the proposed algorithm, it is analysed and compared with a mathematically derived analytical method. This method is an approximated solution with an upper bound in the error compared with the optimal theoretical solution. The outcome of deep analysis and complete simulations is that the proposed distributed algorithm is comparable with the analytical one and hence is able to execute the task of re-establish the network connectivity in a way that is very close to the optimal solution.

To speed-up the rescue operations, the use of UAVs is undoubtedly one of the possible choices. These vehicles have, in fact, the ability of fast movements by fling above the disaster area. Furthermore, the aerial vehicles have the advantage of the line-of-sight for the communication link. One drawback of this technology is the fast drain of the battery energy. The autonomy of these kind of vehicles is in the order of ten of minutes. To overcome this limitation, this thesis proposes a distributed method of charging scheduling where the UAVs continuously swap from the active state, where they are acting as repairing unit in re-establishing the network connectivity, and charging state, where they recharge the battery energy in a specific recharge station. The proposed scheduling method investigated approaches to maximize the lifetime of the aerial mesh by considering a scenario where the UAVs can recharge their batteries by the contact with the ground charging stations. The proposed distributed scheduling algorithm ensures a guaranteed coverage area by the UAVs, while the network lifetime is maximized.

One of the main problem of emergency communication network is that the most common wireless technologies used by the end-user devices (WiFi, 3G/4G, ecc...) are in general crowded in the emergency scenarios. This issue is one of the causes that bring the communication networks unable to support the emergency operations. One of the most promising technologies that will overcome these problems are the *Cognitive Radio Networks* (CRNs) over *TV-White Spaces* (TVWSs). Those frequencies are very useful for rescue operations due to the low shadowing loss that effects these frequencies. However, to use these frequencies, the national regulators have decided that any user that wants to transmit over these frequencies must query a *geolocation spectrum database* that knows which are the free usable channels for each location. The informations stored in this databases are based only on analytical transmission models that can be inaccurate for complex environments. In this thesis, hence, is proposed a system that is able to increase the precision of the data stored in the database by making real frequency scans using a fleet of unmanned aerial vehicles that make real measurements in the scenario giving more capillarity precision to the data stored into the geolocation spectrum database. The proposed distributed exploration/scanning method consider the use of unmanned vehicles (called *Unmanned Aerial Scanning Vehicles*, UASVs) that are able to concentrate the scanning operations especially where the shadowing loss is high and hence more scanning operations are needed to extract more accurate scanning reports. The developed algorithm is based on a field-based self-coordination method that envisage the use of attractive/repulsive field-forces. These forces are derived from the actual shadowing characteristics making this technique self-adaptive to the exploring scenario. The analysis and the simulations have shown the capability of the proposed method to make accurate free/busy decisions of the scanning frequencies and, furthermore, is able to build an accurate shadow map of the scanned area.

The survivors of a disaster event are generally isolated from the main communication networks. However, given the plethora of smart devices that permeated the end-used market, we can assume that each survivor is equipped with a smart device that is able to create a spontaneous communication network. In this way the possibility of having a connection link towards the external networks (Internet or the rescue team network) will increase considerably. In this thesis is proposed the establishing of a spontaneous network by using the STEM-Net architecture where each end-user smart device can act not only as a simple *end-user device*, but also as *router* to help other survivors to communicate inside the spontaneous network, or as a *gateway* to enable the users to communicate towards the external networks. The proposed algorithm is inspired from the stimulus/response method used in the *swarm intelligent* research field, where the devices have a different intensity of stimuli of doing a specific task, while having a threshold that limits the execution of that specific task. The analysis and the simulations demonstrated the ability of this method in increasing the lifetime of the whole spontaneous network (the end-user device are battery powered) while guaranteeing enough data throughput for the emergency communications.

Finally, in this thesis is developed a simulator tool for the research and the analysis of complete networked behavioural systems that include the design of both mobility control systems and communication protocols. In fact, as said before, the spatial motion has high impact on the communication network performance, and vice-versa. For this reason, is needed a simulator tool that is able to integrate both the communication protocols and the mobility controller design. The developed simulator, called *CUSCUS*, integrates in a single tool the Network-Simulator 3 (*NS-3*) and the control system simulator (*FL-AIR*). This tool is a lightweight real-time simulator where the two components of the proposed architecture can exchange data and are able to use each others' strong-points to simulate a wide spectrum of scenarios in the context of UAVs fleets.

11.2 Future works

The research field of use of unmanned vehicles in emergency scenarios is relatively young. There are in fact some arguments that must be deeply analysed:

- **Deployment agility**. The deployment speed and the adaptivity of the deployed system is of fundamental importance. The first hours of the operations are in fact crucial. For this issue, the use of UAVs is the main promising technique. However the ability of these kind of vehicles in creating a communication network is still an open issue.

- **System lifetime**. After the deployment of the vehicles swarm, this system must supply the rescue team for all the time needed for the rescue operations. Faster recharge mechanisms must be developed in order to cope this issue.

- **Standardization of the communication technologies**. Different public safety agencies use different communication technologies over different wireless communication frequencies. A standardization operation must be started in order to allow fast communication among the different organizations that operate in the emergency field.

- **Heterogeneity of the vehicles**. A single type of vehicle could be insufficient to support the rescue operation in a specific environment. A system that would operate in such conditions must support the heterogeneity of the vehicles in order to exploit all the different vehicles characteristics: agility and fast deployment for *Small UAVs* (SUAVs), fast deployment and long lifetime for *Big UAVs* (BUAVs), long lifetime and direct contact with the responders for the *Unmanned Ground Vehicles* (UGVs).

- **Technologies integration**. Analyse the full integration between the spontaneous networks of survivor smart devices and the temporary network of autonomous self-organizing unmanned vehicles set up by the public safety agencies.

- **Real test beds**. After the design of new public safety communication network systems, a real test bed on field must be executed to validate the developed system.

In conclusion, the work presented in this thesis wishes for the use of more and more enhanced technologies during emergency scenario in order to save the highest number of lives, reducing the rescue operation time by helping the rescue teams in their important work.

Bibliography

[1] Y. Sun and K. Chowdhury, "Enabling emergency communication through a cognitive radio vehicular network," *IEEE Communications Magazine*, vol. 52, pp. 68–75, oct 2014.

[2] A. Gorcin and H. Arslan, "Public Safety and Emergency Case Communications: Opportunities from the Aspect of Cognitive Radio," in *2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 1–10, IEEE, oct 2008.

[3] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, pp. 2127–2159, sep 2006.

[4] S. Nagaraj and F. Rassam, "Cognitive Radio in TV White Space," in *Wireless Research Collaboration Symposium (NWRCS), 2014 National*, pp. 84–89, 2014.

[5] G. Villardi, G. Thadeu Freitas de Abreu, and H. Harada, "TV White Space Technology: Interference in Portable Cognitive Emergency Network," *IEEE Vehicular Technology Magazine*, vol. 7, pp. 47–53, jun 2012.

