PH.D. IN ELECTRONIC AND COMPUTER ENGINEERING
Dept. of Electrical and Electronic Engineering
University of Cagliari

Task Allocation in the Internet of Things

Ing. Giuseppe Colistra

Advisor: Prof. Luigi Atzori

Curriculum: ING-INF/03 Telecomunicazioni

XXVII Cycle
April 2015

A Marzia e alla mia famiglia
punti fermi della mia vita

Acknowledgements

My apologies to the readers who are not able to understand Italian, but the following acknowledgements are written in my mother tongue, in order for me to be able to express myself better.

Il primo ringraziamento rivolto a Gigi. In questi anni é stato il mio tutor, mi ha guidato ad affrontare le diverse sfide che si sono presentate e mi ha dato modo di capire come affrontarne sempre di nuove. Sicuramente oggi questo risultato l'ho raggiunto grazie ai suoi consigli, alla sua guida ed i suoi preziosi suggerimenti.

Un ringraziamento particolare a tutti i colleghi dell'MCLab, con cui ho condiviso momenti di crescita professionale ed umana. Grazie a loro ho capito quanto é importante lavorare in un ambiente che stimoli il confronto e la circolazione della conoscenza.

Questi anni mi hanno anche portato ad affrontare la sfida di GreenShare. Volevo ringraziare tutto il team che ogni giorno si impegna per raggiungere tanti successi e non si scoraggia se ogni tanto c'é qualche ostacolo, ma anzi é un'ulteriore spinta a fare meglio e rimanere affamati.

Gli anni della mia formazione sono stati davvero indimenticabili ed ogni tanto ci penso con un pizzico di nostalgia. Sicuramente questo sentimento é suscitato dalle persone che ho incontrato lungo questo percorso. Prima in Calabria e poi in Sardegna, ho incontrato tante persone fantastiche, che non cito uno per uno solo perché verrebbe fuori un elenco troppo lungo. Insieme abbiamo passato tanti bei momenti e quindi li ringrazio tutti di cuore.

Un altro pezzo della mia vita sono sicuramente i Dioscuri. Una grande famiglia che mi ha insegnato cosa vuol dire la collaborazione e il sostegno reciproco per raggiungere una meta. Un grazie a tutti quelli che hanno gravitato intorno alla maglia neroverde.

Vorrei ringraziare anche tutti gli amici storici di Lamezia. Sempre presenti e di supporto in ogni momento. Oggi siamo tutti un po' sparsi per l'Italia e per il mondo, ma tutti aspettiamo ogni anno le vacanze di Natale per stare nuovamente tutti insieme.

Un grazie va anche a tutto il "parentato" che é stato sempre un supporto importante e presente nel momento del bisogno. Grazie a tutti per l'affetto che mi avete sempre dimostrato, soprattutto nei momenti di bisogno.

Il ringraziamento piú grande lo dedico alle donne della mia vita: Roberta, Mamma e Marzia. Roberta, mia sorella, si merita un grazie solo per la pazienza con cui mi sopporta e mi ha sempre sopportato, oltre che per il supporto sempre presente per ogni mia scelta. Non potró mai ringraziare abbastanza mia mamma, perché mi ha sempre sostenuto e mi ha dato la possibilitá di raggiungere questo traguardo. Perché mi ha dato la possibilitá di fare ogni scelta che ritenessi giusta, senza farmi mancare mai il supporto che solo una mamma puó dare. Grazie a Marzia, perché é una presenza fissa nella mia vita. Compagna di tutti i giorni, unica e insostituibile. Grzie per condividere con me ogni gioia e soddisfazione, grazie per essermi vicina nei momenti di sconforto e ancora grazie per darmi forza nei momenti di difficoltá.

Infine, ma non ultimi, un grazie a chi non c'é piú. Questo traguardo é arrivato anche grazie a loro e anche se non possono condividere con me questo momento di gioia, sono sicuro che saranno felici. Grazie di cuore.

Contents

Introduction

The last few years have been involved by the technological revolution represented by the Internet of Things (IoT) [1]. The IoT vision aims to interconnect devices with different capabilities such as sensors, actuators, Radio Frequency Identification (RFID) tags, smart objects (e.g. smartphones), and servers, within the same heterogeneous network. The aim is to enable the network objects to dynamically cooperate and make their resources available, in order to reach a goal, i.e. the execution of one or more applications assigned to the network.

As known since its invention, the Internet interconnects nodes with dissimilar characteristics without central authorities by introducing some simple yet effective protocols that allow for nodes' interoperability so that information is successfully exchanged and services are provided by servers to clients and among peers. Fortunately, the same happens among objects in the IoT so that interoperability is assured and the data sensed by objects distributed and connected to the physical world is now available for the benefit of the human users.

The realization of the IoT paradigm relies on the implementation of systems of cooperative intelligent objects with key interoperability capabilities. However, to reach this goal, it's important to consider some key features that characterize many IoT objects: i) available nodes' resources (electrical energy, memory, processing, node capability to perform a given task) are often limited. This is the case, for example, of battery powered nodes, which have limited energy amounts. ii) sensors may provide information that is not unique but can be generated by set of different objects which for example are capable to sense the same physical measure of the same geographical. iii) the number of nodes in the IoT is quickly overcoming the number of hosts in the ́traditional' Internet and most of these have a low reliability due mostly to the mobility and energy. This entails for a new paradigm of communication according to which objects coordinate with the other objects in groups and provide a unified service to the external world (the application that requires the service), with the intent to distribute the load of the requested services according to specific community defined rules, which could be: lifetime extension, QoS (Quality of Service) maximization, reward maximization, or others.

It is evident that an appropriate coordination of objects' resources utilization would consistently improve their performance. If in the WSN field the problem of task allocation are highly

studied, as far as IoT networks are concerned, resource allocation is an open issue.

Network heterogeneity, which regards both node capabilities and characteristic parameters, makes the resource allocation a challenging task. Given the size of a distributed heterogeneous system such as the IoT network, the optimal creation of communities and the resource allocation within are not trivial issues. Furthermore, typical IoT networks are characterised by the dynamic behaviour of their nodes. In fact, emerging applications in smart environments such as smart cities and smart homes, where IoT is preponderant, are often based on opportunistic networks. In opportunistic networks, connections among nodes are created dynamically in an infrastructure-less way: when forwarding a message, next hops are chosen opportunistically, on the basis of their likelihood to get the message closer to its destination [2]. In such a dynamic context, with frequent and quick changes of scenario, it is not reasonable the community management to be centralized but a distributed approach has to be followed.

In the IoT, key nodes are represented by sensors, actuators, RFID tags, smart objects, and servers connected to the Internet [1]. Nodes have the most diverse characteristics and capabilities: different residual energy, power consumption, processing capacity, available memory, and capability of performing a limited amount of tasks. In the reference scenario, all these types of devices need to interoperate and reconfigure in an autonomous way, in order to perform some given applications. Most of the IoT powerful applications require the collaboration of different nodes, where each one performs a particular task and the mash-up of all the single tasks brings to the most disparate applications, e.g.: smart home monitoring, dangerous situation detection, tracking of goods, urban mobility assistant. Furthermore, it is envisaged that more complex applications will be developed in the forthcoming future, e.g. cars that talk to each other and are able to gather critical information such as accident locations to ambulances, or traffic congestion to other drivers [3].

Such a scenario not only entails heterogeneity among devices, but also heterogeneity among roles that the same device can assume. For example, a temperature sensing node could be used both to periodically send sensed data to a server for monitoring purposes, and to trigger the air conditioning system when the temperature exceeds a threshold.

Furthermore, IoT is strictly related to ubiquitous networking, which is characterised by a huge number of nodes deployed over an extensive area. The network is not only made of static or semi-static devices as it is in traditional networks, but topology changes quickly, so that it is impossible for nodes to be able to know the whole network topology. As a consequence, challenges arise with respect to autonomous reconfiguration and interoperation of nodes [4]. For instance, it may happen that there are no nodes available to perform a given task at the desired geographical location and at a given time. Or else, it may happen that the same node is

required to perform two conflicting tasks. Consider, for example, a smart city where, whenever an ambulance needs to pass the intersection, the corresponding traffic light turns green while the others turn red. If two ambulances are approaching the intersection from two different ways, the same traffic light is required to turn red and green at the same time.

All these features portrait a very complex and dynamic network, where all nodes need to interoperate in order to reason and allocate available resources in a distributed way with the aim of executing the applications assigned to the network. Most of these decisions should be taken autonomously to avoid centralized solutions, which usually limit the flexibility of the system and require intense control data exchanges.

This foreword is necessary to introduce this thesis, which is defined as follows.

Task allocation in the IoT: given the IoT paradigm and the requirements of IoT applications, the nodes involved in the execution of the same application should cooperate to reach the optimal allocation of tasks among them. They should execute tasks to reach the global application target and to satisfy the relevant requirements while optimizing the network performance in terms of resources used. This issue should be continuously addressed to dynamically adapt the system to changes in terms of application requirements and network topology.

The rest of the thesis is structured as follows.

- Chapter 1 deals with the state of the art of the IoT and gives an overview of some scenario and issues with particular emphasis on the task allocation problem.
- In chapter 2 functionalities of the middleware layer for IoT task allocation are discussed. The importance of virtualization is highlighted and the role of the VOs is discussed.The resulting middleware architecture is presented at the end of chapter.
- Chapter 3 presents a distributed optimization strategy based on the consensus algorithm. The challenges faced in this chapter is the deployment of distributed applications in the IoT in terms of cooperation among objects, with the aim of distributing the burden of the execution of the application committed to the network, so that resources are adequately exploited.
- In chapter 4 a new task allocation strategy is introduced. The proposed solution has the goal to deploy distributed applications in the IoT in terms of cooperation among objects, with the aim to solve the problem of resource allocation and management preserving a required QoS.
- Chapter 5 presents details about the experimentation and the performance analysis of algorithm presented in previous chapters, analysing various cases

within simulated and real scenarios.

- Chapter 6 concludes the paper resuming the work presented in this thesis, its contribution in the field of the task allocation in the IoT and introducing some ideas for future works.

Chapter 1

State of the art

The IoT paradigm has been evolving during the last decade towards the creation of a cyber-physical world where everything can be found, activated, probed, interconnected, and updated, so that any possible interaction, both virtual and/or physical, can take place. Its success resides in the massive spread of smarts objects (i.e. smartphones, tablets or smartwatches) which are able to connect to the internet. Nowadays, this high attention on the IoT topic stimulates industry and research to invest a lot of resources in this emerging field, consequently IoT is became a hot research topic, as demonstrated by the increasing attention and the large worldwide investments devoted to it.

In the followings the state of the art related to the problem of task allocation in IoT is summarized.

1.1 Resource allocation problem

One of the main challenges for distributed networks is to solve the problem of resource wasting. Huge progresses have been made to improve optimization inside a single node. Furthermore, a great deal of effort has been made by researchers to find effective strategies to increase the performance into the network also using cooperation among nodes. These strategies encompass task allocation (i.e. using consensus algorithms), smart deployment or data aggregation. In the following, the state of the art regarding these mechanisms will be presented.

1.1.1 Task allocation

Task assignment is performed taking into account various aspects related to energy consumption such as network topology, battery power, and node processing capabilities. However, existing methods have limited scope in studying using of resources extension with regards to application data processing. For instance, in [5], energy problem is studied with the main goal to increase

network lifetime. Researchers consider only communication tasks, but not the tasks generated by applications and assigned to the network for execution. Furthermore, it only focuses on homogeneous networks, which are not common in real scenarios. The same energy problem is studied in [6] considering execution of application tasks, and providing an adaptive task allocation algorithm that aims at reducing the overall energy consumption by balancing node energy levels overall the network. However, this mechanism introduces considerably increases packet overhead.

One method to perform task assignment is the use of a central controller that divides large application programs into smaller and easily executable tasks and then distributes these tasks to nodes. Task allocation solutions that consider a central controller are called centralized solutions. A centralized solution for the maximization of the network lifetime will be described in [7]. In this work, the application assigned to the network can be decomposed into a sequence of distributed tasks. This application could represent diverse operation, such as: computing the average of the temperature in a given geographical area, measuring the light intensity in a room, video surveillance of a specific geographical area, or a combination of these. The same application can be performed in several different ways: gathered data can be immediately sent to a sink or it can be processed before being transmitted. In the case of the latter, the number of bits to be sent would be smaller, and therefore the transmission energy consumption would be lower as well; however, processing energy consumption could be higher in this second case. Quantifying the energy consumption in both cases, it could be possible to establish which one determines a reduction of battery consumption in the sensors. The researchers use this idea to design an algorithm able to increase the network lifetime. Some other centralized lifetime maximization algorithms are studied in [8][9][10]. The problem with centralized algorithms is that they suffer from computational complexity, as well as large control packet overhead due to frequent updates collected from nodes in order to adapt to network dynamism.

To address the problem of dynamism of distributed network, in [11]is designed a framework that determines the distribution of tasks among the nodes in a WSN by means of a distributed optimization algorithm, based on a gossip communication scheme, aimed at maximizing the network lifetime. The proposed algorithm is based on an iterative and asynchronous local optimization of the task allocations among neighbouring nodes. The resulting scheme is based on gossip, which consists in a communication paradigm in which, at each instant of time, each node in the network has some positive probability to interact with one of its neighbours. A similar approach is studied in [12], where the distributed algorithm is based on particle swarm optimization. However, the major drawback of these studies is that they do not take into account the deadline of the applications assigned to the network.

As far as the resource allocation in IoT, the problem is an open issue. Network heterogeneity, which regards both node capabilities and characteristic parameters, makes the resource allocation a challenging task. Semantic descriptions are needed, so that a common middleware can be designed in order to ensure interoperability among different devices. Comprehensive ontologies that provide a semantic model for IoT are defined in [13] and [14].

Most of the existing studies on resource allocation for IoT are focused on IoT service provisioning, such as in [15] and [16]. None of the works found in the literature tries to find the optimal resource allocation associated to the lowest impact of the application assigned to the network.

Consensus protocol

A consensus protocol can be used to take a distributed decision among nodes in the networks. A consensus algorithm is a collection of laws that regulates the interaction and the exchange of information between each node and its neighbours (nodes that are only one hop far from each other). All nodes in the network use the same algorithm in order to take decisions according to information available locally and received from other nodes. Olfati-Saber et al. [17] recall very well the history of consensus protocol from 1960 to recent years. The authors also describe as many seemingly different problems, which involve interconnection of dynamic systems in various areas of science and engineering, happen to be closely related to consensus problems. The applications where the consensus protocol has been used are various:

- *Synchronization of Coupled Oscillators* is important for several applications of engineering such as mobile target tracking, event detection, efficient scheduling, etc. [18][19][20].
- *Flocking Theory* is extensively studied by engineers because the coordination problem affects many applications such as massive distributed sensing using mobile sensor networks in an environment, self-assembly of connected mobile networks, automated delivery of payloads [21].
- *Rendezvous in Space* is equivalent to reaching a consensus in position by a number of agents. This problem is studied in the robotic field because using coordinated devices enables to perform a variety of challenging tasks, including search and recovery operations, surveillance, exploration and environmental monitoring [22].
- *Distributed Sensor Fusion in Sensor Networks* is the combination of sensory data from disparate sources such that resulting information is somewhat better than it would be when these sources are used individually. Consensus is used to coordinate nodes in the network [23].

1.1.2 Smart deployment

An appropriate node deployment is probably the most critical issue to be addressed to reduce wasting of resources in a network. In [24], the spaces between node and sink are adjusted in such a way that nodes with higher traffic have a shorter hop distance than nodes with lower traffic. An algorithm that has the goal to improve coverage and lifetime is presented in [25]. The authors model the coverage and lifetime of a node as a Gaussian random variable, whose parameters depend on some network settings. The nodes are then deployed according to the selected policies, in the network could be applied an algorithm of deploying oriented to improve lifetime or coverage. Instead in [26] is presented an algorithm to maximize the area of coverage, minimize the network energy consumption, maximize the network lifetime and minimize the number of deployed nodes. The problem is modelled as a multi-objective optimization problem where the aim is to find the optimal trade-off among the various indicators.

1.1.3 Data aggregation

The network resources optimization not only is centered on reduction of packet transmission power, but also involves convenient data processing that reduces the amount of data delivered to data sinks. This is the principle behind node clustering protocols, such as LEACH [27], EC [28] and the clustering algorithms in [29], in which cluster head nodes aggregate data and reduce transmitted data volume, which in turn reduces the overall transmission energy consumption of the network.

1.2 IoT paradigm and key issues

The basic idea of IoT concept is the pervasive presence of connected objects and the main idea is that any thing, conveniently tagged, may be able to communicate with other objects equally tagged through internet or any other protocols, to collaborate and to reach a common goal. Before the definition of IoT paradigm, the research has been primarily focused on devices belonging to delimited networks, providing optimized solutions for the resource-constrained devices of which they are characterized this type of networks. However, the concept has grown into multiple dimensions, encompassing sensor networks able to provide real-world intelligence and goal-oriented collaboration of distributed smart objects via local networks or global inter-connections such as the Internet [30]. As well as defined in [31] the current situation is the presence of many ”Intranets” of Things which are evolving into a much more integrated and heterogeneous system for build a Internet of Things ecosystem. A number of different research streams, that connotes the various intranet domain, converge in a melting-pot that characterizes

the IoT theme.

The skein, that characterizes the IoT vision, has been unravelled by Atzori et al. in [1]. The paper identifies that the IoT can be realized in three different paradigms: i) internet oriented, based on middleware definition; ii) things oriented, focused on sensors and actuators; iii) semantic oriented, centred on the context by meta-data and using ontologies and semantic. This type of delineation highlights the interdisciplinary nature of the topic, but the real usefulness of IoT can be unleashed only if the three paradigms are intersected.

Given this foreword the complexity of the IoT paradigm appears clear, so the list of challenges to be addressed is potentially very long and varied. In the subsections below are presented the main challenges that the researchers are taking on during their studies.

1.2.1 Heterogeneity

The realization of the IoT paradigm relies on the implementation of systems of cooperative intelligent objects with key interoperability capabilities. Once known the concepts of IoT, it's clear that one of the main enabling factor of IoT is the integration of several technologies and communications solutions. IoT can be considered as a global network infrastructure that could add a new evolution step to the world of ICT just as Internet once did.

Many efforts are spent to develop protocols for ubiquitous [32] and pervasive [33] networks such as ZigBee, Bluetooth, Wi-Fi, NFC. The studies have allowed to reach a significant maturity and market size for this solution, but each one has its own specific characteristics and application domains. However, all these vertical solutions give a fragmentation that may again hamper objects interoperability and can slow down the development of a unified reference model for the IoT. Fig. 1.1 shows the concept for which vertical solutions fragment the operational domain, whereas IoT horizontal model gives interoperability in a unique domain.

The key issue is that for connecting and integrating all the objects into the IoT, there are and will be many different technologies and protocol which introduce fragmentation in a scenario that should be rich of interoperability. IoT interoperability involves not only the ability of things to exchange information but also includes the capability for interaction and joint execution of common tasks [34]. To solve this issue is need a layer able to encapsulate functionality and available resources of each node. These should be exposed on the network like services accessible by a common interface. The requirement is a layer able to abstract the main features of each things from the underlying hardware and protocols [35].

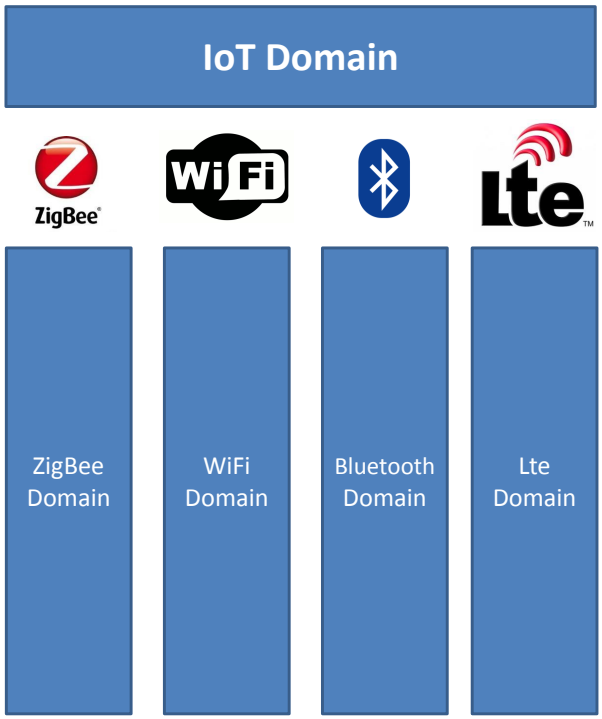


Figure 1.1: Vertical solutions fragment the domains, IoT horizontal model integrates in a unique domain.

1.2.2 Scalability

The number of object connect on IoT system has several orders of magnitude than the participants on conventional Internet of computers [36]. IoT paradigm implies that every tagged object could be part of a connected system. The quick widespread of smart objects, that are equipped of connection capabilities (i.e. smartphones, tablets, sensors, actuators) is increasing, it was presumed that each object will be part of a very-large-scale (VLS) pervasively connected system across the globe. [37]. It is well known that in the presence of many nodes, the effort required to coordinate the network dramatically increases and if the coordination has some bug, the performance of system will suffer or absolutely not work. The main roads to approach this challenge are two: i) reducing the number of messages and the amount of data transmitted throughout all the layers of the system [38] ii) choosing a small cluster of nodes able to execute a task [39]. Both paths have the different goal, but the same requirements to reach its: the presence of framework that enable a smart management in a VLS that characterize the IoT.

1.2.3 Identification

Identifying a device is one of the primary pillars of each type of network, so also in IoT the identification of an object is a main issue. In order to be able to address the billions of objects

in the IoT, the system first need to be able to identify them with a unique ID. Currently, in the internet world the IPv4 protocol identifies each node through a 4-byte address [40]. However, it is well known that the number of available IPv4 addresses is decreasing rapidly and will soon be insufficient [41]. A much better choice is using IPv6 protocol which, with its 128-bit addresses, permit to address a large number of devices [42]. Moreover, IPv6 allows network auto-configuration, stateless operation and IPv6 addressing has been proposed for low-power wireless communication nodes within the 6LoWPAN context [43].

Recently, other solutions have been developed to run in special environments, typically of an IoT, that have requirements like bandwidth-efficient, energy-efficient and capabilities of working with limited hardware resources. As a result, protocols such as Message Queue Telemetry Transport (MQTT) [44] and Constrained Application Protocol (CoAP) [45] are proposed. These protocols usage is optimized when using an all-IP infrastructure, but in IoT scenario this requirement is not available a priori, because nodes involved in the IoT are several heterogeneous and they could use various protocols.

A workaround could be using a high level middleware that allows to maintain some ”islands” that may use arbitrary multi-hop, ad-hoc routing algorithms to deliver object’s data to one or more sink nodes. An example is the Global Sensor Networks (GSN) that provides a middleware layer that abstracts details of access to the sensor data [46]. Solutions like this provide an easy identification of service and available resources of each node. The evolution of network identification techniques is summarized in Fig. 1.2.

1.2.4 Search and discovery

In IoT scenario, like in all distributed networks, a group of nodes want to cooperate to perform a task. Before they can do so, these nodes need to learn of the existence of each other thing which can cooperate [47]. So, the search and discovery services are fundamental blocks of any distributed computing system [48]. It’s trivial that to build a smart application in IoT world the presence of a component which allows to dynamically discover distributed smart objects and, specifically, the services and operations they provide.

Usually, in IoT context even the exact location and form of stored data are initially unknown to the requester, so an intermediate block is needed to find these data or object. In contrast to the user-oriented discovery typical of World Wide Web, where the user looks for a data and consults the result of search, in the IoT scenario the discovery block enables devices to automatically register themselves and advertise their services on the network. This discovery mechanism is essential in scenarios where devices can join the network and discover dynamically the services offered by other devices [49]. The main course to approach this challenge

Figure 1.2: Evolution of network identification techniques.

is to introduce a middleware layer that is able to enrich the information about the object with meta-data [50]. This layer can integrate various tool (i.e. semantic engine) that enable the discovery service and make easy the search of data in IoT. An interesting overview, comparison and analysis of some discovery services in IoT is performed in [51].

1.2.5 Mobility

In the IoT space not all things are always stationary, for example a person owns various devices and during its daily activity it can carry around them. The same user may make data queries or request to activate other things, which will be probably also them in movement. It's clear that the mobility is not a negligible situation in the IoT context, because it's fundamental manage it to enable the access to the things independently from where they are placed. As well surveyed in [52], to maintain connectivity of devices, there are two main component on the mobility issue: i) intra-domain refers to moving between different cells of the same system and ii) inter-domain refers to moving between different backbones, protocols, technologies, or service providers. If the intra-domain mobility is supported by several protocol [53], the inter-domain mobility, specially among heterogeneous systems, is a open issue. To provide seamless connectivity in a system rich of heterogeneity like IoT, methods of reducing handover delay are essential. The characteristics of devices involved in IoT (i.e., tiny, battery powered, and wireless) assume that

the mobility protocol should be supported in a lightweight fashion [54].

The object mobility causes an other problem, as a result of an object movement from one place to another, there is a change of context that should be stored to maintain consistency on the data offered by node. This task may result very difficult, particularly when the object becomes unavailable for long periods of time due to lack of connectivity. So the mobility is not only a problem of connectivity, but it's also a issue regarding the data consistency.

1.2.6 Plug and play

When a new device partakes in the network, it spends some effort to register itself, advertise its available services and resources. The described process should be "plug and play", so without requiring human intervention to configure some parameters the thing can immediately interact with other objects. Even the most trivial manual configuration task can become impossible when there are thousands of devices needing attention like in the IoT scenario. So the auto-configuration is a pillar to build the IoT. As described in [55] the "plug and play" feasibility (to not become "plug and pray") depends on more design factors like: information availability in a standard, or at least recognizable and unique format; standard APIs and protocol availability; a selection of good default values whenever possible. Aligned with this vision it has been developed the Universal Plug and Play technology [56] that supports zero-configuration networking and automatic discovery of devices.

In IoT system the challenge is to enable mechanism to make the objects plug-and-playable, despite the high level of heterogeneity and the difficult to acquire, analyse and interpret information about the context.

1.2.7 Security and privacy

Security and privacy are critical issue for the IoT. Measures ensuring the architecture's resilience to attacks, data authentication, access control and client privacy are requested by all involved stakeholders [57]. The paper [58] propose a good classification of security and privacy requirements: i) resilience to attacks, ii) data authentication, iii) access control and iv) client privacy. All of these requirements should be included in the risk management of all IoT system, to avoid that the vulnerabilities of a single node can be exploited and malicious users can access the IoT to launch service attacks or to damage the privacy of some users. Although each node might be perfectly safe by itself, when it interacts in an heterogeneous system with other nodes, the security is not taken for granted because also aspect of interaction should be considered. The design of specific security IoT mechanisms is still in its infancy, but the real challenge is to

approach this issue with a holistic view, because an ”impermeable view” is not relevant in the IoT context.

1.2.8 QoS management

Generally, the QoS requirement, in traditional wired networks, is supported via the over-provisioning of resources and/or traffic engineering. For instance, RFC 2386 [59] defines QoS as a set of service requirements in end-to-end communication obtained by packet transmission. In this scenario, QoS requirements are measurable by some attributes as: delay, jitter, available bandwidth and packet loss. Unfortunately, the QoS mechanisms used to support QoS in wired data networks cannot be directly applied to a very heterogeneous network with a dynamic topology [60]. It is well known that in WSN traffic characteristics and the measurable attributes strongly depend on the application scenario [61]. As much as the focus is oriented only on the traffic generated inside a WSN there aren't problem, but in IoT paradigm exist a bundle of technology and a sensor node becomes part of the overall Internet. In fact, in the IoT scenario exist several heterogeneous nodes which are deployed for the more various scope, so it's predictably that each node generates traffic with different characteristics. The datagram which traverse the IoT is highly diversified and this is a open issue in the QoS management field, as surveyed in [62] to enable end-to-end QoS in highly heterogeneous networking environment, the resources allocation mechanisms require high operational costs. A large research effort is still needed to find innovative solution in this field.

1.2.9 Constrained resources

The IoT is characterized by a lot of tiny devices connected. To maintain the device small, it needs to possess only computational capabilities for the task it has to perform and networking abilities allowing connectivity on the Internet. So, this embedded computing devices deployed within the IoT are expected to be resource constrained [63]. Available resources on node such as electrical energy, memory, processing, and node capability to perform a given task, are often limited. This is the case, for example, of wireless sensor nodes, which are often battery powered, and therefore have limited energy amounts. Another example is represented by the scarce processing capabilities of RFID tags. In this scenario is clear that resources management is need to avoid wastes which are not justifiable and they are dangerous for the network operation. A cooperation among node and use of optimization algorithms for the task allocation, as well as proposed in [34], [64] and [65], are an hot trend to solve the problem of resources scarcity. An other trend on the rise is to demand some functionality of nodes on remote system with higher capabilities and resources (i.e. cloud [66], [67] or fog [68], [69]).

1.3 Opportunistic network

IoT nodes can use two distinct approaches for enabling wireless mobile interface to communicate with each other: *Infrastructure based* or *Infrastructure less*.

In the first case the wireless mobile networks have traditionally been based on the cellular concept, centralized management and a good infrastructure support, in which mobile devices communicate with access points like base stations connected to the fixed network infrastructure. Typical examples of this kind of wireless networks are GSM, UMTS, 3G, LTE, etc.

The second case is the approach named infrastructure less and in this context the nodes use the short-range radio protocol (Bluetooth, WiFi, Zigbee, etc.) and they have totally decentralized management. A mobile wireless network with this characteristic is also commonly known as a Mobile Ad hoc NETwork (MANET). The MANET is a collection of wireless nodes that can dynamically form a network to exchange information without using any pre-existing fixed network infrastructure. This is a very important part of communication technology that supports truly pervasive computing, because in many contexts information exchange between mobile units cannot rely on any fixed network infrastructure, but on rapid configuration of a wireless connections on-the-fly [70]. A natural evolution of the MANETs is the *opportunistic network*. Opportunistic networking represents one of the emerging communication paradigms for pervasive and ubiquitous environment, typical of IoT scenario, by supporting wireless communications in intermittently connected network [71]. The use of short-range technology makes the opportunistic network human-centric because they inherently follow the way that people opportunistically get into contact.

As far as the research project, the dynamic task allocation issue in opportunist networks has not been addressed yet. The closest scenario is that of the works that consider the allocation of tasks in mobile ad hoc networks (MANETs). This is the case of [72] and [73], where computational tasks are assigned by cluster heads to cluster nodes, in such a way that the energy load is balanced among network nodes. However, the nodes are assumed to be almost statically part of the network.

The traditional view of IoT attempts to connect all the physical objects to build a global, infrastructure based IoT. Guo et al in [74] propose a novel approach with opportunistic connection in IoT. In this context the information are addressed, disseminated and shared within and among opportunistic communities of devices that are formed based on the movement and opportunistic contact nature of human. Each person has various personal devices such as mobile phones, wearable devices, smart object, that can form opportunistic IoT when they are equipped with the short-range communication and sensing modules. For instance, a person has various

smart objects with short-range radio interface, can connect with other objects that opportunistically meet on the train during a travel to build an ad-hoc mobile network. Information can be shared among the members of this opportunistic community. When a new person which has smart objects, gets on the train the community increases with other new nodes, instead if the people gets off the train, some nodes leave the community.

The strategy used in the opportunistic IoT allows to extend the capabilities of the network. This approach gives the opportunity to participate also at nodes that have limited communication capabilities. In the same time these changing of nodes make the network very time variable. In this described scenario the network is more variable and it involves very heterogeneous nodes such is typical in IoT framework.

1.4 Opportunistic sensing

In the IoT scenario there are a lot of devices that provide tremendous potential for information collection and processing in a variety of application domains. The first generation of ubiquitous network scenarios included stationary devices sensing ephemeral features of the environment around them.

In the last time researcher are exploring a new kind of sensing employing the smart devices that people use and carry with them. *Opportunistic sensing* has been introduced as a term to describe a new paradigm: small computational devices, carried by individuals in their daily activities, sensing information directly or indirectly related to human activity, as well as aspects of the environment around them. These smart objects share the information that they acquire and they collaborate to reach the common goal expected by the application. In an urban setting, one could leverage millions of personal smart objects, and a near-pervasive wireless-network infrastructure, to collect sensor data on a grand scale without the need to deploy thousands of static sensors. Thus, the opportunistic sensing model is a good solution for not using a lot of static sensor with infrastructure to support the communication. Nodes adopt opportunistic practices for sensing and networking, allowing their sensors to be remotely tasked on someone else's behalf, collecting and reporting sensor data on a best-effort basis when the conditions permit. [75].

An example of the concept with a real application in this field can be illustrate. Opportunistic sensing task will be deployed on the mobile devices to form interactive, cooperative sensor networks that enable public and professional users to gather, analyse and share local knowledge. In opportunistic sensing devices automatically determine when to use their sensor to meet the application's sensing requests. Instead of requiring human intervention to actively and con-

sciously participate in the sensing, opportunistic sensing requests that a sensing device be used automatically whenever its state (location, user activity, and so on) matches an application's requirements. [76] This type of application requires more resources for decision making, such as a determination frequency sensing or number of sample that the device have to storage in its buffer. As such, an opportunistic system must adapt to the number of devices that participate at the task.

In this described context it is clear that it's fundamental a totally distributed algorithm that manages the nodes community. So the devices involved in the opportunistic sensing application work harmoniously and don't waste important resources.

About the research projects that consider the presence of nodes performing opportunistic sensing, some experience has been studied. This is the case for instance of: BikeNet [77], which considers a number of sensors embedded into a cyclist's bicycle to gather quantitative data about the cyclist's rides and environment; Bubble Sensing [78], which is a sensor network abstraction that allows mobile phone users to create a binding between sensing tasks and the physical world at locations of interest, which remains active for a duration set by the user; DEAMON [79], which proposes an approach that considers a node which needs to perform a sensing task if a sensing condition is true.

1.5 IoT architectures

The IoT domain is characterized by a significant fragmentation and by the presence of heterogeneous systems based on dissimilar architectures. This makes a synergistic integration process difficult to be carried out. The need for a clear reference architectural model that will allow the different systems to cooperate is, thus, strongly felt. Several papers presented their own solution to provide conceptualizations for the IoT domain proposing at the first semantic middleware solutions and more recently virtualization layers that use the notion of virtual objects.

In the subsections below are presented the main projects which proposing solutions for the IoT architecture.