[6] FCC, "Unlicensed Operation in the TV Broadcast Bands - THIRD MEMORANDUM OPINION AND ORDER," 2012.

[7] Ofcom, "TV white spaces - A consultation on white space device requirements," 2012.

[8] J. P. Davis, K. M. Eisenhardt, and C. B. Bingham, "Developing theory through simulation methods," *Academy of Management Review*, vol. 32, no. 2, pp. 480–499, 2007.

[9] J. Fedorowicz, S. Sawyer, C. Williams, M. L. Markus, M. Tyworth, D. Jacobson, S. Gantman, M. Dias, and A. Tomasino, "Design Observations Regarding Public Safety Networks," in *Proceedings of the 12th Annual*

International Digital Government Research Conference: Digital Government Innovation in Challenging Times, dg.o '11, (New York, NY, USA), pp. 272–281, ACM, 2011.

[10] G. Iapichino, D. Câmara, C. Bonnet, and F. Filali, "Public Safety Networks," in *Handbook of Research on Mobility and Computing*, pp. 267–284, IGI Global.

[11] M. M. Sohul, M. Yao, X. Ma, E. Y. Imana, V. Marojevic, and J. H. Reed, "Next generation public safety networks: A spectrum sharing approach," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 30–36, 2016.

[12] G. Baldini, S. Karanasios, D. Allen, and F. Vergari, "Survey of Wireless Communication Technologies for Public Safety," *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 619–641, 2014.

[13] A. R. McGee, M. Coutière, and M. E. Palamara, "Public safety network security considerations," *Bell Labs Technical Journal*, vol. 17, no. 3, pp. 79–86, 2012.

[14] M. Casoni and A. Paganelli, "Security issues in emergency networks," in *2011 7th International Wireless Communications and Mobile Computing Conference*, pp. 2145–2150, 2011.

[15] P. Stavroulakis, *Terrestrial trunked radio - TETRA: A global security tool.* 2007.

[16] M. Nouri, D. Ball, M. Rayne, V. Lottici, R. Reggiannini, and M. Carta, "TEDS: a High Speed Digital Mobile Communication Air Interface for Professional Users Part I: Overview of Physical Layer," in *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, pp. 959–963, IEEE, apr 2007.

[17] S. Glass, V. Muthukkumarasamy, M. Portmann, and M. Robert, "Insecurity in public-safety communications: APCO project 25," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, vol. 96 LNICST, pp. 116–133, 2012.

[18] G. Iapichino, C. Bonnet, O. Del Rio Herrero, C. Baudoin, and I. Buret, "A mobile ad-hoc satellite and wireless mesh networking approach for public safety communications," in *10th International Workshop on Signal Processing for Space Communications, SPSC 2008*, 2008.

[19] T. Doumi, M. F. Dolan, S. Tatesh, A. Casati, G. Tsirtsis, K. Anchan, and D. Flore, "LTE for public safety networks," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 106–112, 2013.

[20] R. Favraud, A. Apostolaras, N. Nikaein, and T. Korakis, "Toward moving public safety networks," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 14–20, 2016.

[21] E. Yaacoub and O. Kubbar, "On the performance of distributed base stations in LTE public safety networks," in *IWCMC 2012 - 8th International Wireless Communications and Mobile Computing Conference*, pp. 927–932, 2012.

[22] R. Fantacci, F. Gei, D. Marabissi, and L. Micciullo, "Public safety networks evolution toward broadband: sharing infrastructures and spectrum with commercial systems," *IEEE Communications Magazine*, vol. 54, pp. 24–30, apr 2016.

[23] R. Grodi and D. B. Rawat, "UAV-assisted broadband network for emergency and public safety communications," in *2015 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2015*, pp. 10–14, 2016.

[24] A. Trotta, M. Di Felice, L. Bedogni, and L. Bononi, "Re-establishing network connectivity in post-disaster scenarios through mobile cognitive radio networks," in *2013 12th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, pp. 18–25, IEEE, jun 2013.

[25] A. Trotta, M. Di Felice, L. Bedogni, L. Bononi, and F. Panzieri, "Connectivity recovery in post-disaster scenarios through Cognitive Radio swarms," *Computer Networks*, aug 2015.

[26] R. R. Murphy, "A roadmap for the adoption of unmanned systems into public safety," in *2013 IEEE International Conference on Technologies for Homeland Security (HST)*, pp. 350–353, IEEE, nov 2013.

[27] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[28] D. T. Okereafor, U. Diala, N. Onuekwusi, L. O. Uzoechi, and G. Chukwudebe, "Improving security and emergency response through the use of

unmanned vehicles," in *2013 IEEE International Conference on Emerging & Sustainable Technologies for Power & ICT in a Developing Society (NIGERCON)*, pp. 263–269, IEEE, nov 2013.

[29] C. Wargo, G. C. Church, J. Glaneueski, M. Strout, and Others, "Unmanned Aircraft Systems (UAS) research and future analysis," in *Aerospace Conference, 2014 IEEE*, pp. 1–16, IEEE, 2014.

[30] Y. Tas, A. Yesildirek, and A. Sertbas, "Development of a networked control system platform for unmanned aerial vehicles," in *Computer Systems and Industrial Informatics (ICCSII), 2012 International Conference on*, pp. 1–6, IEEE, 2012.

[31] J. Fink, *Communication for teams of networked robots*. PhD thesis, Citeseer, 2011.

[32] V. Isler, B. Sadler, L. Preuchil, and S. Nishio, "Networked Robots [TC Spotlight]," *Robotics & Automation Magazine, IEEE*, vol. 22, no. 3, pp. 25–29, 2015.

[33] V. Kumar, D. Rus, and G. S. Sukhatme, "Networked Robots," in *Springer Handbook of Robotics*, pp. 943–958, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[34] R. J. Hall, "An Internet of Drones," *IEEE Internet Computing*, vol. 20, no. 3, pp. 68–73, 2016.

[35] M. Gharibi, R. Boutaba, and S. L. Waslander, "Internet of Drones," *CoRR*, vol. abs/1601.0, 2016.

[36] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles*. Springer Publishing Company, Incorporated, 2014.

[37] P. Bupe, R. Haddad, and F. Rios-Gutierrez, "Relief and emergency communication network based on an autonomous decentralized UAV clustering network," in *SoutheastCon 2015*, pp. 1–8, IEEE, IEEE, apr 2015.

[38] D. Behnke, K. Daniel, and C. Wietfeld, "Comparison of Distributed Ad-Hoc Network Planning Algorithms for Autonomous Flying Robots," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pp. 1–6, IEEE, dec 2011.

[39] J. Leonard, A. Savvaris, and A. Tsourdos, "Towards a fully autonomous swarm of unmanned aerial vehicles," in *Proceedings of 2012 UKACC International Conference on Control*, pp. 286–291, IEEE, sep 2012.