1.5.1 Semantic middleware

Since device interoperability is a crucial challenge for the IoT community, many different middlewares have been proposed in recent years. One of the most well-known is LINKSMART, formerly known as HYDRA [80]. LINKSMART is a middleware based on Service Oriented Application (SOA), which provides a transparent communication layer for embedded devices. Through the use of ontologies, LINKSMART ensures interoperable access to data, information

and knowledge across heterogeneous platforms, and support ambient intelligence for ubiquitous networked devices. Another example of SOA-based middleware is SOCRADES [81], a middleware focused on coupling web service enabled devices with business applications such as Enterprise Resource Planning (ERP) systems.

Examples of middlewares for sensor networks interoperability are SATware [82] and GSN [83]. In both works, devices are abstracted as combinations of virtual sensors, each of which is able to perform a single service. GSN is a framework for the distributed deployment of WSNs and data processing. SATware provides a middleware for sentient systems, where multimodal sensor data streams are queried, analysed and transformed.

The ASPIRE project [84] addresses the design of a middleware for the interoperability of RFID nodes. It aims to reduce cost barriers due to RFID network deployment, particularly for small and medium-sized enterprises, by developing a programmable open source middleware.

In the IoT, embedded systems are prevalent. Since resources are limited, dynamic ad-hoc networks are preferred to client-server networks. For this reason, embedded peer-to-peer (EP2P) systems are emerging rapidly. In the SMEPP project [85] a middleware providing abstraction to ease application and service development is realised. Furthermore, the SMEPP middleware includes mechanisms for secure interaction between peers and abstract developers.

One of the main applications of the IoT is represented by the Smart City paradigm. UbiRoad [86] is a middleware intended for Intelligent Transportation Systems (ITS), which uses semantic languages and semantic technologies for declarative specification of behaviour of devices and services, in order to facilitate and govern interoperation of devices, services and humans.

1.5.2 Virtual Object solution

Recently, several papers presented their own solution to provide conceptualizations for the IoT domain using the notion of virtual objects, such as [87], [88] and [89]. However, they consider the virtual objects simply as the digital counterpart of physical objects and focus more on the middleware framework rather than the modeling of related information. As a result, the absence of a common format for virtual objects causes problems of interaction and communication, since there is no standardized ways to obtain the actions or services associated with a virtual object.

In the last years, several research projects were founded to propose an architecture for the IoT, leading to an evolution of the definition of virtual object, and of its functionalities. Virtual objects are not anymore only digital interfaces to the real world but now provide a semantic enrichment of the data acquired, which makes easier the discovery of services.

The CONVERGENCE project [90], for example, makes use of Versatile Digital Item (VDI), a common container for all kinds of digital content, derived from the MPEG21 standard, containing one or more resources and metadata. This definition is similar to the one provided in [91] with a many-to-one association between real objects and VDI. However, the CONVERGENCE project provides a first attempt to implement the discovery of a particular VDI in the virtualization layer.

Another example is SENSEI [92], which enables the integration of heterogenous and distributed Sensor and Actuator Network (SAN) islands into a homogenous framework for real world information and interactions, by providing an abstraction level of resources corresponding to the real world consisting of Real World Entities or Entities of Interest (EoI). Resources may be associated with one or more EoIs for which they can either provide information or provide control over their surrounding environment, thus providing the same type of association of CONVERGENCE. In SENSEI, resources acquire the ability to enhance the data received by the sensors with environmental information.

Interesting definitions come from the IoT-A [3], the COMPOSE [93] and the iCore [94] projects. In IoT-A, physical entities are represented in the digital world via virtual entities, which have two fundamental properties. Firstly, while ideally there is only one physical entity for each virtual entity, it is possibile that the same physical entity can be associated to several virtual entities. Secondly, virtual entities are a synchronized representation of a given set of aspects of the physical entity. The association between virtual and physical entity is achieved by connecting one or more ICT devices to the physical entity so as to provide the technological interface for interacting with the virtual world. The physical object is decomposed in its functionalities thus providing a one-to-many correspondence with the virtual entities.

COMPOSE focuses on objects service composition and for this reason they need to abstract the heterogeneity of physical objects in terms of computing power, protocols and communication mechanisms, by introducing the concept of Service Object. The Service Object then represents a standard internal digital representation that makes easier the creation of COMPOSE services and applications.

In iCore, a virtual object is a virtual representation of an ICT object that may be associated to one (or more) real-world objects. The term real-world object refers to any object that exists in the real/physical world and then can be classified both as ICT objects, e.g. an email or a smartphone, and a non-ICT object, e.g. a person or a fruit; an important trait of the iCore project is that also a real-world object can be associated to one or more virtual objects. The virtualization layer, where all the virtual objects are located, acts as a management level that manages and provides interfaces for accessing virtual object to other iCore components.

1.6 IoT application taxonomy

The IoT versatility has led to its widespread diffusion throughout many different application domains. As defined in [95], three main application domains are distinguished and summarized in Fig. 1.3, based on the scale of the impact of generated data:

- Personal and Home domain, at the scale of an individual or home;
- Enterprise domain, at the scale of a community;
- Utilities domain, at the scale of a region or nation.

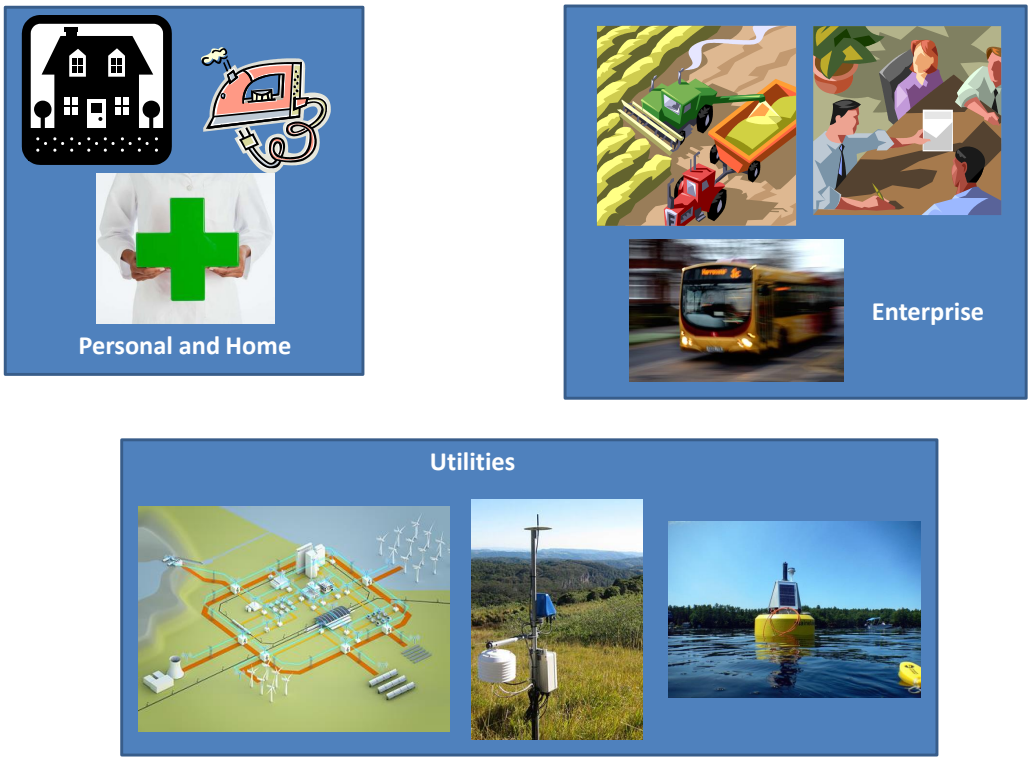


Figure 1.3: Three main IoT application domains.

In the following, some representative application examples will be provided for each domain.

1.6.1 Personal and home domain

The main applications in the Personal and Home domain are: ubiquitous healthcare systems, assisted living systems, power management systems, and social networking systems.

Ubiquitous healthcare systems are characterised by pervasive continuous vital sign data collection. These vital signs are monitored by means of body area sensors [96], which are limited in memory, energy, computation and communication capabilities. Data are then sent to

electronic devices such as smartphones, tablets, or laptops, through wireless interfaces such as Bluetooth or ZigBee, where they are preprocessed. Finally, preprocessed data are sent through the Internet to [97]: a server to be recorded into a database; doctors as automated alerts; hospitals as automated emergency alerts. Ubiquitous healthcare raises three basic challenges: sensors collect a huge amount of raw data, which need to be preprocessed to remove noise, disambiguate data and check consistency; information about meaningful physiological parameters is derived from complex models, and this requires significant computing capacity; based on their urgency, data messages need different priorities in terms of their degree of importance for diagnosis, to avoid network traffic congestion and to maximize reliability.

Analogous to ubiquitous healthcare system applications are assisted living systems [98]. These systems use information technologies to support people to increase their autonomy and quality of life, being sensitive, adaptive, and responsive to their needs, habits, gestures and passions [99]. Sensors are equipped inside the house, in order to improve the detection of anomalies or behavioral changes. Data collected by sensors are then processed to produce useful information that can be used to trigger actuators.

A considerable amount of smart home applications consist of power management systems, which monitor the power consumption of the devices inside the house and are capable of turning them on and off automatically to reduce energy consumption [100]. In this kind of systems, appliances are connected together through a short range Intranet, usually based on ZigBee technology [101]. ZigBee modules gather appliance statuses and send them to a computer or a server to be processed. After data have been processed, commands to turn on/off are sent to the appliances.

An emerging application within the Personal domain is that of social networking systems [102]. In this kind of network, objects try to emulate people and create social interactions. The aim is to guarantee an effective network navigation, and to establish a level of trustworthiness that is proportional to the degree of interaction among friend nodes.

1.6.2 Enterprise

The main applications in the Enterprise domain are: smart retail systems, smart factories, smart mobility systems, and smart agriculture systems.

Smart retail applications are conceived to optimize the sales process in retail stores [103]. One of their enabling technology is represented by RFID tags, which are placed on products. Further information related to each product are stored on a server. This information is usually accessed by means of mobile phones or tablets, which scan the RFID tag using the Contact-

less Communication API and send a request through the mobile data connection (e.g. GPRS, UMTS).

IoT can affect factory operations in many different forms: from detection of risky conditions to efficient logistics management [104], from task scheduling to machine states monitoring [105]. In this kind of scenario, sensors are ubiquitously placed to monitor the environment. Data collected by the short range network is sent to a computer or a server to be processed. On the basis of decisions made after the processing, actuators can be triggered. In such a complex scenario, some messages such as scheduling of urgent tasks or alerts of risky situations need more priority than others.

Among the IoT applications that will mostly impact everyday life are smart mobility systems [106]. In smart mobility systems, an intelligent traffic management is achieved, as well as emergency situations handling. Traffic information are collected by sensors placed on all the elements of the transport system, i.e. vehicles (whether private or public), roads, and people. Vehicular sensors gather information about the vehicle status (e.g. speed, location, direction); road information include weather conditions and vehicle traffic acquisition. People involved are drivers, passengers and pedestrians, which are usually equipped with devices such as smartphones, tablets, and navigation terminals. Such a complex system is mostly based on mobile communications of big amounts of data, which need to be processed (usually by a remote server). Since emergency situations need to be efficiently handled [107], an accurate message prioritization need to be studied.

In smart agriculture systems, precision agriculture is accomplished with the objective of increasing agricultural production and reducing environmental pollution caused by abusing agricultural chemicals [108]. Smart agriculture combines data gathered by sensors placed under the ground surface and location returned from the GPS system, with information of the Web GIS (Geography Information System). Web GIS is a server used to store, process, analyse, display and apply spatial data. On the basis of data collected by the short range network and Web GIS information, a processing is made in order to rationalise plants and scientifically predict agriculture disasters, thus increasing production.

1.6.3 Utilities

The main applications in Utilities domain are smart grid, media based IoT, and environmental monitoring.

The idea on the basis of smart grids is perfectly in line with Horizon 2020 principles. Indeed, the objective of smart grids is to enable a highly efficient energy production, transport

and consumption from the source to the user [109]. Smart grids are based on smart meters, which collect usage information from home appliances and send them to a computer or a server through a short range network [110]. Data are then processed or aggregated and sent to an application server, where decisions are made in order to dynamically match utility supply with the demand.

An emerging application is that of media (principally video) based IoT, mostly used for surveillance purposes, based on sophisticated video analysis [95]. In video based IoT a huge amount of data is gathered by video cameras and microphones, and then sent to a computer or a server. A complex processing is performed on data, in order for automatic behaviour analysis and event detection to be performed.

Environmental monitoring IoT applications are the most direct evolution of monitoring WSNs [111]. They are used to detect anomalies and keep track of parameters that describe the environment conditions, in particular for water, soil and air monitoring, as well as for infrastructure monitoring. Data acquired from sensors are usually aggregated and then sent to a server to be processed. Resulting information is analysed to detect potential threats or to establish strategies that may enhance the quality of the environment.

Chapter 2

Middleware layer functionalities for IoT task allocation

In the depicted scenario described in the introduction, it frequently happens that some nodes perform the same sensing operation, such as the measurement of the traffic in the same street, the measurement of the humidity and/or the temperature in a room, the detection of moving objects/persons in a given environment, the monitoring of the luminosity in a public square. However, not all nodes have usually the same amount of resources to be dedicated to the same tasks and the set of nodes that can cooperate in performing a given operations changes quickly as opportunistic behaviours make the scenario quite dynamic.

Accordingly, groups of nodes are identified, namely, *task groups*, that perform similar and replaceable tasks. To understand the meaning of *task group*, suppose, for example, that the network is performing a temperature sensing in a specific area: only those nodes that are equipped with a temperature sensor and that are deployed within that area are included in the *task group* related to this task. These *task groups* are assigned with the relevant task by the application deployment server, which could decide which exact node should perform each needed task. Alternatively, it may leave these groups of nodes to autonomously decide how to distribute the burden of tasks among them without the need for the central server to keep the role of single physical node controller. According to the latter vision, the IoT is made of *virtual objects* (VOs) [112] which are activated by the Central Deployment Server. The VO role may be implemented by a node in the *task group* and is in charge of processing the requests generated by the central server and forwarding configuration messages to the other physical nodes (note that the virtual node may coincide with the only single physical node that is capable of implementing the required task). At this point, allocating the proper resources to the required task is a duty of the nodes in the *task group*.

Fig. 2.1 provides a sketch of the above described reference scenario. The central server, or a leader node, transmits the activation signal to the VO. Since the VO is responsible for keeping

track of the physical nodes that belong to the same *task group* it leads, it knows which nodes the activation signal is addressed to. Therefore, it is able to forward the activation signal to the appropriate nodes, on the basis of their belonging to a determined *task group*.

To build this system, a middleware that supports described features should be designed. In the following is discussed the contribution towards the definition of the middleware functionalities that are needed in the IoT towards the allocation of tasks among different objects. In the proposed middleware, these functionalities should be mostly implemented at the VOs, so that these are augmented with additional capabilities to coordinate the selection on which tasks to be performed by each of the members of the *task groups*.

The proposed role of the VOs is described in the first Subsection, whereas the resulting middleware architecture is presented in the second Subsection.

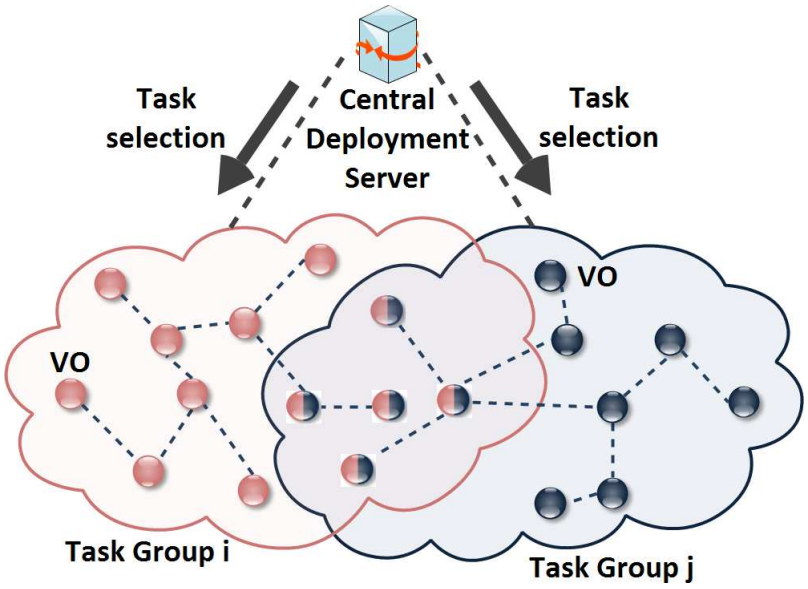


Figure 2.1: The reference scenario.

2.1 Role of virtual objects

The reference scenario is that of an opportunistic IoT, where nodes continuously join and leave the network. In particular, this work focuses on opportunistic network, which autonomously choose to participate in an opportunistic way to some sensing tasks they are able to perform. The algorithms that will be presented in the following allows for nodes to dynamically negotiate the effort they put into performing a task, in terms of network resources. With the proposed protocol, nodes involved in the same task, i.e. belonging to the same *task group*, converge to a common goal, i.e. the frequency at which a task is executed, and to the same local buffer usage.

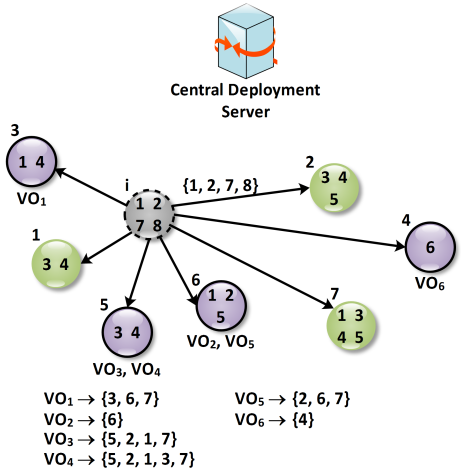
In order for nodes to start a negotiation, they need to have already joined the related *task*

group. This join procedure will now be explained in detail. As soon as a node i joins the network, it broadcasts to its one-hop neighbours the information related to the tasks that it is able to perform. Accordingly, the VOs related to those specific tasks add node i to the list of nodes that belong to their *task groups*, and reply with an acknowledgment. If node i is the second node in that list, they designate it as the vice-VO, and the acknowledgement contains this information. If no VO is associated to one or more tasks yet, node i is designated as VO for those tasks. Then, node i notifies to the Central Deployment Server its designation as VO and/or vice-VO. As shown in Figure 2.2, suppose, for example, that node i is able to perform 4 tasks. Thanks to the middleware running on top of the network, the description of those tasks is converted to 4 different identification numbers, corresponding to the identification numbers of the equivalent *task groups*. Suppose that, in this example, node i is associated to *task groups* $\{1, 2, 7, 8\}$, the VOs VO_1 and VO_2 are associated to *task groups* 1 and 2, and that no VO has been associated to *task groups* 7 and 8, yet. Thus, VO_1 and VO_2 will add node i to their list of nodes. Furthermore, VO_2 acknowledges to node i its designation as vice-VO for *task group* 2. Obviously, node i will not receive acknowledgements for *task groups* 7 and 8. Hence, node i will assume the role of VO_7, VO_8 and vice-VO_2, and it will inform the Central Deployment Server.

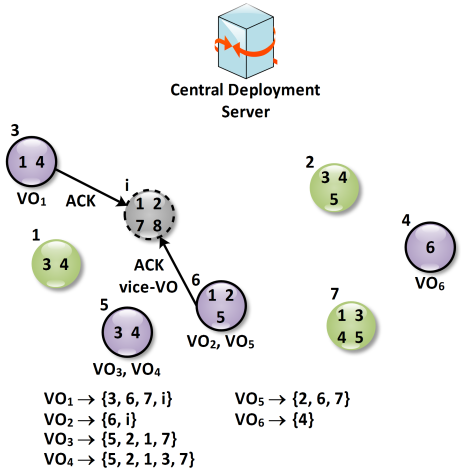
VOs periodically send Hello messages to their related vice-VOs. One of the following things might happen:

- The VO sends the Hello message and the vice-VO acknowledges the message. No further actions are performed.
- The VO sends the Hello message and the vice-VO does not acknowledge the message. In this case the VO assumes that the vice-VO is not reachable. If present, the VO designates the second node on its list as vice-VO and informs it, which in turns inform the Central Deployment Server.
- The VO does not send the Hello message when it is supposed to do it. The vice-VO notices that the VO is not reachable. It broadcasts a request to know which nodes belong to its *task group*. The first one to reply is designated as vice-VO. Information about the failed VO and the new vice-VO is delivered to the Central Deployment Server.

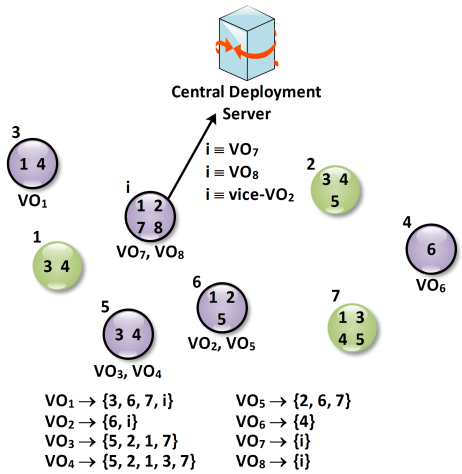
In order to avoid communication overhead, when a VO notices that it is about to leave the *task group* (e.g. for depletion of residual energy, or because it is moving) it notifies it to its vice-VO, sending the list of the nodes belonging to the *task group*. Then, the vice-VO becomes the VO, and the next node on the list becomes the vice-VO. Relevant information is delivered to the Central Deployment Server.



(a) Node i joins the network



(b) VO_1 and VO_2 send ACK to node i



(c) Node i becomes VO_7, VO_8 and vice-VO_2

Figure 2.2: Sequence of steps when a new node i joins the network

When an application requires the execution of a given task, the Central Deployment Server sends an activation signal to the appropriate VOs, which forward it to their list of nodes. Then, the negotiation algorithm is started. If the Central Deployment Server does not succeed in reaching a VO, it tries to contact its vice-VO. If even the vice-VO does not reply, the Central Deployment Server assumes that no nodes are available anymore for the related task, and thus it cannot be activated. Once the VO has sent the activation signal to all the nodes in the group, then they start the consensus algorithm. What happens is that each node sends to all its neighbours (one-hop broadcast) the consensus messages for the reference *task group*, containing data that specifies the update values for the algorithm. Then they check whether the *task group* to which this message is related is of interest to them, i.e. they belong to this *task group*. If yes, they process the data further and then exploit this information in the next iteration of the algorithm. Note that they do not need to keep a list of nodes that are taking part to the negotiation, so there isn't waste of memory to maintain a table of neighbours that take part to the same group. Independently from the number of direct neighbours, each member of the consensus algorithm will reach the convergence if the general communication graph, in the network, is connected. Accordingly, it is necessary that the nodes in the consensus are interconnected but there is no need for a full-mesh connectivity.

2.2 The reference middleware

In order for devices to communicate and interoperate, a common middleware is proposed, which has to be able to manage objects' and tasks' discovery, and to allocate tasks to objects so that resource exploitation is shared. Figure 2.3 shows the reference middleware architecture.

Two layers are proposed. The *Semantic layer* requires the adoption of semantic technologies for the description of: objects capabilities and characteristics, such as capability to sense temperature or energy consumption to process an instruction; application subdivision into tasks and definition of task characteristics; requirements, such as QoS/QoI (Quality of Information) parameters; network characteristics such as the communication protocol used and objects' deployment. All this information is exchanged by the objects and, on the basis of this information, the *task groups* are created. This activity is performed by the *Task Group Management* block, in the *Resource Allocation and Management layer*. Then, task allocation to network objects is carried out by the Allocation block.

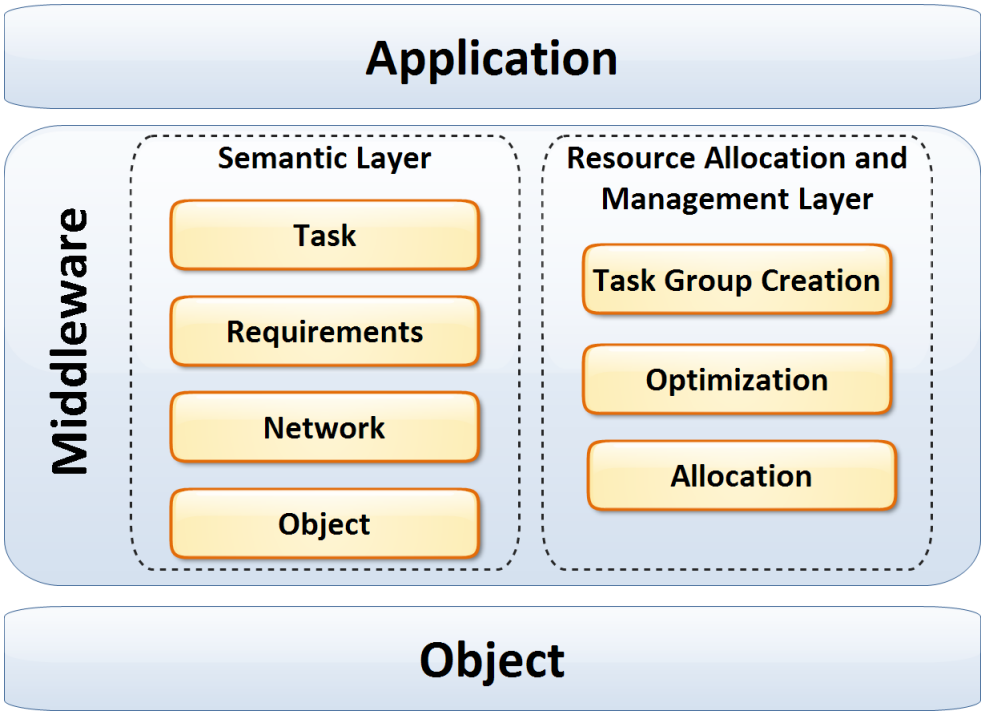


Figure 2.3: The reference middleware.

2.2.1 The semantic layer

The semantic description proposed is based on the ontologies presented in [13][14][113]. In particular, the Semantic Sensor Network (SSN) ontology [113] is used to model sensor parameters, resources, services, and QoS/QoI related parameters. Fig. 2.4 shows an overview of the modules needed.

Network: The Network module defines network characteristics such as communication protocol and topology. Furthermore, it defines how network objects are connected.

Object: The Object module provides the description of object characteristics and capabilities. Objects can be sensors, actuators, processors, storage devices, tags, or a combination of these. Object resources are described in the Resource module. The Location module provides information about object position. Sensor devices are also related to the Task module. This relation describes the tasks that the object is able to perform.

Resource: The Resource module describes the type of resource associated to an object, and its related parameters. Since resources are needed for task execution, the Resource module is related to the Task module.

Application: The Application module defines the characteristics of applications assigned to the reference IoT network. It also determines relations among tasks, i.e. the sequence of tasks in which the application is decomposed.

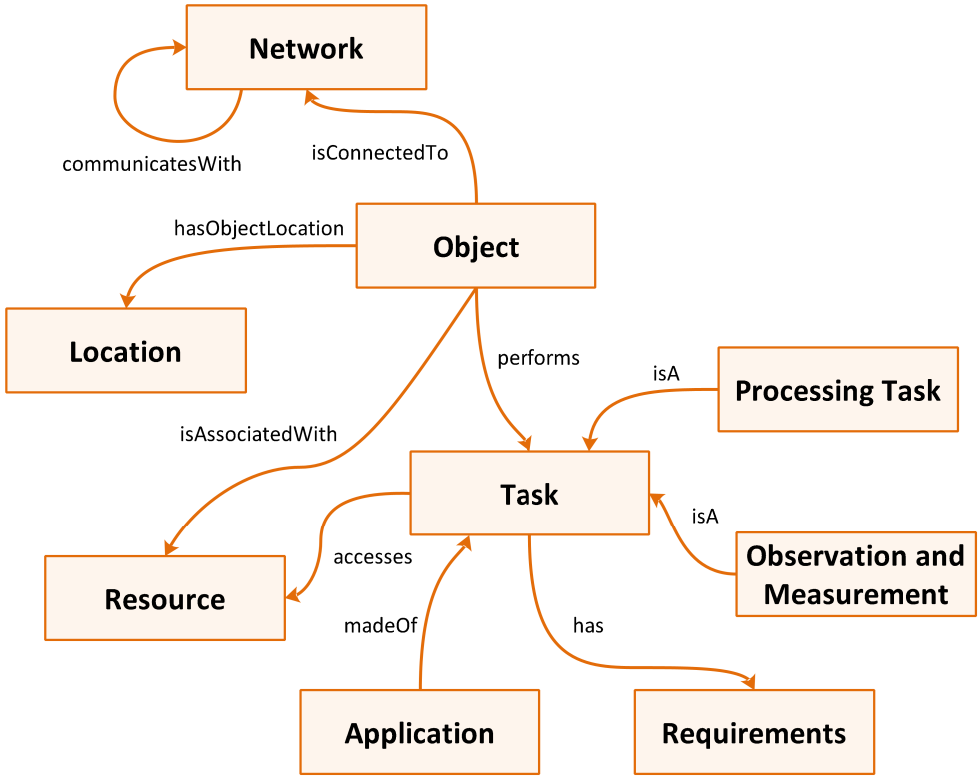


Figure 2.4: Overview of the modules in the ontology.

Task: The Task module provides an interface for the interaction between objects and their related processes. In particular, it defines how object resources are allocated to enable task execution, and which requirement (i.e. QoS/QoI) characteristics are needed for that task. IoT tasks are modelled using the most widely used Web service languages, such as Unified Service Description Language (USDL), Web Service Definition Language (WSDL), and Web Application Description Language (WADL). Tasks can be either processing tasks or sensing tasks, as reflected by the relations with the Processing Task module and the Observation and Measurement module.

Processing Task: The Processing Task module describes the characteristics associated to a specific processing task, how processing is performed on input data, and what data is delivered as output.

Observation and Measurement: The Observation and Measurement module describes how data are generated by sensor devices within the IoT scenario. Data are typically needed to perform tasks useful for the network application execution. When a certain QoS/QoI is required on measured data (e.g. a given data accuracy), the Observation and Measurements characteristics must enable the fulfilment of these requirements.

Requirements: The Requirements module defines the QoS/QoI constraints that gathered measurements and provided tasks must fulfill. Since it is not always required, this is not a

mandatory module.

Location: The Location module makes use of the GeoName ontology [114] to define object positions. Since object position is not always needed, this is not a mandatory module.

2.2.2 The resource allocation and management layer

Information collected by the Semantic layer is used by the Resource Allocation and Management layer to identify *task groups* and to correctly allocate resources to the network objects. As described in previous section, *task groups* are created according to the Central Deployment Server, which assigns nodes to the appropriate *task group* on the basis of their capabilities and application requirements.

Resource allocation is performed by the Allocation block in a distributed way, using a protocol that is based on totally distributed algorithm. The two proposed protocol, described in the following chapters, use this architecture.

Chapter 3

Strategy for an homogeneous resources consumption

In this chapter is presented a distributed optimization protocol based on consensus algorithm. The main goal of this solution is to solve the problem of resource allocation and management in IoT heterogeneous networks. The proposed protocol is robust against links or nodes failures, so it's adaptive in dynamic scenarios where the network topology changes in runtime, like a typical IoT scenario. The challenges faced in this chapter is the deployment of distributed applications in the IoT in terms of cooperation among objects, with the aim of distributing the burden of the execution of the application committed to the network, so that resources are adequately exploited.

To face the challenge, a distributed protocol based on the consensus algorithm proposed in [18] is adapted to solve the problem of resource allocation and management in IoT networks. In particular, the use case analysed focuses on adjusting sensing functionalities of objects so that resources are equally shared among nodes participating into the application execution. Simulation and real scenario results prove that the convergence of the consensus algorithm is quickly reached.

This chapter is organized as follows. In 3.1 the resource utilization model used for analysis and protocol design is presented. Section 3.2 describes the distributed algorithm, based on consensus, that is used to adjust the frequency with which the task is executed. In the last section 3.3 three different applications of the proposed protocol are presented.

3.1 Resource utilization model

The design of protocol stars from the algorithms described in [18], which has been devised for clock synchronization among nodes. Accordingly, the model has been completely rewrite to be

adapted of the scenario, which now focuses on date generation and power consumption as a function of the task frequency, as described in the following first Subsection. The implementation of the protocol in terms of data exchanged among nodes and how this is used to agree on a common frequency are modified, too.

Each node i that performs task k collects data with frequency $f_{i,k}(t)$. The power consumed by node i is expressed by:

$$P_{i,k}(t) = E_{i,k} \times f_{i,k}(t) \tag{3.1}$$

where $E_{i,k}$ is the energy per task execution spent by node i for task k. Let N nodes perform task k. The total power consumption for task k is the following:

$$P_k^c(t) = \sum_{i=1}^{N} P_{i,k}(t) \tag{3.2}$$

Whereas the total power consumption for node i:

$$P_i^c(t) = \sum_{k=1}^{N_i} P_{i,k}(t) \tag{3.3}$$

where N_i is the total number of tasks performed by node i.

Similarly, the amount of data collected by node i at time t for task k is:

$$D_{i,k}(t) = B_k \int_0^t f_{i,k}(x)\, dx + M_{i,k}(t) \tag{3.4}$$

where B_k is the amount of output data for task k, and $M_{i,k}(t)$ is the occupancy of node i's storage buffer at time t. $M_{i,k}(t)$ can be controlled by data fusion operations or transmitting data to the sink.

Similarly, the total amount of collected data due to node i, for all tasks, at time t is:

$$D_i(t) = \int_0^t \sum_{k=1}^{N_T} B_k f_{i,k}(x)\, dx + M_i(t) \tag{3.5}$$

where N_i is the total number of tasks executed by node i.

Now a simplification of the models (Eqs. (3.4),(3.5)) of the data sensing process are provided. By considering the first order dynamic of node i, which is written as follows (from here also subscript k is dropped to simplify the notation):

$$\phi_i\,(t) = \lambda_i t + \iota_i \tag{3.6}$$

where ϕ_i is the number of samples collected by node i, λ_i is the local task slope which determines the task frequency, and ι_i is the local bias that describes the number of samples stored in

the node buffer. Note that when applying the simplification to Eq.(3.4), the analysis is restricted to the data collected only for a specific task, otherwise the data total amount of data for all the tasks executed in a node is considered.

Recall that the objective is to obtain the same virtual dynamic on all nodes, which is represented with the following equation:

$$\phi_v(t) = \lambda_v t + \iota_v \tag{3.7}$$

Every node keeps an estimation of the virtual dynamic using a linear function of its own local function:

$$\tilde{\phi}_i = \tilde{\lambda}_i \phi_i + \tilde{o}_i \tag{3.8}$$

The goal is to find $\tilde{\lambda}_i$ and $\tilde{o}_i$ that compensate the difference among all node dynamics, and thus to converge to the virtual dynamic in Eq.(3.7). So, for each node i, the aim is to obtain that: $\tilde{\phi}_i \rightarrow \phi_v$.