[40] J. Vanualailai, A. Sharan, and B. Sharma, “A swarm model for planar formations of multiple autonomous unmanned aerial vehicles,” in *2013 IEEE International Symposium on Intelligent Control (ISIC)*, pp. 206–211, IEEE, aug 2013.

[41] P. Dasgupta, “A Multiagent Swarming System for Distributed Automatic Target Recognition Using Unmanned Aerial Vehicles,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 38, 2008.

[42] K. Daniel, S. Rohde, N. Goddemeier, and C. Wietfeld, “Cognitive agent mobility for aerial sensor networks,” *IEEE Sensors Journal*, vol. 11, pp. 2671–2682, 2011.

[43] I. Bekmezci, O. K. Sahingoz, and S. Temel, “Flying Ad-Hoc Networks (FANETs): A survey,” *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013.

[44] E. Yanmaz, R. Kuschnig, and C. Bettstetter, “Channel measurements over 802.11 a-based UAV-to-ground links,” in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, pp. 1280–1284, IEEE, 2011.

[45] N. Goddemeier and C. Wietfeld, “Investigation of Air-to-Air Channel Characteristics and a UAV Specific Extension to the Rice Model,” in *2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–5, 2015.

[46] N. Goddemeier, K. Daniel, and C. Wietfeld, “Role-based connectivity management with realistic air-to-ground channels for cooperative UAVs,” *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 5, pp. 951–963, 2012.

[47] O. K. Sahingoz, “Networking Models in Flying Ad-Hoc Networks (FANETs): Concepts and Challenges,” *Journal of Intelligent {&} Robotic Systems*, vol. 74, no. 1, pp. 513–527, 2014.

[48] S. Rosati, K. Kruzelecki, G. Heitz, D. Floreano, and B. Rimoldi, “Dynamic Routing for Flying Ad Hoc Networks,” *Vehicular Technology, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2015.

[49] Y. Zheng, Y. Wang, Z. Li, L. Dong, Y. Jiang, and H. Zhang, “A Mobility and Load aware OLSR routing protocol for UAV mobile ad-hoc networks,” in *Information and Communications Technologies (ICT 2014), 2014 International Conference on*, pp. 1–7, IET, 2014.

[50] J. D. M. M. Biomo, T. Kunz, and M. St-Hilaire, "Routing in unmanned aerial ad hoc networks: Introducing a route reliability criterion," in *Wireless and Mobile Networking Conference (WMNC), 2014 7th IFIP*, pp. 1–7, 2014.

[51] O. Andryeyev, O. Artemenko, and A. Mitschele-Thiel, "Improving the system capacity using directional antennas with a fixed beam on small Unmanned Aerial Vehicles," in *Networks and Communications (EuCNC), 2015 European Conference on*, pp. 139–143, 2015.

[52] E. d. S. Moreira, R. M. P. Vanni, D. L. Função, and C. A. C. Marcondes, "A Context-Aware Communication Link for Unmanned Aerial Vehicles," in *Telecommunications (AICT), 2010 Sixth Advanced International Conference on*, pp. 497–502, 2010.

[53] S. Temel and I. Bekmezci, "Scalability analysis of Flying Ad Hoc Networks (FANETs): A directional antenna approach," in *Communications and Networking (BlackSeaCom), 2014 IEEE International Black Sea Conference on*, pp. 185–187, IEEE, 2014.

[54] C. h. An, J. Yang, D. K. Kim, U. Y. Pak, and K. Kwon, "Numerical analysis of transmission efficiency to guarantee the QoS for densely distributed UAV with randomly located ground station," in *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, vol. 2, pp. 426–429, feb 2010.

[55] Y. Gu, M. Zhou, S. Fu, and Y. Wan, "Airborne WiFi networks through directional antennae: An experimental study," in *Wireless Communications and Networking Conference (WCNC), 2015 IEEE*, pp. 1314–1319, IEEE, 2015.

[56] A. I. Alshbatat and L. Dong, "Cross layer design for mobile ad-hoc unmanned aerial vehicle communication networks," in *Networking, Sensing and Control (ICNSC), 2010 International Conference on*, pp. 331–336, IEEE, 2010.

[57] A. I. Alshbatat and L. Dong, "Adaptive MAC protocol for UAV communication networks using directional antennas," *2010 International Conference on Networking, Sensing and Control (ICNSC)*, pp. 598–603, 2010.

[58] S. Temel and I. Bekmezci, "LODMAC: Location Oriented Directional MAC protocol for FANETs," *Computer Networks*, 2015.

[59] S. Say, N. Aomi, T. Ando, and S. Shimamoto, "Circularly multi-directional antenna arrays with spatial reuse based MAC for aerial sensor networks," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, pp. 2225–2230, 2015.

[60] C. K. Lin, H. T. Kung, T. H. Lin, S. J. Tarsa, and D. Vlah, "Achieving high throughput ground-to-UAV transport via parallel links," in *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, 2011.

[61] W. H. Robinson and A. P. Lauf, "Aerial MANETs: Developing a resilient and efficient platform for search and rescue applications," *Journal of Communications*, vol. 8, no. 4, pp. 216–224, 2013.

[62] O. Bouachir, F. Garcia, N. Larrieu, and T. Gayraud, "Ad hoc network qos architecture for cooperative unmanned aerial vehicles (uavs)," in *Wireless Days (WD), 2013 IFIP*, pp. 1–4, IEEE, 2013.

[63] P.-B. Bök and Y. Tüchelmann, "Context-aware qos control for wireless mesh networks of uavs," in *Computer Communications and Networks (ICCN), 2011 Proceedings of 20th International Conference on*, pp. 1–6, IEEE, 2011.

[64] J. Stephan, J. Fink, B. Charrow, A. Ribeiro, and V. Kumar, "Robust routing and Multi-Confirmation Transmission Protocol for connectivity management of mobile robotic teams," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 3753–3760, IEEE, 2014.

[65] I. Stojmenović, "Position-based routing in ad hoc networks," *IEEE Communications Magazine*, vol. 40, no. 7, pp. 128–134, 2002.

[66] M. T. Hyland, B. E. Mullins, R. O. Baldwin, and M. A. Temple, "Simulation-Based Performance Evaluation of Mobile Ad Hoc Routing Protocols in a Swarm of Unmanned Aerial Vehicles," in *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, vol. 2, pp. 249–256, 2007.

[67] E. Kuiper and S. Nadjm-Tehrani, "Geographical Routing With Location Service in Intermittently Connected MANETs," *IEEE Transactions on Vehicular Technology*, vol. 60, pp. 592–604, feb 2011.

[68] I. Stojmenović, "Location Updates for Efficient Routing in Ad Hoc Networks," in *Handbook of Wireless Networks and Mobile Computing*, pp. 451–471, New York, NY, USA: John Wiley & Sons, Inc., 2002.