In the following is described how this goal is achieved.

3.2 Consensus algorithm within the task group

The consensus algorithm is implemented by the nodes within each *task group* and entails an iterative procedure that updates the slope and bias values. At each iteration other than the new slope and bias values, each node computes a new $\phi_i(t)$. The updates of the function $\phi_i(t)$ are stored by the node as these are required by the estimation algorithm.

To estimate the new slope value, every node i estimates the relative slope with respect to its neighbours j as: $\lambda_{ij} = \dfrac{\lambda_j}{\lambda_i}$ The value of λ_{ij}, according to [18], can be estimated as:

$$\gamma_{ij}^{+} = \rho_n \gamma_{ij} + (1 - \rho_n) \frac{(\phi_j(t_2) - \phi_i(t_2))}{(\phi_j(t_1) - \phi_i(t_1))} \tag{3.9}$$

where γ_{ij} is the appraisal of relative slope, γ_{ij}^{+} indicates the update of variable γ_{ij} and $\rho_n \in (0, 1)$ is a tuning parameter to compensate the noise. In [18] Proposition 1 demonstrates that $\lim_{t \to \inf} \gamma_{ij}(t) = \lambda_{ij}$. Therefore, each node sends to its neighbours only the local counter $\phi(t)$. The related amount of data is very small, and it can be put inside another data packet producing a small overhead. From the point of view of the buffer occupancy, each node stores five variable for each neighbour: $\phi_j(t_2), \phi_i(t_2), \phi_j(t_1), \phi_i(t_1), \gamma_{ij}$.

As soon as the node estimates the new value of the relative slope and as soon as node i receives a packet from node j, it updates the value of $\tilde{\lambda}_i$ according to:

$$\tilde{\lambda}_i^{+} = \rho_v \tilde{\lambda}_i + (1 - \rho_v) \gamma_{ij} \tilde{\lambda}_i \tag{3.10}$$

where ρ_v is a tuning parameter. For simplicity, in this work the initial condition is setted as $\tilde{\lambda_i}(0) = 1$.

The Eq.(3.10) can be expressed in matrix terms to demonstrate some important advantages of the algorithm. The vectors $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N)^T$ and $1 = (1, 1, \ldots, 1)^T$ can be defined, and Eq.(3.10) can be written as follows:

$$\lambda^+ = A\lambda \tag{3.11}$$

The matrix $A \in R^{NxN}$ is made by all ones on the diagonal and zeros in the other side except the terms: $A_{ii} = \rho_v$ and $A_{ij} = 1 - \rho_v$, that correspond with nodes involved in the communication. Considering this form of matrix A it's possible that the matrix satisfies two important conditions: $A_{ij} \geq 0$ and $A1 = 1$, that make a stochastic problem. To obtain the convergence of the algorithm, two proprieties about communication are considered: there is a time window T in which each node i communicates with another node at least once and there is a communication path from each node i to any other node j (communication graph strongly connected). Noted all these proprieties, $\lim_{t \to \text{inf}} \tilde{\lambda_i} \lambda_i(t) = \lambda_v$ is obtained. A very interesting read on the papers [18] and [17] is addressed for a summary of many results and details on the conditions for convergence.

The previous demonstration entails some important remarks. The first is that the node transmission order is not important, nor is the exact moment the transmission takes place. So, this implies that the protocol is fully asynchronous and nodes can transmit at different rates. The only important condition is that the graph is sufficiently closely connected. Advantage of the asynchronous execution is that it allows the node to enter into sleep-mode (and then save battery energy), because it is not needed that each node replies instantly to the other updates. Accordingly, the nodes can wait until other data needs to be sent to the neighbours so the parameters can be sent within the relevant packets. Another important observation is that if any message is lost, the condition of strongly connected graph is still guaranteed. This implies that the algorithm is robust even against link failures, nodes failures and packet collisions. So the proposed protocol is very adaptive in dynamic scenario where the network topology change in runtime. From the performance point of view, with reference to transmitted and stored variables introduced previously, each node sends only the virtual slope estimation $\tilde{\lambda_i}$, which is a low amount of data. Also in this case, these parameters can be put inside a data packet, producing a small overhead. From the point of view of the node buffer, only one variable per each neighbour is needed: $\tilde{\lambda_j}$.

As to the bias compensation, it is performed in a way similar to the slope estimation:

$$\tilde{\iota_i}^+ = \rho_b \tilde{\iota_i} + (1 - \rho_b)(\tilde{\lambda_j}\phi_j + \tilde{\iota_j} - \tilde{\lambda_i}\phi_i - \tilde{\iota_i}) \tag{3.12}$$

where ρ_b is a tuning parameter.

Eq.(3.12) has the same structure of Eq.(3.10), so if it's done the same hypothesis of the previous, all ι_i converge to the same value. From Eq.(3.12) follows that this algorithm transmits only the virtual bias estimation $\tilde{\iota_i}$, and it stores only one variable for each neighbour: $\tilde{\iota_j}$. Considering two properties about protocol the convergence is demonstrated. The first is that there exists equation $\lim_{t\to\inf} \tilde{\lambda_i}\lambda_i(t) = \lambda_v$ and also there exists $\lambda_v > 0$ for all i and t. The second is that suppose there is a time window T in which each node i communicates at least once the couple of data $\tilde{\lambda_i}, \tilde{\iota_i}$ (communication graph strongly connected). If all these conditions are true, the proof of the convergence of the algorithm is done. In [18] the proof and the speed of convergence are demonstrated with more details.

3.3 Application of the consensus protocol

In this section three different applications of the consensus protocol: single task - single frequency; single task - total frequency; entire network.

Single task - single frequency

The objective is to make the objects in the *task group* to agree on a common frequency for the execution of only the relevant task. This approach brings to a distribution of the burden for the execution of each specific task but does not take into account the fact that there may be some objects already involved in the execution of other tasks that then may become over-utilized. Accordingly, the considered dynamic is that of Eq.(3.4) that depends on frequency $f_{i,k}$. The new frequency set at each iteration $z + 1$ is:

$$f_{i,k}^{z+1} = \tilde{\lambda_{i,k}} f_{i,k}^z \tag{3.13}$$

where $\lambda_{i,k}$ is evaluated by Eq.(3.10). Whereas the offset compensation is:

$$M_{i,k}^{z+1} = \tilde{\lambda_{i,k}} M_{i,k}^z + \tilde{\iota_{i,k}} \tag{3.14}$$

where $\tilde{\iota_{i,k}}$ is evaluated by Eq.(3.12).

After the convergence, it's possible to focus on the impact on resources. Assuming that node i energy per task execution $E_{i,k}$ value (Eq.(3.1)) is the same for each node involved in task

k, after the proposed protocol converges, node i task frequency will be:

$$\forall i = 1, 2, ..., N \Rightarrow f_{i,k} = f_k^m$$

while the buffer occupancy will be:

$$\forall i = 1, 2, ..., N \Rightarrow M_{i,k} = M_k^m$$

It follows that the total power consumption (Eq.(3.2)) for task k have the same value as before the algorithm execution. Nevertheless, after the consensus protocol converges, resources are equally shared by nodes involved in the same task.

Single task - total frequency

The objective of this approach is to make the nodes agree on the frequency for the execution of all tasks, i.e, f_i^c. For this reason the linearization is applied to the process modelled in Eq.(3.5). The new frequency set in each iteration $z + 1$ results:

$$f_i^c = \sum_{k=1}^{N_T} f_{i,k}^{z+1} = \sum_{k=1}^{N_T} \tilde{\lambda_{i,k}} f_{i,k}^z \tag{3.15}$$

where N_T is the total number of tasks processed by node, while the offset compensation:

$$M_i^{z+1} = \sum_{k=1}^{N_T} M_{i,k}^{z+1} = \sum_{k=1}^{N_T} (\tilde{\lambda_{i,k}} M_{i,k}^z + \tilde{\iota_{i,k}}) \tag{3.16}$$

The impact on resources after the convergence, assuming that nodes participate to multiple *task groups*, will be now evaluated. The goal is to share resources in the network, and not only in the *task group*. The protocol is oriented on reaching the same frequency of all tasks processing for each node. So after the proposed protocol converges it will be:

$$\forall i = 1, 2, ..., N \Rightarrow \sum_{k=1}^{N_T} f_{i,k} = f_i^m$$

while the buffer occupancy will be:

$$\forall i = 1, 2, ..., N \Rightarrow \sum_{k=1}^{N_T} M_{i,k} = M_i^m$$

Assuming that node i's energy per task execution value $E_{i,k}$ (Eq.(3.1)) is the same order of magnitude for each task processed in the network, after the proposed protocol converges it's obtained that the total power consumption (Eq.(3.3)) for each node i have the same value. Nevertheless, after the consensus protocol converges, resources are equally shared by nodes involved in the network.

Entire network

The objective of this approach is to achieve an inter-group agreement. It works as the previous algorithm single task total frequency, with the difference that the execution of the protocol in a group may trigger the execution of other instances. Assume to be in a steady state so that in the network a number of tasks (say tasks A, B and C) have already been assigned and the consensus on the total frequency reached. Consider then that the request of a new task D arrives and the new consensus protocol activated by the relevant VO. It's also considered that some nodes in the *task group* D also belong to *task group* A. Accordingly, the activation of the consensus among nodes in group D triggers the re-execution of the consensus among the nodes in group A. In this way the total frequency is (almost) uniformly assigned to nodes in both groups A and D. If other groups overlap with A, these are recursively re-activated by the execution of the protocol in A. In this way the execution is propagated through the network as far as a tasks' overlap exists. Clearly in this way the energy consumption for the execution of the protocol significantly increases in this application.

Chapter 4

Strategy for preserving lifetime and QoS

Before providing the required information about the physical world, objects involved in IoT coordinate with the other objects in *groups* and provide a unified service to the external world (the application that requires the service), with the intent to distribute the load of the requested services according to specific community defined rules, which could be: lifetime extension, QoS (Quality of Service) maximization, reward maximization, or others. In this chapter other than describing the characteristics of this new communication paradigm and challenges it is called to address, It's also proposed a first solution for its implementation that relies on a distributed optimization protocol based on the consensus algorithm.

The challenges faced in this chapter is the deployment of distributed applications in the IoT in terms of cooperation among objects, with the aim to solve the problem of resource allocation and management preserving the required QoS.

To face the challenge, a distributed protocol based on the consensus algorithm ids designed. In particular, the use case analysed focuses on a IoT scenario with many heterogeneous nodes involved in the same task. The proposed protocol allowed for improving the lifetime network because each node's lifetime tend to a value that is equal for all the nodes involved in the same task.

This chapter is organized as follows. In 4.1 the problem is modelled. The proposed model design a bond between power consumption and frequency of task execution. Section 4.2 describes a totally distributed protocol based on average consensus algorithm that is used to achieve lifetime and QoS preservation.

4.1 Agreement on task frequency among nodes

Each node i that performs task k collects data with frequency $f_{i,k}(t)$. The power consumed by node i to perform task k is expressed by:

$$P_{i,k}(t) = E_{i,k} f_{i,k}(t) \tag{4.1}$$

where $E_{i,k}$ is the energy spent by node i for task k. Assuming to keep the total number of samples per second generated by all the nodes in the *task group* constant and equal to the application requirement F_k, $f_{i,k}(t)$ can be changed by nodes within the same *task group* to find the optimal combination that distributes the task burden among them. Energy $E_{i,k}$ can be split in different contributions such as sensing energy $E_{i,k}^{sens}$, transmission energy $E_{i,k}^{tx}$ and computing energy $E_{i,k}^{comp}$:

$$E_{i,k} = E_{i,k}^{sens} + E_{i,k}^{tx} + E_{i,k}^{comp} \tag{4.2}$$

Each node i is also associated to a residual energy $E_i^{res}(t)$ that depends on its residual battery charge – and thus decreases with time –, and on its lifetime $\tau_i(t)$, which is the time before running out of battery. Node i's lifetime is expressed as:

$$\tau_i(t) = \frac{E_i^{res}(t)}{\sum_k P_{i,k}(t)} \tag{4.3}$$

In the proposed solution the goal is to reach the higher network lifetime, which is the time before the first node fails. This objective is equivalent to the target of having uniform objects' lifetimes in the community. Considering only task k, the contribution in lifetime of node i to task k is defined as:

$$\tau_{i,k}(t) = \frac{E_i^{res}(t)}{P_{i,k}(t)} = \frac{E_i^{res}(t)}{E_{i,k} f_{i,k}(t)} = \frac{1}{c_{i,k}(t) f_{i,k}(t)} \tag{4.4}$$

In order for nodes to tend to a uniform lifetime, this contribution should tend to an amount τ_k that is equal for all the nodes involved in task k's execution, i.e.

$$\lim_{t \to \infty} \tau_{i,k}(t) = \tau_k \qquad \forall i \in X_k \tag{4.5}$$

To lighten the notation, in Eq. 4.4 $c_{i,k}(t)$ is added as a cost function that reflects the extent to which the node can be used, according to the ratio between its energy consumption for task k and its residual energy. At the same time t if $c_{i,k}(t) < c_{j,k}(t)$, i can be used more extensively than j.

For a graphical representation of the problem at a time t, the residual energy $E_i^{res}(t)$ and the consumed power $P_{i,k}(t)$ can be drawn on the x and y axes, respectively. As shown in Fig. 5.7,

mapping all combinations for all the nodes, a constellation of points is obtained where each node has a different lifetime $\tau_{i,k}(t)$, which depends on the initial node battery status (yellow points in Fig. 5.7). By applying Eq. 4.5 these points are forced to move into a a straight line with a slope τ_k by changing each frequency $f_{i,k}(t)$, so that each node will have the same lifetime (white points in Fig. 5.7). The required changes are driven by the community QoS target expressed in terms of the total number of samples per second provided by the community F_k, so that the following constraint is introduced:

$$\sum_{i=1}^{N_k} f_{i,k}(t) = F_k \tag{4.6}$$

When the appropriate VO forwards an activation signal to the list of interested nodes, the message conveys F_k. Eq. 4.4 can be applied into Eq. 4.6 as follows:

$$\tau_k(t) = \frac{1}{F_k \sum_{i=1}^{N_k} c_{i,k}(t)} = \frac{1}{F_k N_k \overline{C_k}(t)} \tag{4.7}$$

To reach the goal each node in the task group needs to know $\overline{C_k}(t)$, i.e. the mean value of all the $c_{i,k}(t)$ values for task k. After a node knows this value, it can evaluate the frequency that corresponds to its optimal lifetime:

$$f_{i,k}(t) = \frac{1}{c_{i,k}(t)\tau_k(t)} = \frac{F_k N_k \overline{C_k}(t)}{c_{i,k}(t)} \tag{4.8}$$

To compute the value of $\overline{C_k}(t)$ the numerosity of the *task group* is needed. This is possible because the VO has the list of interested nodes, so N_k can be forwarded at the beginning of the process by means of the activation signal. If the conditions of the network change during the task execution (e.g., a new node enter the community or a node fault happens), the nodes detecting the change flood the message so that each node is reached. A centralized solution is computationally very simple, but with respect to a decentralized solution, it requires higher transmission costs due to a lot of control messages and the system has a slower reactivity due to topology or node status (i.e. residual energy) changes.

4.2 The distributed solution based on the consensus protocol

To reach the goal, in a totally distributed way, an average consensus protocol is implemented by the nodes within the *task group* to evaluate $\overline{C_k}(t)$. The study focus on a particular class of iterative algorithms for average consensus. To estimate the average value across the network, a consensus protocol propagation, that is totally asynchronous and distributed, is used. When the

procedure starts, each node allocates a variable $\overline{c_{i,k}}(t)$ to iteratively estimate the value of $\overline{C_k}(t)$. At an initial time $t = 0$:

$$\overline{c_{i,k}}(0) = c_{i,k}(0) \qquad \forall i = \{1, \cdots, N_k\} \tag{4.9}$$

In order for the estimated $\overline{c_{i,k}}(t)$ to converge towards the correct average $\overline{C_k}(t)$ (computed on all nodes in the *task group*), each node follows the rule of the consensus protocol and updates the local estimation by adding a weighted sum of the local discrepancies, i.e., the differences between neighbouring node estimated values and its own. At each time step $t + 1 > t$, the update rule of the consensus protocol is the following:

$$\overline{c_{i,k}}(t+1) = \overline{c_{i,k}}(t) + \sum_{i=1}^{O_j} W_{i,j}(\overline{c_{j,k}}(t) - \overline{c_{i,k}}(t)) \tag{4.10}$$

where O_i defines the number of node j neighbours.

$W_{i,j}$ is a weight associated with the communication between i and j. If the weights are associated with undirected edges, the result is $W_{i,j} = W_{j,i}$. It's also possible to consider asymmetric weights, associated with ordered pairs of nodes, so the previous equality is not true. Weights have to satisfy some basic constraints, as well as the convergence condition, such as defined by Xiao et al. in [115]. So to take very simple set of weights that define a one hop communication (the node i communicates only with the neighbours):

$$W_{i,j} = \begin{cases} 1 & if \qquad j \in O_i \\ 0 & if \qquad j \notin O_i \end{cases} \tag{4.11}$$

From the described protocol follows that the $i - th$ node transmits only the value of $\overline{c_{i,k}}(t)$ and subscribes this value each iteration with the neighbour. So the protocol does not rely on extensive transmissions and the related amount of data exchanged is very small. From the point of view of the node buffer occupancy, each node stores only one variable and it subscribes this at each iteration with the neighbours, so the memory required by the protocol is very limited.