[69] Y. Li, R. Shirani, M. St-Hilaire, and T. Kunz, "Improving routing in networks of Unmanned Aerial Vehicles: Reactive-Greedy-Reactive," *Wireless Communications and Mobile Computing*, vol. 12, no. 18, pp. 1608–1619, 2012.

[70] R. Costa, D. Rosario, E. Cerqueira, and A. Santos, "Enhanced connectivity for robust multimedia transmission in UAV networks," in *Wireless Days (WD), 2014 IFIP*, pp. 1–6, IEEE, 2014.

[71] C. Lee, S. Kang, and K.-I. Kim, "Design of hierarchical routing protocol for heterogeneous airborne ad hoc networks," in *Information Networking (ICOIN), 2014 International Conference on*, pp. 154–159, IEEE, 2014.

[72] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Network*, vol. 16, no. 4, pp. 11–21, 2002.

[73] A. Maghsoudlou, M. St-Hilaire, and T. Kunz, "A survey on geographic routing protocols for mobile ad hoc networks," *Carleton University, Systems and Computer Engineering, Technical Report SCE-11-03*, 2011.

[74] J. D. M. M. Biomo, T. Kunz, and M. St-Hilaire, "Routing in unmanned aerial ad hoc networks: A recovery strategy for Greedy geographic forwarding failure," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2236–2241, apr 2014.

[75] L. Lin, Q. Sun, S. Wang, and F. Yang, "A geographic mobility prediction routing protocol for Ad Hoc UAV Network," in *2012 IEEE Globecom Workshops*, pp. 1597–1602, IEEE, IEEE, dec 2012.

[76] S. Kacianka and H. Hellwagner, "Adaptive video streaming for UAV networks," in *Proceedings of the 7th ACM International Workshop on Mobile Video*, pp. 25–30, ACM, 2015.

[77] D. Rosário, Z. Zhao, T. Braun, E. Cerqueira, A. Santos, and I. Alyafawi, "Opportunistic routing for multi-flow video dissemination over Flying Ad-Hoc Networks," in *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on*, pp. 1–6, IEEE, 2014.

[78] J. Hu, L. Xie, K.-Y. Lum, and J. Xu, “Multiagent information fusion and cooperative control in target search,” *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 4, pp. 1223–1235, 2013.

[79] W. Meng, L. Xie, and W. Xiao, “Communication aware UAV motion coordination for source localization and tracking,” in *Control Conference (CCC), 2013 32nd Chinese*, pp. 7451–7455, IEEE, 2013.

[80] J. Schleich, A. Panchapakesan, G. Danoy, and P. Bouvry, “UAV fleet area coverage with network connectivity constraint,” in *Proceedings of the 11th ACM international symposium on Mobility management and wireless access*, pp. 131–138, ACM, 2013.

[81] A. N. Kopeikin, S. S. Ponda, L. B. Johnson, O. Toupet, and J. P. How, “Real-time dynamic planning to maintain network connectivity in a team of unmanned air vehicles,” in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*, pp. 1303–1307, 2011.

[82] B. M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, “Graph-theoretic connectivity control of mobile robot networks,” *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525–1540, 2011.

[83] I. Bekmezci, M. Ermis, and S. Kaplan, “Connected multi UAV task planning for Flying Ad Hoc Networks,” in *Communications and Networking (BlackSeaCom), 2014 IEEE International Black Sea Conference on*, pp. 28–32, IEEE, 2014.

[84] R. Dutta, L. Sun, M. Kothari, R. Sharma, and D. Pack, “A cooperative formation control strategy maintaining connectivity of a multi-agent system,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1189–1194, 2014.

[85] P. R. Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, “A passivity-based decentralized strategy for generalized connectivity maintenance,” *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[86] V. K. Mehta and F. Arrichiello, “Connectivity maintenance by robotic Mobile Ad-hoc NETwork,” *CoRR*, vol. abs/1312.2, 2013.

[87] J. Senthilnath, S. N. Omkar, V. Mani, and A. R. Katti, “Cooperative communication of UAV to perform multi-task using nature inspired techniques,” in *2013 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pp. 45–50, apr 2013.

[88] E. Yanmaz, "Connectivity versus area coverage in unmanned aerial vehicle networks," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 719–723, IEEE, 2012.

[89] B. Di, R. Zhou, and H. Duan, "Potential field based receding horizon motion planning for centrality-aware multiple UAV cooperative surveillance," *Aerospace Science and Technology*, vol. 46, pp. 386–397, 2015.

[90] D. Behnke, P. B. Bok, and C. Wietfeld, "UAV-Based Connectivity Maintenance for Borderline Detection," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pp. 1–6, 2013.

[91] C. Dixon and E. W. Frew, "Optimizing cascaded chains of unmanned aircraft acting as communication relays," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 5, pp. 883–898, 2012.

[92] D. B. Rawat, R. Grodi, and C. Bajracharya, "Enhancing connectivity for communication and control in Unmanned Aerial Vehicle networks," in *Radio and Wireless Symposium (RWS), 2015 IEEE*, pp. 200–202, IEEE, 2015.

[93] E. W. Frew and T. X. Brown, "Networking Issues for Small Unmanned Aircraft Systems," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1, pp. 21–37, 2009.

[94] M. Balduccini, D. N. Nguyen, and W. C. Regli, "Coordinating UAVs in Dynamic Environments by Network-Aware Mission Planning," in *Military Communications Conference (MILCOM), 2014 IEEE*, pp. 983–988, IEEE, 2014.

[95] Y.-N. L. Y.-N. Lien, H.-C. J. H.-C. Jang, and T.-C. T. T.-C. Tsai, "A MANET Based Emergency Communication and Information System for Catastrophic Natural Disasters," *2009 29th IEEE International Conference on Distributed Computing Systems Workshops*, pp. 412–417, 2009.

[96] H. Kuntze, C. W. Frey, I. Tchouchenkov, B. Staehle, E. Rome, K. Pfeiffer, a. Wenzel, and J. Wollenstein, "SENEKA - sensor network with mobile robots for disaster management," *Homeland Security (HST), 2012 IEEE Conference on Technologies for*, pp. 406–410, 2012.

[97] Y. Takeuchi, "Radio Policy in Japan, Ministry of Internal Affairs and Communications, IEEE 802.15-11-0674-00-0000," 2011.

[98] F. C. C. (FCC), "Deployable Aerial Communications Architecture in Emergency Communications," tech. rep., 2011.

[99] "White Paper on Emergency Communication," Space and Advanced Communications Research Institute (SACRI) of the Georgia Washington University, 2006.

[100] H.-C. Jang, Y.-N. Lien, and T.-C. Tsai, "Rescue information system for earthquake disasters based on MANET emergency communication platform," *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing Connecting the World Wirelessly - IWCMC '09*, p. 623, 2009.