Chapter 5

Performance analysis

This chapter presents some details about the performance analysis of algorithm presented in previous chapters, analysing various cases within simulated and real scenarios.

About the organization this chapter is organized as follows. In 5.1 the performance of protocol presented in chapter 3 are evaluated. This section contain three paragraphs: 5.1.1 analyses a simulation, 5.1.2 presents a real scenario and 5.1.3 compare the impact of the three different approaches described in 3.3. The second section 5.2 analyses performance of protocol discussed in chapter 4. This section contain two paragraphs: 5.2.1 analyses a simulation, instead 5.2.2 evaluated performance in a real scenario with embedded systems and in another real scenario composed by android systems.

5.1 Protocol for an homogeneous resources consumption

This section analyses the protocol presented in chapter 3. At the first is evaluated the exchange of messages through the proposed middleware, then the convergence of the algorithm within simulated and real scenarios is analysed. Finally, the different distribution of resource when applying the different approaches described in Section 3.3 is analysed.

5.1.1 Simulation scenario

In this case study Matlab software is used to implement a framework to simulate the protocol focusing on two types of communication: i. broadcast mode and ii. gossip mode. In both cases, the topology has been created following a pseudo-random geometric distribution with reference to the geographical position, and transmissions on the network are asynchronous. The broadcast communication entails that if the node i sends a packet, this is received by all neighbours, which update their values. On the other hand, the gossip mode entails that two nodes are selected in a pseudo-random way and communicate to update their values. The choice is pseudo-random

because only two neighbours can communicate. Furthermore, for the simulation to be more realistic, a certain probability of using a given link is considered. A situation where the update values are inserted as data packet overhead is simulated. The simulation was run on 20 nodes (i.e. $N = 20$) in a pseudo-random topology. All tuning parameters are setted as: $\rho_n = \rho_v = \rho_b = 0.5$. Node dynamics are initialized with pseudo-random values of λ and ι. The assumption is that nodes transmit a total amount of 5000 packets, so on average each node transmits 250 packets. The approach "Single task - single frequency" is implemented in this simulation but the results do not depend on the approach adopted as the focus is on the convergence.

Broadcast communication

With this simulation the goal is to study the performance of the protocol in terms of convergence speed and error, considering a broadcast communication among nodes.

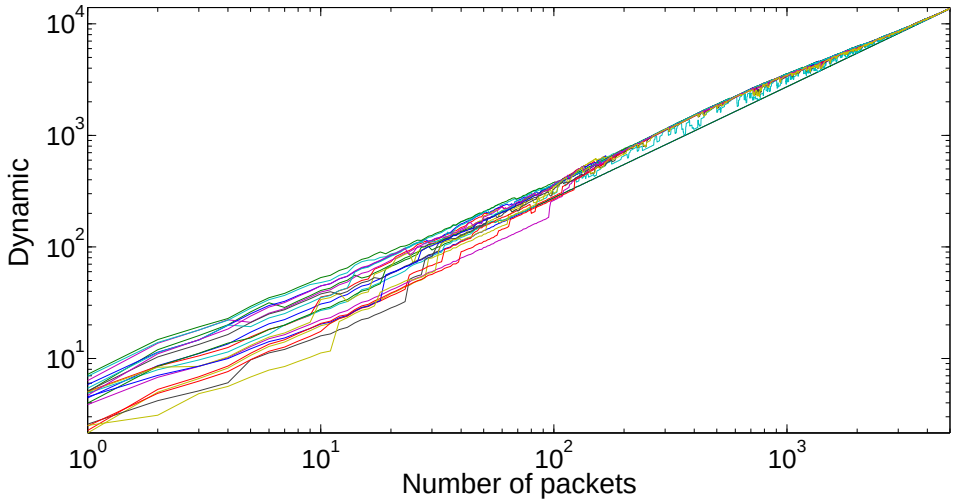


Figure 5.1: Dynamics evolution using broadcast communication.

Fig. 5.1 shows the algorithm convergence. After 100 packets transmitted on the network, (5 on average for each node) the dynamics can be considered converged. As the number of packets exchanged increases, nodes reach a better consensus. From the error point of view, Fig. 5.2 shows that initially the error is $\pm 60\%$, but after 100 packets are transmitted this value decreases by $\pm 20\%$, and eventually a very low error value of about $\pm 5\%$ is obtained.

Gossip communication

With this simulation the achieve is to study the protocol performance in a more realistic scenario. Since the information exchanged to implement the protocol is limited, as explained in Section **??**, it can be inserted in data packets. In this way, the algorithm's burden on network

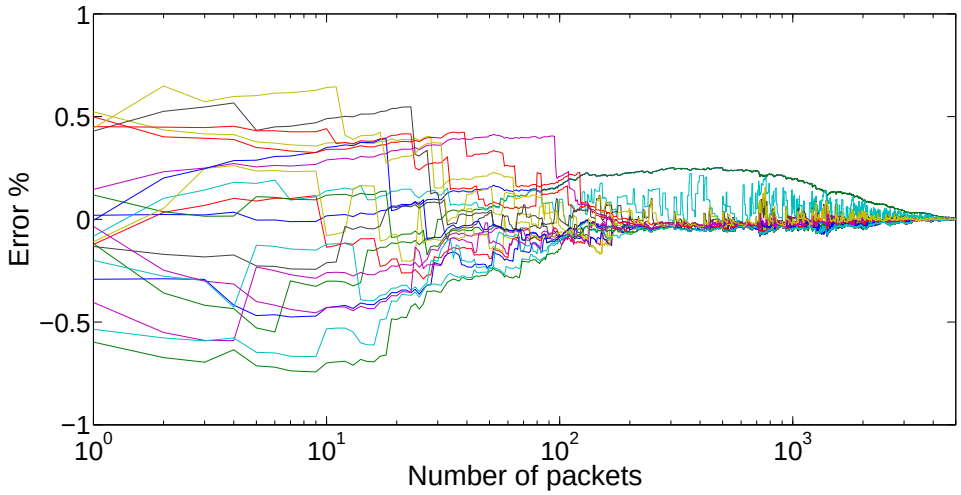


Figure 5.2: Error evolution using broadcast communication.

traffic is low, as it's possible to combine the consensus protocol with another application running on the network.

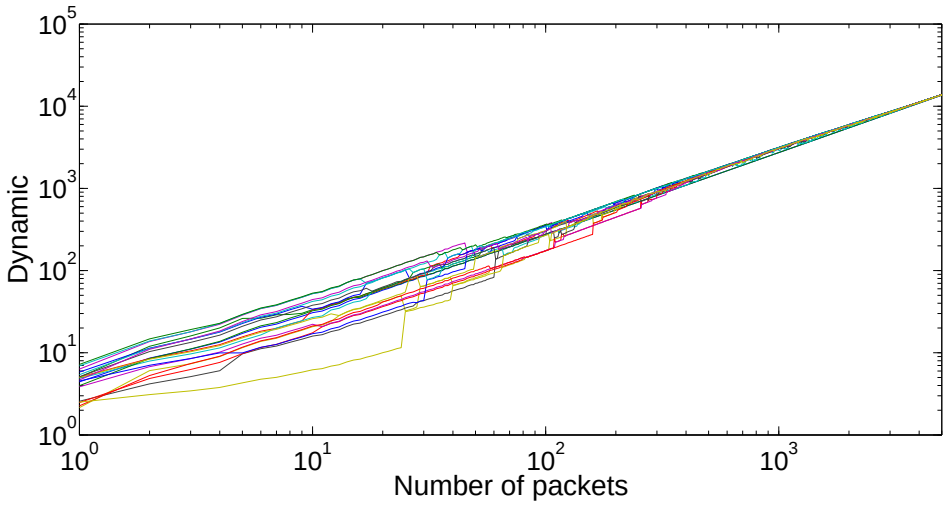


Figure 5.3: Dynamics evolution by gossip communication.

Fig. 5.3 shows the algorithm convergence. As in the broadcast communication scenario, nodes converge at consensus, but in this case convergence is slower. Fig. 5.4 shows this in more detail. When the packets transmitted on the network are about 100, the error is $\pm 40\%$, and this value decreases more slowly than in the broadcast communication scenario. This happens because in the gossip mode, at each iteration, the communication is enabled only between two neighbouring nodes, so only these two nodes update their values. On the other hand, in the broadcast communication scenario all neighbours update their values simultaneously whenever a node transmits a packet.

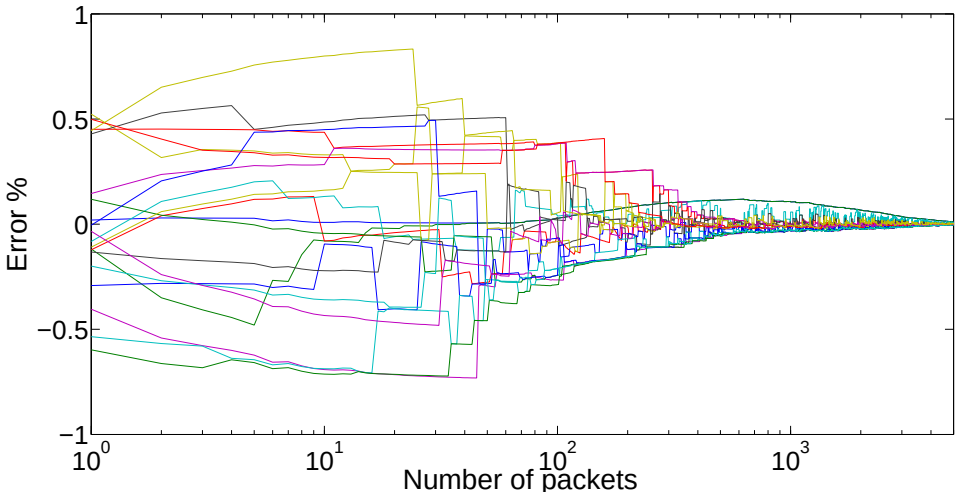


Figure 5.4: Error evolution by gossip communication.

5.1.2 Real scenario

The last experiment consists in the study of the protocol performance in a real scenario. In this Section, firstly is illustrated the tools that have been used. Then, the results to validate the performance of the proposed consensus application are analysed. The tools used for the experiments are the following:

- Development kit case provided by Telit Wireless Solutions. This kit is made of five ZigBee radio boards that are based on the Texas Instruments CC2530 System on Chip with the Embedded Telit Z-One ZigBee-PRO Stack. The antennas are external dipoles characterized by an omnidirectional pattern.
- The software used to inspect the packet content is Wireshark. To analyze the performance of the network from the Wireshark output and to conduct network discovery and commissioning, a specific tool named SRManager Tool has been developed by Telit Wireless Solution in collaboration with our lab. In this experiment, this tool has been used to set up the consensus protocol.

During the experiments three devices are used, they communicated using the ZigBee standard on channel number 14 in the 2.4 GHz ISM frequency band. The type of communication is the gossip. To reduce communication overhead, the necessary information are inserted inside the overhead of the packets.

Fig. 5.5 shows the algorithm convergence. As in the simulation discussed in the previous Section, nodes converge at consensus. A good consensus has been reached after about 15 packets exchanged, corresponding to a mean of 5 update for node. From the error point of view, Fig. 5.6 shows that initially the error reaches peaks of $+80\%$ and -60%. Nevertheless, after 15

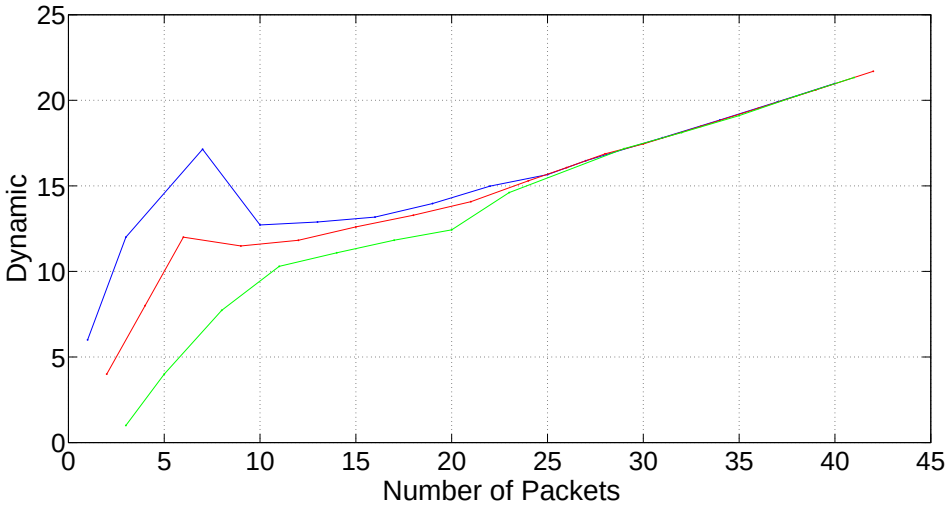


Figure 5.5: Dynamics evolution by real scenario.

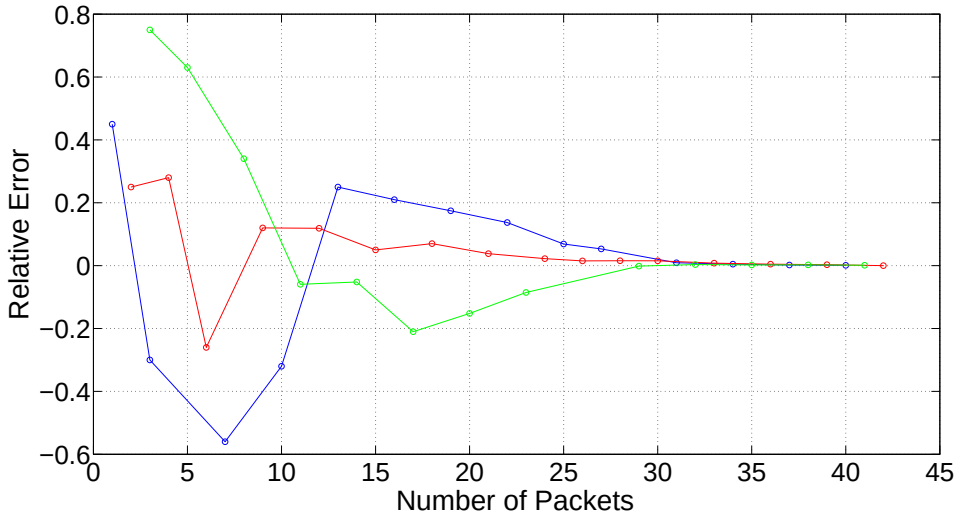


Figure 5.6: Error evolution by real scenario.

packets are transmitted, this value decreases by $\pm 20\%$, and eventually a very low error value of about $\pm 3\%$ is obtained.

5.1.3 Variance in the total task frequency

To compare the impact of the three different approaches described in Section 3.3, the variance in the total frequency $f_i^c(t) = \sum_{k=1}^{N_i} f_{i,k}(t)$ over all the network nodes is evaluated. Simulations run with 50 nodes with 5 *task groups* at different percentages of group overlapping (0%, 20%, 30%, 50%) and number of nodes (2, 4, 6, 10) in each *task group*. $X\%$ of overlapping means that $x\%$ of nodes in a group are also part of another group, on average. 100 runs per scenario are performed and computed the average value of the variance. The results are shown in Tables 5.1 - 5.3 for the three different approaches. Results for the first approach, i.e. single task - single frequency, are shown in Table 5.1. It can be observed as the variance increases as the group overlapping increases. This is due to the fact that in this case there will be nodes involved in more than one task without any inter-task coordination. These nodes will be characterized by a total frequency given by the sum of frequencies related to different tasks, and then higher than that of single-task nodes. Differently, the increase in the number of nodes in a *task group* has not any effect. The motivation is that the variance depends on the number of nodes that perform more than a single task and this does not depend on the size of each group.

		Overlapping			
		0%	**20%**	**30%**	**50%**
Nodes in a group	2	0.065	0.115	0.261	0.654
	4	0.073	0.131	0.219	0.711
	6	0.071	0.118	0.241	0.636
	10	0.069	0.126	0.252	0.688

Table 5.1: Evaluation of variance in single task single frequency scenario

As to the second approach (Table 5.2, single task - total frequency), the consensus protocol is used to find an agreement on the total frequency. For this reason, the variance in general is lower than the case of the previous approach, and this was expected. The difference is that this time the nodes that are involved in different tasks select frequencies for each tasks that are lower than those adopted by single task nodes so that the total frequency is invariant with respect to the number of tasks. This improvement is however not that significant as the agreement achieved by nodes in different groups can converge towards different frequencies. A more significant improvement is achieved with the third algorithm, i.e. entire network, as show in Table 5.3. Recall that this time the execution of a new instance of the protocol triggers the execution of a past instance where the nodes had already achieved the consensus. This further execution is

intended to achieve similar frequencies among different *task groups*. As for the single task - single frequency, these performance of these two algorithms are not affected by the size of the *task groups*.

		Overlapping			
		0%	**20%**	**30%**	**50%**
Nodes in a group	2	0.065	0.063	0.058	0.045
	4	0.073	0.065	0.061	0.049
	6	0.071	0.061	0.055	0.043
	10	0.069	0.060	0.054	0.046

Table 5.2: Evaluation of variance in single task total frequency scenario

		Overlapping			
		0%	**20%**	**30%**	**50%**
Nodes in a group	2	0.065	0.041	0.030	0.021
	4	0.073	0.050	0.036	0.019
	6	0.071	0.046	0.028	0.015
	10	0.069	0.040	0.023	0.016

Table 5.3: Evaluation of variance in entire network scenario

From the point of view of the distribution of the burden of executing the tasks in the considered IoT scenario, the entire network approach is the one that allows for achieving the best performance. However, it comes at the expenses of an increase in the energy consumed for the execution of the consensus as the consensus in a *task group* triggers the execution of the algorithm in groups that already reached an agreement.