[101] J. Wang, Y. Wu, N. Yen, S. Guo, and Z. Cheng, "Big Data Analytics for Emergency Communication Networks: A Survey," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, p. 1, 2016.

[102] M. Shimamura, "Challenge to Predict Natural Disasters and Current Status of Research," *JR East Technical Review*, vol. 2, pp. 51–53, 2003.

[103] D. Krishnaswamy, R. Krishnan, A. Qamar, and K. Rajagopal, "Sankat-Edge: A distributed network edge infrastructure framework for disaster recovery," in *2014 IEEE Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS)*, pp. 216–221, IEEE, sep 2014.

[104] T. Do-Duy and M. A. Vazquez-Castro, "Efficient communication over cellular networks with network coding in emergency scenarios," in *2015 2nd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pp. 71–78, IEEE, nov 2015.

[105] "Corriere della sera, Online news, http://www.corriere.it, (05-29-2012)."

[106] Bbc.com, "Italy quake rescuers ask locals to unlock their wi-fi (http://www.bbc.com/news/technology-37186290)," aug 2016.

[107] J. Wang, Z. Cheng, P. Li, J. Chen, and Y. Zhou, "Design of a Best Load Balancing Method for Anti-disaster Mobile Mesh Communication Networks," in *Proceedings of the 2013 IEEE Second International Conference on Mobile Services*, MS '13, (Washington, DC, USA), pp. 63–69, IEEE Computer Society, 2013.

[108] R. Shaw, B. Peary, A. Ideta, and Y. Takeuchi, "Knowledge Note 3-2 Emergency Communication, Kyoto University and Japan's Ministry of Internal Affairs and Communication. Available: http://www.recoveryplatform.org/assets/publication/GFDRR/drm_kn3-2.pdf."

[109] Manesh Shrestha; Ray Sanchez; Wayne Drash; CNN, "Earthquake slams Nepal; devastating loss of people, history (http://edition.cnn.com/2015/04/25/asia/nepal-earthquake-7-5-magnitude/)."

[110] S. Bhattacharjee, S. Basu, S. Roy, and S. Das Bit, "Best-effort delivery of emergency messages in post-disaster scenario with content-based filtering and Priority-enhanced PRoPHET over DTN," in *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–7, IEEE, jan 2016.

[111] C. Raffelsberger and H. Hellwagner, "Evaluation of MANET routing protocols in a realistic emergency response scenario," *WISES 2012 - Proceedings, Workshop on Intelligent Solutions in Embedded Systems*, pp. 88–92, 2012.

[112] Z. Zhang, K. Long, and J. Wang, "Self-organization paradigms and optimization approaches for cognitive radio technologies: a survey," *IEEE Wireless Communications*, vol. 20, pp. 36–42, apr 2013.

[113] S. Ghafoor, P. Sutton, C. Sreenan, and K. Brown, "Cognitive radio for disaster response networks: survey, potential, and challenges," *IEEE Wireless Communications*, vol. 21, pp. 70–80, oct 2014.

[114] D. Srinivasarao, P. D. Pradeep, and B. A. Kumar, "A Survey of Emergency Communication Network Architectures," *International Journal of u- and e-Service, Science and Technology*, vol. 8, pp. 61–68, apr 2015.

[115] Y. Li, "A Survey on Communication Networks in Emergency Warning Systems," tech. rep., Washington University in St Louis, 2011.

[116] F. Legendre, "30 years of ad hoc networking research," in *Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief - ACWR '11*, (New York, New York, USA), p. 217, ACM Press, 2011.

[117] D. G. Reina, M. Askalani, S. L. Toral, F. Barrero, E. Asimakopoulou, and N. Bessis, "A Survey on Multihop Ad Hoc Networks for Disaster Response Scenarios," *International Journal of Distributed Sensor Networks*, vol. 2015, pp. 1–16, 2015.

[118] G. Mohandas, S. Silas, and S. Sam, "Survey on routing protocols on mobile adhoc networks," in *2013 International Mutli-Conference on Automation,*

Computing, Communication, Control and Compressed Sensing (iMac4s), pp. 514–517, IEEE, mar 2013.

[119] S. Kumar, "A Survey on Delay Tolerant Network in Disaster Management," *International Journal of Engineering Research & Technology*, vol. 3, no. 7, 2014.

[120] M. Liu, Y. Yang, and Z. Qin, "A survey of routing protocols and simulations in delay-tolerant networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6843 LNCS, pp. 243–253, 2011.

[121] H. Suzuki, Y. Kaneko, K. Mase, S. Yamazaki, and H. Makino, "An Ad Hoc Network in the Sky, SKYMESH, for Large-Scale Disaster Recovery," in *IEEE Vehicular Technology Conference*, pp. 1–5, IEEE, sep 2006.

[122] T. Sakano, Z. M. Fadlullah, N. D. Thuan, H. Nishiyama, M. Nakazawa, F. Adachi, N. Kato, A. Takahara, T. Kumagai, H. Kasahara, and S. Kurihara, "Disaster-resilient networking: a new vision based on movable and deployable resource units," *IEEE Network*, vol. 27, pp. 0–1, 2013.

[123] K. Gomez, S. Kandeepan, M. M. Vidal, V. Boussemart, R. Ramos, R. Hermenier, T. Rasheed, L. Goratti, L. Reynaud, D. Grace, Q. Zhao, Y. Han, S. Rehan, N. Morozs, T. Jiang, I. Bucaille, T. Wirth, R. Campo, and T. Javornik, "Aerial base stations with opportunistic links for next generation emergency communications," *IEEE Communications Magazine*, vol. 54, pp. 31–39, apr 2016.

[124] A. Qiantori, A. B. Sutiono, H. Hariyanto, H. Suwa, and T. Ohta, "An Emergency Medical Communications System by Low Altitude Platform at the Early Stages of a Natural Disaster in Indonesia," *Journal of Medical Systems*, vol. 36, pp. 41–52, feb 2012.

[125] S. Basagni, M. Conti, S. Giordano, I. Stojmenovic, and I. Stojmenovi, *Mobile Ad Hoc Networking*, vol. 10. Hoboken, NJ, USA: John Wiley & Sons, Inc., jun 2004.

[126] T. A. Ramrekha and C. Politis, "A Hybrid Adaptive Routing Protocol for Extreme Emergency Ad Hoc Communication," in *2010 Proceedings of 19th International Conference on Computer Communications and Networks*, pp. 1–6, IEEE, aug 2010.

[127] E. Onem, S. Eryigit, T. Tugcu, and A. Akurgal, "QoS-Enabled Spectrum-Aware Routing for Disaster Relief and Tactical Operations over Cognitive

Radio Ad Hoc Networks," in *MILCOM 2013 - 2013 IEEE Military Communications Conference*, pp. 1109–1115, IEEE, nov 2013.