5.2 Protocol for preserving Lifetime and QoS

The performance analysis focuses on two case studies: the first one is a simulated scenario; the second one is a real scenario. To better understand the problem and to simplify the analysis a scenario where the residual energy decreases very slowly than the convergence of protocol is considered, so it's possible to consider all terms as time independent.

5.2.1 Simulation Scenario

In this case study Matlab software is used to implement a framework to simulate the protocol using broadcast communications among the nodes. The network topology has been created following a random geometric distribution and transmissions on the network are asynchronous.

The broadcast communication entails that if i sends a packet, this is received by all neighbours, which update their values. The simulation was run on 20 nodes (i.e. $N_k = 20$) in a random topology. Nodes values are initialized as: $E_i^{res}(t)$, $E_{i,k}$, $f_{i,k}(t)$ with random values, the QoS request F_k has a random value, too. It's assumed that each node transmit 500 packets. By the simulation the intention is to study the performance of the protocol in terms of convergence speed and error, considering a broadcast communication among nodes. Fig. 5.7 shows the protocol lifetime convergence. Yellow points show the initial condition of network nodes. After 20 packets transmitted on the network, each node has corrected its task frequency and tends to the convergence (red points). Green points show the state of the nodes after 80 iterations. In the end, white points show the final state of the node, after 500 iterations. At the end of the simulation the protocol can be considered converged. In fact, in Fig. 5.7 white points are very near the ideal position that is marked by the blue line. So by Fig. 5.7 is clear that as the number of exchanged packets increases, nodes reach a better consensus and, in this case, the same lifetime.

From the QoS point of view, the percentage error with respect to the QoS constraint F_k is analysed. Fig. 5.8 shows that the initial error is -25%, but before 50 packets are transmitted this value decreases by 10%, and after 50 iterations a very low error value of about 5% is obtained. So it's possible, after a first transition time, to consider also the QoS constraint satisfied.

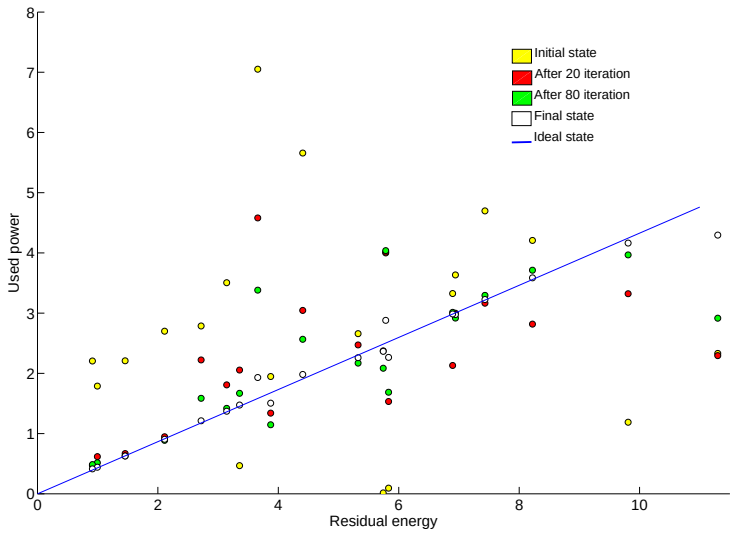


Figure 5.7: Lifetime consensus

5.2.2 Real Scenario

The last experiment consists in the study of the protocol performance in a real scenario. The analysis focuses on two case studies: the first one is a scenario with embedded systems; the second one is a scenario with android smartphones. For each test firstly is illustrated the tools used, then the results, to validate the performance of the proposed protocol, are analysed.

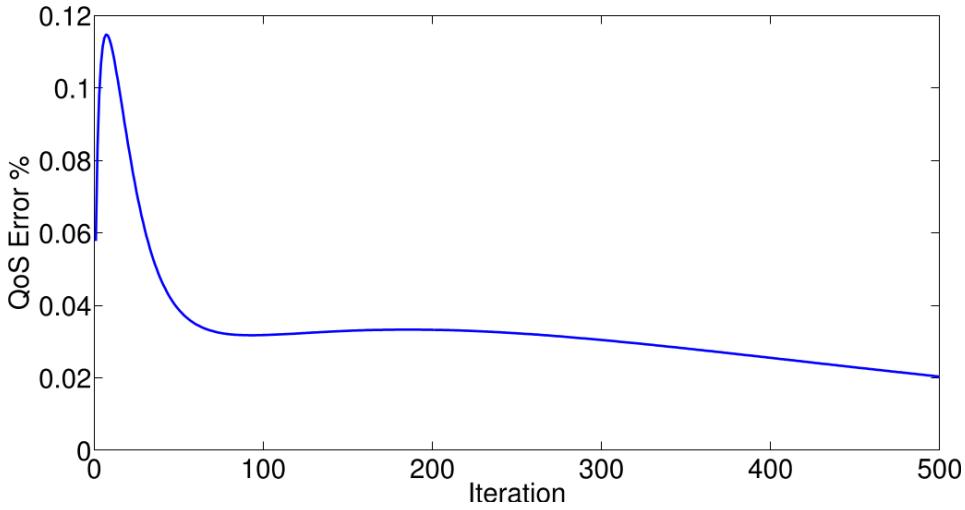


Figure 5.8: QoS error

Embedded systems

In this experiment the tools used are the following:

- Development kit case provided by Telit Wireless Solutions. This kit is made of ZigBee radio boards that are based on the CC2530 Chip with the Embedded Telit Z-One ZigBee-PRO Stack. The antennas are external dipoles characterized by an omnidirectional pattern.
- The software Wireshark is used to inspect the packet content. To analyse the performance of the network from the Wireshark output and to conduct network discovery and commissioning, a specific tool named SRManager Tool has been developed by Telit Wireless Solution in collaboration with our lab. In this experiment, this tool has been used to set up the consensus protocol experiments.

During the experiments three devices are used that communicated using the ZigBee standard on channel number 14 in the 2.4 GHz ISM frequency band and one device in sniffer mode to capture the packets on the network is used. The type of communication is the gossip. This modality entails that two nodes communicate to update their values, instead the broadcast when each node communicate with all neighbours. To reduce communication overhead, the necessary information inside are inserted on the overhead of the packets.

Fig. 5.9 shows the algorithm convergence in real scenario. As in the simulation discussed in the previous subsection, nodes converge at consensus. A good consensus has been reached after about 15 packets exchanged, corresponding to a mean of 5 update for node. From the error point of view the initially error reaches peaks of 60% and -30%. Nevertheless, after 15 packets are transmitted, this value decreases by $\pm 20\%$, and eventually a very low error value of about $\pm 3\%$ is obtained.

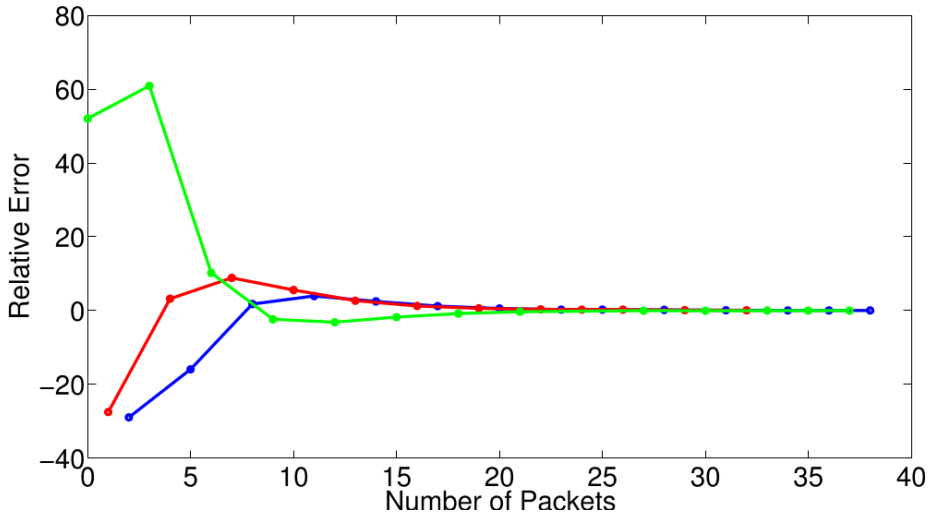


Figure 5.9: Convergence error in real scenario

Android systems

The last experiment consists in the study of the protocol performance in a real scenario with android systems. This scenario is typical example of crowdsensing community. At the first the area of interest is divided in cells i.e. example in Fig. 5.10), the devices in the same cell participate at the same task and so they are clustered on the same VO.

Figure 5.10: Example of area of interest.

During the experiment each device has a random mobility pattern, so the environment is very dynamic. Each device can change cell or can leave the area of interest or new device can

enter in the area. It's crucial the method used to update the position of each device and how the other devices discover the change. In the experiment are used three methods: centralized trigger, asynchronous trigger and decentralized trigger.

- Centralized trigger where a central server ping all devices each 60 sec and updates their position.
- Asynchronous trigger where each device send a packet to the server to signal if it changes the cell.
- Decentralized trigger where each device sends a packet at other devices, so there isn't a transit on server.

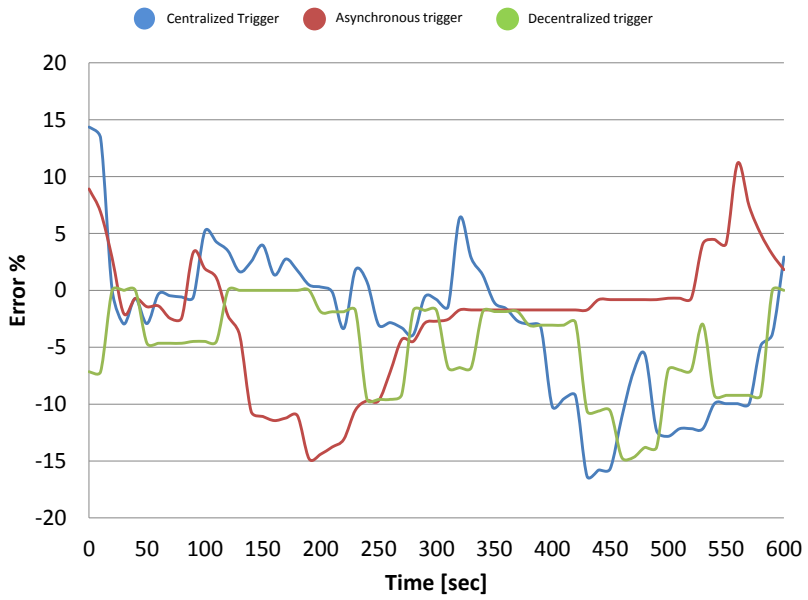


Figure 5.11: Convergence of three experimented methods.

Fig. 5.11 shows the algorithm convergence in the discussed scenario. The peaks of errors don't exceed the $\pm 20\%$. Centralized trigger solution converges slowly than the other two solutions. Asynchronous trigger solution has less peaks than the other solutions. The decentralized trigger solution has peaks quite attenuated than other solutions and the convergence is quick. From the error point of view in terms of mean and variance the figure is unlikely readable, so some results has been summarized in this table:

The table 5.4 shows that the case with centralized trigger has the greater value of variance and the mean error. In fact analysing Fig. 5.11 this method presents the slower convergence. Instead in the second and third cases the values of variance and mean error are similar and smaller than the first method. The results are similar because if in the asynchronous trigger

Method	Mean	Variance
Centralized trigger	5.27%	23.28%
Asynchronous trigger	4.52%	17.82%
Decentralized trigger	4.09%	16.28%

Table 5.4: Evaluation of error in three proposed methods.

solution has low peaks of error, the convergence is slower than the decentralized trigger solution. Whereas the last solution has not more high peaks, but presents a lot of fluctuations around the convergence value. After this comparative analysis, emerge that in all cases the mean error value has a good result, it is lower than $\pm 6\%$.

After the analysis of convergence, the energy aspect is evaluated. The three proposed methods have different impact on the energy consumption, because the number of transmission is different. The energy consumed to transmit packets is the prevalent contribute on the energy consumption, so in this analysis only the energy spent in transmission is evaluated. To simplify the analysis is assumed that all devices communicate with the same protocol and the energy consumed to transmit packets is proportional to the number of bit which form the packet.

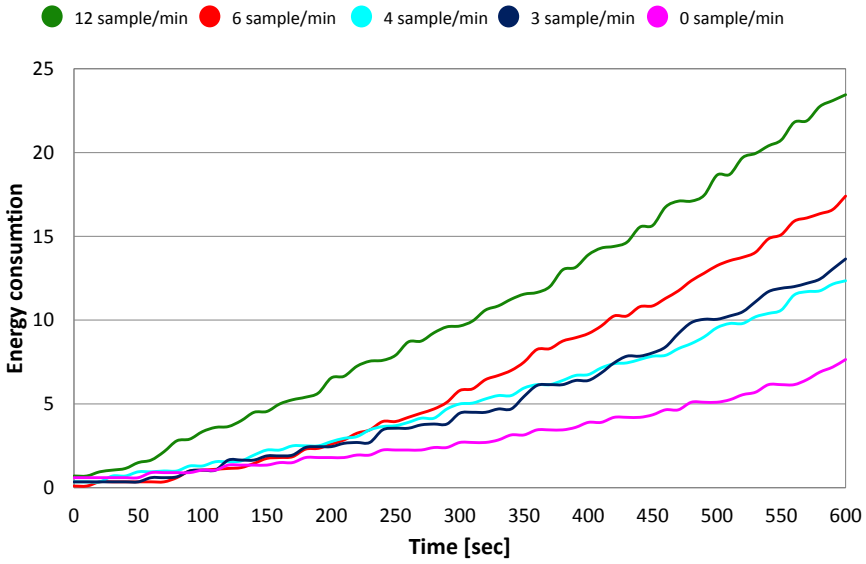


Figure 5.12: Analysis of energy consumption in centralized trigger solution.

Figs. 5.12, 5.13 and 5.14 show the trend of energy consumption for the three methods proposed in this thesis. In all case the energy consumption increase according to the number of sample required, this happened because some control information are packaged in the data packet to minimize the overhead. So it's clear that a high rate of sample required produces a

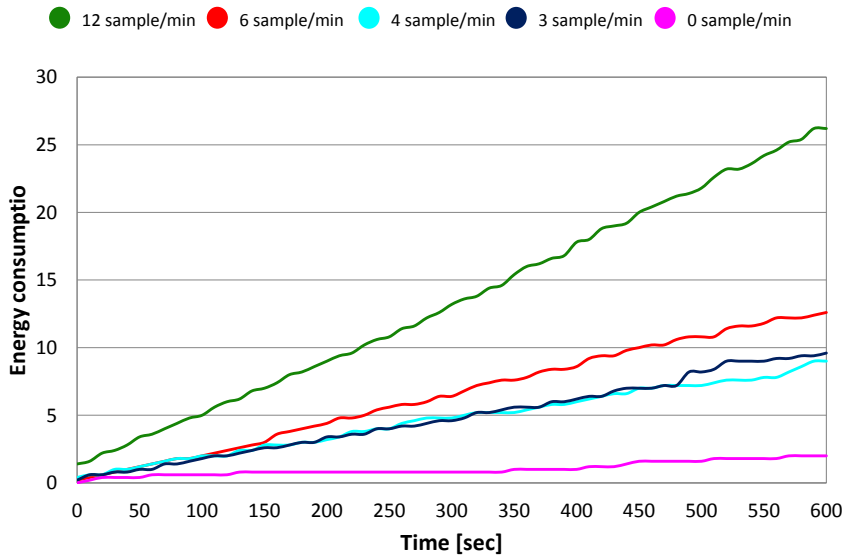


Figure 5.13: Analysis of energy consumption in asynchronous trigger solution.

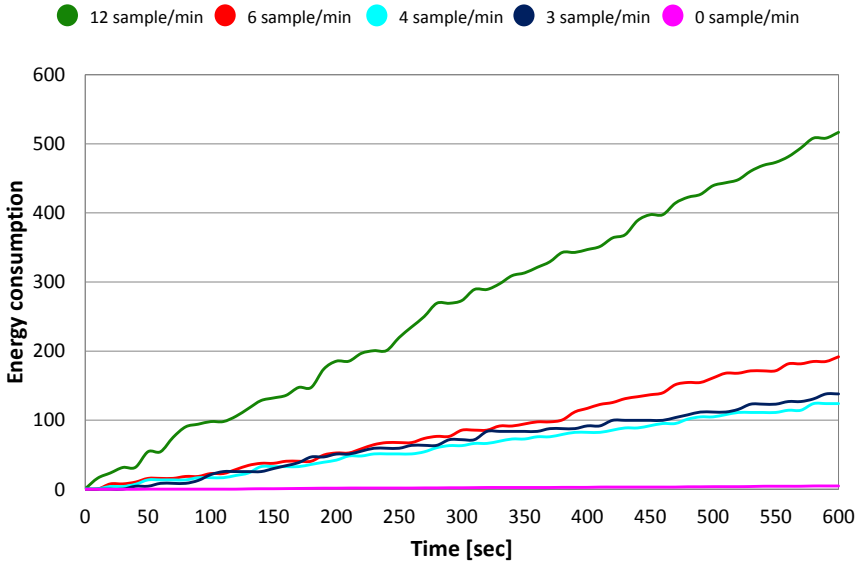


Figure 5.14: Analysis of energy consumption in decentralized trigger solution.

proportional control information. Analysing the Fig. 5.12, which shows the energy consumption in the first method, also the devices that are not involved in the measure (i.e. 0 sample required) consume energy for the control. This happened because the control is centralized and the server sends a request at all devices each 60 sec. The analysis of the second and the third methods are shown in Figs. 5.13 and 5.14. In these cases the energy consumed by devices which are not involved in measure is negligible, a bit consume is caused by the change of position of devices from active cell to inactive cell (i.e. 0 sample required). The most difference between the methods is the order of size of energy consumed. In fact in the third case each device send a packet control to all devices in the same cell, so the energy consumed dramatically increases.