[128] S. George, W. Zhou, H. Chenji, M. Won, Y. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah, "DistressNet: a wireless ad hoc and sensor network architecture for situation management in disaster response," *IEEE Communications Magazine*, vol. 48, pp. 128–136, mar 2010.

[129] H. Verma and N. Chauhan, "MANET based emergency communication system for natural disasters," in *International Conference on Computing, Communication & Automation*, pp. 480–485, 2015.

[130] K. Derr and M. Manic, "Extended Virtual Spring Mesh (EVSM): The Distributed Self-Organizing Mobile Ad Hoc Network for Area Exploration," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 5424–5437, dec 2011.

[131] Beibei Wang and K. J. R. Liu, "Advances in cognitive radio networks: A survey," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, pp. 5–23, feb 2011.

[132] Y. Tachwali, F. Basma, and H. Refai, "Cognitive radio architecture for rapidly deployable heterogeneous wireless networks," *IEEE Transactions on Consumer Electronics*, vol. 56, pp. 1426–1432, aug 2010.

[133] P. Conder, L. Linton, and M. Faulkner, "Cognitive radio developments for emergency communication systems," in *2011 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 658–659, IEEE, may 2011.

[134] P. Pawelczak, R. Prasad, and I. Niemegeers, "Cognitive radio emergency networks - requirements and design," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, pp. 601–606, 2005.

[135] R. Ferrus, O. Sallent, G. Baldini, and L. Goratti, "Public Safety Communications: Enhancement Through Cognitive Radio and Spectrum Sharing Principles," *IEEE Vehicular Technology Magazine*, vol. 7, pp. 54–61, jun 2012.

[136] N. Pratas, N. Marchetti, N. R. Prasad, A. Rodrigues, and R. Prasad, "Decentralized Cooperative Spectrum Sensing for Ad-Hoc Disaster Relief Network Clusters," in *2010 IEEE 71st Vehicular Technology Conference*, pp. 1–5, IEEE, 2010.

[137] S. GVK and J. Bapat, "Adaptive Model based on Proactive Spectrum Sensing for Emergency Cognitive Ad hoc Networks," in *Proceedings of the 7th International Conference on Cognitive Radio Oriented Wireless Networks*, pp. 89–94, IEEE, 2012.

[138] Chihkai Chen, R. E. Hudson, and Kung Yao, "Energy-based cooperative spectrum sensing for emergency communications over fading channels in Cognitive Radio public safety networks," in *2012 IEEE Global Communications Conference (GLOBECOM)*, pp. 1345–1350, IEEE, dec 2012.

[139] W.-L. Shen, C.-S. Chen, K. C.-J. Lin, and K. a. Hua, "Autonomous Mobile Mesh Networks," *IEEE Transactions on Mobile Computing*, vol. 13, pp. 364–376, feb 2014.

[140] B. Shucker and J. Bennett, "Target Tracking with Distributed Robotic Macrosensors," in *MILCOM 2005 - 2005 IEEE Military Communications Conference*, pp. 1–7, IEEE.

[141] A. Derbakova, N. Correll, and D. Rus, "Decentralized self-repair to maintain connectivity and coverage in networked multi-robot systems," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3863–3868, IEEE, may 2011.

[142] K. Gomez, T. Rasheed, L. Reynaud, and S. Kandeepan, "On the performance of aerial LTE base-stations for public safety and emergency recovery," in *2013 IEEE Globecom Workshops, GC Wkshps 2013*, pp. 1391–1396, 2013.

[143] K. Gomez, A. Hourani, L. Goratti, R. Riggio, S. Kandeepan, and I. Bucaille, "Capacity evaluation of Aerial LTE base-stations for public safety communications," in *2015 European Conference on Networks and Communications, EuCNC 2015*, pp. 133–138, 2015.

[144] R. Fedrizzi, L. Goratti, T. Rasheed, and S. Kandeepan, "A heuristic approach to mobility robustness in 4G LTE public safety networks," in *2016 IEEE Wireless Communications and Networking Conference*, pp. 1–6, IEEE, apr 2016.

[145] M. Di Felice, A. Trotta, L. Bedogni, L. Bononi, F. Panzieri, G. Ruggeri, V. Loscri, and P. Pace, "STEM-mesh: Self-organizing mobile cognitive radio network for disaster recovery operations," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 602–608, IEEE, jul 2013.

[146] “Socrates EU-FP7 Project.” http://www.fp7-socrates.org.

[147] “Semafour EU-FP7 Project.” http://www.fp7-semafour.eu.

[148] G. Aloi, L. Bedogni, M. D. Felice, V. Loscrì, A. Molinaro, E. Natalizio, P. Pace, G. Ruggeri, A. Trotta, and N. R. Zema, “STEM-Net: an evolutionary network architecture for smart and sustainable cities,” *Transactions on Emerging Telecommunications Technologies*, vol. 25, pp. 21–40, jan 2014.

[149] L. Filipponi, A. Vitaletti, G. Landi, V. Memeo, G. Laura, and P. Pucci, “Smart city: an event driven architecture for monitoring public spaces with heterogeneous sensors,” in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, pp. 281–286, IEEE, 2010.

[150] C. Clancy, J. Hecker, E. Stuntebeck, and T. O’Shea, “Applications of Machine Learning to Cognitive Radio Networks,” *IEEE Wireless Communications*, vol. 14, no. 4, pp. 47–52, 2007.

[151] K. Chowdhury, R. Doost-Mohammady, W. Meleis, M. Di Felice, and L. Bononi, “Cooperation and communication in Cognitive radio networks based on TV spectrum experiments,” *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 1–9, jun 2011.

[152] F. Dressler and O. B. Akan, “A survey on bio-inspired networking,” *Comput. Netw.*, vol. 54, pp. 881–900, apr 2010.

[153] D. Miorandi, L. Yamamoto, and F. De Pellegrini, “A survey of evolutionary and embryogenic approaches to autonomic networking,” *Computer Networks*, vol. 54, no. 6, pp. 944–959, 2010.

[154] E. Natalizio and V. Loscrí, “Controlled mobility in mobile sensor networks: advantages, issues and challenges,” *Telecommunication Systems*, pp. 1–8, 2011.

[155] IEEE 802.21, “IEEE standard for local and metropolitan area networks: media independent handover services,” 2008.

[156] N. Omheni, F. Zarai, M. S. Obaidat, K.-F. Hsiao, and L. Kamoun, “A novel media independent handover-based approach for vertical handover over heterogeneous wireless networks,” *International Journal of Communication Systems*, vol. 27, no. 5, pp. 811–824, 2014.

[157] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*. Upper Saddle River, NJ, USA: Prentice Hall Press, 5th ed., 2010.

[158] Q. Dong and W. Dargie, "A Survey on Mobility and Mobility-Aware MAC Protocols in Wireless Sensor Networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 88–100, 2013.

[159] A. V. Gondha and A. A. Bavarva, "A Review on Mobility and Mobility Aware MAC Protocols in Wireless Sensor Network," *International Journal of Computer Applications*, vol. 91, pp. 46–56, apr 2014.

[160] S. M. Das, Y. C. Hu, C. S. G. Lee, and Y. H. Lu, "Mobility-aware ad hoc routing protocols for networking mobile robot teams," *Journal of Communications and Networks*, vol. 9, no. 3, pp. 296–311, 2007.

[161] R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.

[162] C. Wietfeld and K. Daniel, "Cognitive Networking for UAV Swarms," in *Handbook of Unmanned Aerial Vehicles*, pp. 749–780, Springer, 2014.

[163] N. Pogkas, G. E. Karastergios, C. P. Antonopoulos, S. Koubias, and G. Papadopoulos, "Architecture Design and Implementation of an Ad-Hoc Network for Disaster Relief Operations," *IEEE Transactions on Industrial Informatics*, vol. 3, pp. 63–72, feb 2007.

[164] N. Uchida, K. Takahata, and Y. Shibata, "Evaluation of Cognitive Wireless Networks in Rural Area for Disaster Information Network," in *2011 International Conference on Computational Science and Its Applications*, pp. 135–142, IEEE, jun 2011.

[165] A. Srinivas, G. Zussman, and E. Modiano, "Construction and maintenance of wireless mobile backbone networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 239–252, 2009.

[166] D. S. Hochbaum and W. Maass, "Approximation schemes for covering and packing problems in image processing and VLSI," *Journal of the ACM*, vol. 32, pp. 130–136, jan 1985.

[167] A. Borchers and D.-Z. Du, "The k-Steiner ratio in graphs," in *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing - STOC '95*, (New York, New York, USA), pp. 641–649, ACM Press, 1995.

[168] Z. a. Melzak, "On the problem of Steiner," *Canadian Mathematical Bulletin*, vol. 4, no. 2, pp. 143–148, 1961.

[169] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità, "An improved LP-based approximation for steiner tree," in *Proceedings of the 42nd ACM symposium on Theory of computing - STOC '10*, (New York, New York, USA), p. 583, ACM Press, 2010.

[170] G. Robins and A. Zelikovsky, "Tighter Bounds for Graph Steiner Tree Approximation," *SIAM Journal on Discrete Mathematics*, vol. 19, pp. 122–134, jan 2005.

[171] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2nd ed., 2001.

[172] S. Nouyan and M. Dorigo, "Chain Based Path Formation in Swarms of Robots," pp. 120–131, 2006.

[173] M. Di Felice, A. Trotta, L. Bedogni, K. R. Chowdhury, and L. Bononi, "Self-organizing aerial mesh networks for emergency communication," in *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, pp. 1631–1636, IEEE, sep 2014.

[174] B. D. Song, J. Kim, J. Kim, H. Park, J. R. Morrison, and D. H. Shim, "Persistent UAV service: An improved scheduling formulation and prototypes of system components," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 915–925, IEEE, may 2013.

[175] D. Behnke, K. Daniel, and C. Wietfeld, "Comparison of Distributed Ad-Hoc Network Planning Algorithms for Autonomous Flying Robots," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pp. 1–6, IEEE, dec 2011.

[176] I. Dalmasso, I. Galletti, R. Giuliano, and F. Mazzenga, "WiMAX networks for emergency management based on UAVs," in *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*, pp. 1–6, IEEE, oct 2012.

[177] A. Al-Hourani and S. Kandeepan, "Cognitive Relay Nodes for airborne LTE emergency networks," in *2013, 7th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–9, IEEE, dec 2013.

[178] K. Curran, J. Crumlish, and G. Fisher, "OpenStreetMap," *International Journal of Interactive Communication Systems and Technologies*, vol. 2, no. 1, pp. 69–78, 2012.

[179] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A computationally inexpensive empirical model of IEEE 802.11p radio shadowing in urban environments," in *2011 8th International Conference on Wireless On-Demand Network Systems and Services, WONS 2011*, pp. 84–90, 2011.

[180] T. Baykas, M. Kasslin, M. Cummings, Hyunduk Kang, J. Kwak, R. Paine, A. Reznik, R. Saeed, and S. J. Shellhammer, "Developing a standard for TV white space coexistence: technical challenges and solution approaches," *IEEE Wireless Communications*, vol. 19, pp. 10–22, feb 2012.

[181] M. Noda, T. Yukimatsu, T. Kinoshita, and M. Shida, "Propagation characteristics of data communication system for protection and disaster relief operations using TV white space," in *Electromagnetic Compatibility, Tokyo (EMC'14/Tokyo), 2014 International Symposium on*, pp. 282–285, 2014.

[182] J. Wang, M. Ghosh, and K. Challapali, "Emerging cognitive radio applications: A survey," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 74–81, 2011.

[183] I. G. Askoxylakis, A. Makrogiannakis, A. Miaoudakis, S. Papadakis, N. E. Petroulakis, M. Surligas, A. Traganitis, and N. Vervelakis, "A Rapid Emergency Deployment mobile communication node," in *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD 2014*, pp. 290–294, 2014.

[184] A. Achtzehn, J. Riihijarvi, G. M. Vargas, M. Petrova, and P. Mahonen, "Improving coverage prediction for primary multi-transmitter networks operating in the TV whitespaces," in *SECON*, pp. 623–631, 2012.

[185] L. Bedogni, A. Trotta, and M. Di Felice, "On 3-dimensional spectrum sharing for TV white and Gray Space networks," in *2015 IEEE WoWMoM*, pp. 1–8, IEEE, jun 2015.

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

[187] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *Electronics Letters*, vol. 27, no. 23, p. 2145, 1991.

[188] M. Mamei and F. Zambonelli, "Field-based Approaches to Adaptive Motion Coordination in Pervasive Computing Scenarios," *in "Handbook of Algorithms for Mobile and Wireless Networking and Computing" CRC Handbook*, 2004.

[189] A. Ghasemi and E. S. Sousa, "Opportunistic Spectrum Access in Fading Channels Through Collaborative Sensing," *Journal of Communications*, vol. 2, pp. 71–82, mar 2007.

[190] I. F. Akyildiz, D. M. Gutierrez-Estevez, and E. C. Reyes, "The evolution to 4G cellular systems: LTE-Advanced," *Physical Communication*, vol. 3, pp. 217–244, dec 2010.

[191] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[192] N. Mckeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, and S. Louis, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, p. 69, 2008.

[193] O. Mokryn, D. Karmi, A. Elkayam, and T. Teller, "Help Me: Opportunistic smart rescue application and system," in *2012 the 11th Annual Mediterranean Ad Hoc Networking Workshop, Med-Hoc-Net 2012*, pp. 98–105, 2012.

[194] T. Catarci, M. de Leoni, A. Marrella, M. Mecella, B. Salvatore, G. Vetere, S. Dustdar, L. Juszczyk, A. Manzoor, and H.-L. Truong, "Pervasive Software Environments for Supporting Disaster Responses," *IEEE Internet Computing*, vol. 12, pp. 26–37, jan 2008.

[195] A. Iera, A. Molinaro, S. Y. Paratore, G. Ruggeri, and A. Zurzolo, "Making a mesh router/gateway from a smartphone: Is that a practical solution?," *Ad Hoc Networks*, vol. 9, pp. 1414–1429, nov 2011.

[196] W. Y. Lin, K.-P. Hsueh, and P.-S. Pa, "The Development of Emergency Communication APP Using Ad Hoc Network with IPv6," in *2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pp. 41–44, IEEE, sep 2015.

[197] P. Mitra and C. Poellabauer, "Emergency response in smartphone-based Mobile Ad-Hoc Networks," in *2012 IEEE International Conference on Communications (ICC)*, pp. 6091–6095, IEEE, jun 2012.

[198] X. Wu, M. Mazurowski, Z. Chen, and N. Meratnia, "Emergency message dissemination system for smartphones during natural disasters," in *2011 11th International Conference on ITS Telecommunications, ITST 2011*, pp. 258–263, 2011.

[199] G. Aloi, L. Bedogni, L. Bononi, O. Briante, M. Di Felice, V. Loscrì, P. Pace, F. Panzieri, G. Ruggeri, and A. Trotta, "STEM-NET: How to deploy a self-organizing network of mobile end-user devices for emergency communication," *Computer Communications*, vol. 60, pp. 12–27, apr 2015.

[200] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," in *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, WMCSA '99, (Washington, DC, USA), pp. 90—-, IEEE Computer Society, 1999.

[201] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY, USA: Oxford University Press, Inc., 1999.

[202] M. Di Felice, L. Bedogni, and L. Bononi, "Group communication on highways: An evaluation study of geocast protocols and applications," *Ad Hoc Networks*, vol. 11, pp. 818–832, may 2013.

[203] A. Ralston, E. D. Reilly, and D. Hemmendinger, *Encyclopedia of computer science*, vol. 536. Van Nostrand Reinhold New York, 1993.

[204] "Flair: Framework Libre Air." available from https://devel.hds.utc.fr/software/flair.

[205] P. Fuxjaeger and S. Ruehrup, "Validation of the NS-3 interference model for IEEE802.11 networks," in *Wireless and Mobile Networking Conference (WMNC), 2015 8th IFIP*, 2015.

[206] D. Yue, E. Tian, and Q.-L. Han, "A delay system method for designing event-triggered controllers of networked control systems," *Automatic Control, IEEE Transactions on*, vol. 58, no. 2, pp. 475–481, 2013.

[207] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systems—a survey," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 1, pp. 403–416, 2013.

[208] X. Wang and M. D. Lemmon, "Event-triggering in distributed networked control systems," *Automatic Control, IEEE Transactions on*, vol. 56, no. 3, pp. 586–601, 2011.

[209] "Phoenix RC." http://www.phoenix-sim.com.

[210] "Simdrone from H-Sim." http://www.h-sim.com.

[211] "Realfight Drone." http://www.realflight.com/drone.

[212] “Gazebo.” http://gazebosim.org.

[213] “Morse.” http://www.openrobots.org/morse/doc/stable/morse.html.

[214] “ROS.” http://www.ros.org.

[215] “ArduPilot Mega.” http://www.ardupilot.co.uk.

[216] “Pixhawk.” https://pixhawk.org.

[217] Q. N. Simulator, “Scalable network technologies,” *Inc.[Online]. Available: www. qualnet. com*, 2011.

[218] H. NAWAZ+, H. M. ALI, and G. NABI, “Simulation based analysis of Reactive Protocols Metrics in MANET Using OPNET,” *Cell*, vol. 92, pp. 333–7588408, 2014.

[219] X. Zeng, R. Bagrodia, and M. Gerla, “GloMoSim: a library for parallel simulation of large-scale wireless networks,” in *Parallel and Distributed Simulation, 1998. PADS 98. Proceedings. Twelfth Workshop on*, pp. 154–161, IEEE, 1998.

[220] “Network Simulator - NS (version 2).” available on-line from http://www.isi.edu/nsnam/ns/.

[221] F. Kargl and E. Schoch, “Simulation of MANETs: a qualitative comparison between JiST/SWANS and ns-2,” in *Proceedings of the 1st international workshop on System evaluation for mobile platforms*, pp. 41–46, ACM, 2007.

[222] A. Varga, “OMNeT++,” in *Modeling and Tools for Network Simulation*, pp. 35–59, Springer, 2010.

[223] E. B. Hamida, G. Chelius, and J.-M. Gorce, “Impact of the physical layer modeling on the accuracy and scalability of wireless network simulation,” *Simulation*, 2009.

[224] “WNS3 ’15: Proceedings of the 2015 Workshop on Ns-3,” (New York, NY, USA), ACM, 2015.

[225] V. Kumar, L. Lin, D. Krajzewicz, F. Hrizi, O. Martinez, J. Gozalvez, and R. Bauza, “iTETRIS: Adaptation of ITS Technologies for Large Scale Integrated Simulation,” in *2010 IEEE 71st Vehicular Technology Conference*, pp. 1–5, IEEE, 2010.

[226] C. Sommer and F. Dressler, “Progressing toward realistic mobility models in VANET simulations,” *IEEE Communications Magazine*, vol. 46, pp. 132–137, nov 2008.

[227] M. Kudelski, L. M. Gambardella, and G. A. Di Caro, “RoboNetSim: An integrated framework for multi-robot and network simulation,” *Robotics and Autonomous Systems*, vol. 61, pp. 483–496, may 2013.

[228] C. Pinciroli, V. Trianni, R. O’Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, “ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems,” *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012.

[229] France, “Heudiasyc Laboratory, UMR CNRS 7253.”

[230] “UDT: Breaking the Data Transfer Bottleneck.” available from http://udt.sourceforge.net/.

[231] N. Point, “Optitrack,” *Natural Point, Inc.,[Online]. Available: http://www. naturalpoint. com/optitrack/.[Accessed 22 2 2014]*, 2011.

[232] R. L. P. Castillo and A. Dzul, *Modelling and Control of Mini-Flying Machines*. Springer-Verlag in Advances in Industrial Control.

[233] “Irrlicht 3D engine.” available from http://irrlicht.sourceforge.net/.

[234] “Xenomai.” http://xenomai.org.

[235] M. Helsley, “LXC: Linux container tools,” *IBM devloperWorks Technical Library*, 2009.

[236] “TapBridge class reference.” available on-line from https://www.nsnam.org/docs/doxygen/classns3_1_1_tap_bridge.html.