The last analysis is a comparison between the mean error of convergence and the energy consumed to the control.

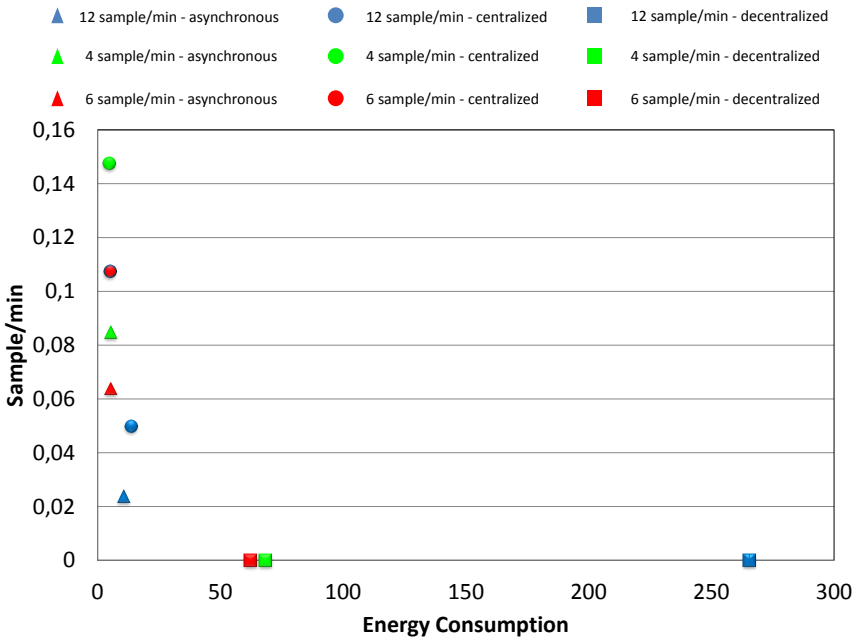


Figure 5.15: comparison between the mean error of convergence and the energy consumed to the control.

Fig. 5.15 shows the compared analysis for the three proposed methods. The third method has the high energy consumption, but it guarantees the best mean error. The first and the second method are comparable in terms of energy consumption, but the method totally centralized has an higher mean error than the asynchronous trigger.

Chapter 6

Conclusions and Future Works

In this thesis, the problem of task allocation in the Internet of Things has been studied. The typical IoT scenario where applications require the collaboration of different nodes has been analysed. Accordingly, a middleware solution able to support the task allocation in an IoT scenario have been defined. Algorithms proposed are intended to dynamically adapt the task allocation among nodes to obtain an improvement of performance in the network.

In the first chapter 1 the thesis starts with a summary on the state of the art of the main topic discussed during the rest of the document like the problem of the task allocation, the key issues in the IoT scenario and the main characteristics of typical opportunistic scenarios. At the end of chapter is discussed some IoT architecture proposed in literature and a taxonomy of IoT application.

In chapter 2 the Middleware layer functionalities for IoT task allocation are discussed. The importance to design a virtualization layer is introduced and the role of Virtual Object in this layer is discussed. In the second part of this chapter is proposed a reference middleware oriented to the problem of task allocation in the IoT. This layer is composed by two sublayers: the first is a semantic layer which guarantees the interoperability and the second contains the modules that achieve the management of resources.

Two protocols are presented in the chapters 3 and 4. Both protocols achieve a task allocation to reach a goal. In the first protocol the challenges faced in this chapter is the deployment of distributed applications in the IoT in terms of cooperation among objects, with the aim of distributing the burden of the execution of the application committed to the network, so that resources are adequately exploited. At the first the problem has been modelled and after a consensus algorithm is applied to reach the goal. At the end of chapter three different applications of the consensus protocol are presented. The chapter 4has faced the challenge of the deployment of distributed applications in the IoT in terms of cooperation among objects, with the aim to solve the problem of resource allocation and management preserving the required QoS. In the first

subsection has been presented the model that provide an agreement on task frequency among nodes, instead in the second subsection is presented the solution, that is totally distributed and based on a consensus algorithm.

Both proposed protocols are tested in simulated and real scenario. The chapter 5 presents the performance analysis of the two protocols presented in this thesis. At the first the protocol for an homogeneous resources consumption has been analysed. The performance has been evaluated considering the convergence of the algorithm within simulated and real scenarios and such as the distribution of resource when different application are applied. During the simulation, two type of communication paradigm are tested: broadcast and gossip. Instead in real scenario are used tiny embedded systems.

Performance of protocol for preserving lifetime and QoS are analysed in the same chapter. Also in this case the performance has been evaluated considering the convergence of the algorithm within simulated and two real scenarios. The first real scenario is composed by tiny embedded systems which use ZigBee protocol to communicate, the other scenario is composed by smartphones equipped with android operative system. In the last use case analysed also the impact of algorithm on the energy consumption is evaluated.

In light of the improvements presented throughout this work, the need for juice up the discussion of the importance of task allocation in the IoT scenario. The study done so far has led to the acquisition of the expertise required to widen the problem of task allocation from the only energy consumption point of view to a new viewpoint oriented to improve also the quality of information (QoI) without resources consumption. Future work will be focused on the study of new cooperation algorithms that will evaluate also the quality of acquired data in order to achieve optimal performance in terms of QoI. Last, but not least an improvement of QoI can bring an improvement of the satisfaction of users who access at information, so it will be open a new topic of Quality of Experience (QoE) in the IoT scenario.

Bibliography

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010.

[2] C.-M. Huang, K.-c. Lan, and C.-Z. Tsai, "A survey of opportunistic networks," in *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on*. IEEE, 2008, pp. 1672–1677.

[3] IoT-A, "Internet of Things – Architecture." [Online]. Available: http://www.iot-a.eu/public/public-documents/d1.2/view

[4] T. Teixeira, S. Hachem, V. Issarny, and N. Georgantas, "Service oriented middleware for the internet of things: a perspective," in *Towards a Service-Based Internet*, 2011, pp. 220–229.

[5] Y. Yu and V. K. Prasanna, "Energy-balanced task allocation for collaborative processing in wireless sensor networks," *Mobile Networks and Applications*, vol. 10, pp. 115–131, 2005.

[6] N. Edalat, W. Xiao, C. Tham, E. Keikha, and L. Ong, "A price-based adaptive task allocation for wireless sensor network," in *Mobile Adhoc and Sensor Systems, 2009. MASS'09. IEEE 6th International Conference on*, 2009, pp. 888–893.

[7] V. Pilloni and L. Atzori, "Deployment of distributed applications in wireless sensor networks," *Sensors*, vol. 11, no. 8, pp. 7395–7419, 2011.

[8] Y. Jin, J. Jin, A. Gluhak, K. Moessner, and M. Palaniswami, "An intelligent task allocation scheme for multihop wireless networks," *Parallel and Distributed Systems, IEEE Transactions on*, pp. 444–451, 2012.

[9] J. Zhu, J. Li, and H. Gao, "Tasks allocation for real-time applications in heterogeneous sensor networks for energy minimization," in *Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, vol. 2, 2007, pp. 20–25.

[10] S. Abdelhak, C. Gurram, S. Ghosh, and M. Bayoumi, "Energy-balancing task allocation on wireless sensor networks for extending the lifetime," in *Circuits and Systems (MWSCAS), 2010 53rd IEEE International Midwest Symposium on*, 2010, pp. 781 –784.

[11] V. Pilloni, M. Franceschelli, L. Atzori, and A. Giua, "A decentralized lifetime maximization algorithm for distributed applications in wireless sensor networks," in *Communications (ICC), 2012 IEEE International Conference on*, 2012, pp. 1392–1397.

[12] Y. Shen and H. Ju, "Energy-efficient task assignment based on entropy theory and particle swarm optimization algorithm for wireless sensor networks," in *Green Computing and Communications (GreenCom), 2011 IEEE/ACM International Conference on*, 2011, pp. 120 –123.

[13] W. Wang, S. De, R. Toenjes, E. Reetz, and K. Moessner, "A comprehensive ontology for knowledge representation in the internet of things," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, 2012, pp. 1793–1798.

[14] S. De, T. Elsaleh, P. Barnaghi, and S. Meissner, "An internet of things platform for real-world and digital objects," *Scalable Computing: Practice and Experience*, vol. 13, no. 1, 2012.

[15] D. Guinard, V. Trifa, F. Mattern, and E. Wilde, "From the internet of things to the web of things: Resource-oriented architecture and best practices," in *Architecting the Internet of Things*. Springer, 2011, pp. 97–129.

[16] B. Silverajan and J. Harju, "Developing network software and communications protocols towards the internet of things," in *Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE*. ACM, 2009, p. 9.

[17] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[18] L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," in *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007, pp. 2289–2294.

[19] L. Schenato and F. Fiorentin, "Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, 2011.

[20] Z. Li, Z. Duan, G. Chen, and L. Huang, "Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 57, no. 1, pp. 213–224, 2010.

[21] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401–420, 2006.

[22] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *Automatic Control, IEEE Transactions on*, vol. 51, no. 8, pp. 1289–1298, 2006.

[23] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*. IEEE, 2005, pp. 63–70.

[24] I. Howitt and J. Wang, "Energy balanced chain in distributed sensor networks," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 3, 2004, pp. 1721–1726.

[25] D. Wang, B. Xie, and D. Agrawal, "Coverage and lifetime optimization of wireless sensor networks with gaussian distribution," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 12, pp. 1444–1458, 2008.

[26] S. Sengupta, S. Das, M. Nasir, and B. Panigrahi, "Multi-objective node deployment in wsns: In search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity," *Engineering Applications of Artificial Intelligence*, 2012.

[27] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000, p. 10pp.

[28] D. Wei, Y. Jin, S. Vural, K. Moessner, and R. Tafazolli, "An energy-efficient clustering solution for wireless sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 11, pp. 3973–3983, 2011.

[29] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, 2003, pp. 1713–1723.

[30] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, "A survey on facilities for experimental internet of things research," *Communications Magazine, IEEE*, vol. 49, no. 11, pp. 58–67, 2011.

[31] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's intranet of things to a future internet of things: a wireless-and mobility-related view," *Wireless Communications, IEEE*, vol. 17, no. 6, pp. 44–51, 2010.

[32] J. Zheng and M. J. Lee, "Will ieee 802.15. 4 make ubiquitous networking a reality?: a discussion on a potential low power, low bit rate standard," *Communications Magazine, IEEE*, vol. 42, no. 6, pp. 140–146, 2004.

[33] D. Saha and A. Mukherjee, "Pervasive computing: a paradigm for the 21st century," *Computer*, vol. 36, no. 3, pp. 25–31, 2003.

[34] G. Colistra, V. Pilloni, and L. Atzori, "The problem of task allocation in the internet of things and the consensus-based approach," *Computer Networks*, vol. 73, pp. 98–111, 2014.

[35] K.-D. Moon, Y.-H. Lee, C.-E. Lee, and Y.-S. Son, "Design of a universal middleware bridge for device interoperability in heterogeneous home network middleware," *Consumer Electronics, IEEE Transactions on*, vol. 51, no. 1, pp. 314–318, 2005.

[36] F. Mattern and C. Floerkemeier, "From the internet of computers to the internet of things," in *From active data management to event-based systems and more*. Springer, 2010, pp. 242–259.

[37] D. J. Cook and S. K. Das, "Pervasive computing at scale: Transforming the state of the art," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 22–35, 2012.

[38] S. Li, L. Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and internet of things," 2011.

[39] N. Vlajic and D. Xia, "Wireless sensor networks: to cluster or not to cluster?" in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*. IEEE Computer Society, 2006, pp. 258–268.

[40] J. Postel, "Internet official protocol standards," 1998.

[41] G. Goth, "The end of ipv4 is nearly here &# x2014; really," *Internet Computing, IEEE*, vol. 16, no. 2, pp. 7–11, 2012.

[42] S. Weber and L. Cheng, "A survey of anycast in ipv6 networks," *Communications Magazine, IEEE*, vol. 42, no. 1, pp. 127–132, 2004.

[43] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the internet of things: A survey," in *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*. IEEE, 2011, pp. 1–6.

[44] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "Mqtt-sa publish/subscribe protocol for wireless sensor networks," in *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*. IEEE, 2008, pp. 791–798.

[45] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, "Constrained application protocol (coap), draft-ietf-core-coap-13," *Orlando: The Internet Engineering Task Force–IETF, Dec*, 2012.

[46] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks," in *Mobile Data Management, 2007 International Conference on*. IEEE, 2007, pp. 198–205.

[47] M. Harchol-Balter, T. Leighton, and D. Lewin, "Resource discovery in distributed networks," in *Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing*. ACM, 1999, pp. 229–237.

[48] W. K. Edwards, "Discovery systems in ubiquitous computing," *Pervasive Computing, IEEE*, vol. 5, no. 2, pp. 70–77, 2006.

[49] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *Services Computing, IEEE Transactions on*, vol. 3, no. 3, pp. 223–235, 2010.

[50] G. Fortino, M. Lackovic, W. Russo, and P. Trunfio, "A discovery service for smart objects over an agent-based middleware," in *Internet and Distributed Computing Systems*. Springer, 2013, pp. 281–293.

[51] S. Evdokimov, B. Fabian, S. Kunz, and N. Schoenemann, "Comparison of discovery service architectures for the internet of things," in *Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 237–244.

[52] I. F. Akyildiz, J. Xie, and S. Mohanty, "A survey of mobility management in next-generation all-ip-based wireless systems," *Wireless Communications, IEEE*, vol. 11, no. 4, pp. 16–28, 2004.

[53] D. Saha, A. Mukherjee, I. S. Misra, and M. Chakraborty, "Mobility support in ip: a survey of related protocols," *Network, IEEE*, vol. 18, no. 6, pp. 34–40, 2004.

[54] S. Hong, D. Kim, M. Ha, S. Bae, S. J. Park, W. Jung, and J.-E. Kim, "Snail: an ip-based wireless sensor network approach to the internet of things," *Wireless Communications, IEEE*, vol. 17, no. 6, pp. 34–42, 2010.

[55] C. Perkins, "Autoconfiguration plug & play internet," *Internet Computing, IEEE*, vol. 3, no. 4, pp. 42–44, 1999.

[56] B. A. Miller, T. Nixon, C. Tai, and M. D. Wood, "Home networking with universal plug and play," *Communications Magazine, IEEE*, vol. 39, no. 12, pp. 104–109, 2001.

[57] R. H. Weber, "Internet of things–new security and privacy challenges," *Computer Law & Security Review*, vol. 26, no. 1, pp. 23–30, 2010.

[58] B. Fabian and O. Gunther, "Distributed ons and its impact on privacy," in *Communications, 2007. ICC'07. IEEE International Conference on*. IEEE, 2007, pp. 1223–1228.

[59] E. Crawley, H. Sandick, R. Nair, and B. Rajagopalan, "A framework for qos-based routing in the internet," 1998.

[60] K. Wu, J. Harms *et al.*, "Qos support in mobile ad hoc networks," *Crossing Boundaries-the GSA Journal of University of Alberta*, vol. 1, no. 1, pp. 92–106, 2001.

[61] D. Chen and P. K. Varshney, "Qos support in wireless sensor networks: A survey." in *International Conference on Wireless Networks*, vol. 233, 2004, pp. 1–7.

[62] H. El-Sayed, A. Mellouk, L. George, and S. Zeadally, "Quality of service models for heterogeneous networks: overview and challenges," *annals of telecommunications-annales des télécommunications*, vol. 63, no. 11-12, pp. 639–668, 2008.

[63] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder, "Management of resource constrained devices in the internet of things," *Communications Magazine, IEEE*, vol. 50, no. 12, pp. 144–149, 2012.

[64] G. Colistra, V. Pilloni, and L. Atzori, "Task allocation in group of nodes in the iot: a consensus approach," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3848–3853.

[65] ——, "Objects that agree on task frequency in the iot: A lifetime-oriented consensus based approach," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 383–387.

[66] M. Kovatsch, S. Mayer, and B. Ostermaier, "Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*. IEEE, 2012, pp. 751–756.

[67] K. Hwang, J. Dongarra, and G. C. Fox, *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan Kaufmann, 2013.

[68] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[69] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 169–186.

[70] J.-Z. Sun, "Mobile ad hoc networking: an essential technology for pervasive computing," in *Info-tech and Info-net, 2001. Proceedings. ICII 2001-Beijing. 2001 International Conferences on*, vol. 3. IEEE, 2001, pp. 316–321.

[71] C. Boldrini, M. Conti, F. Delmastro, and A. Passarella, "Context-and social-aware middleware for opportunistic networks," *Journal of Network and Computer Applications*, vol. 33, no. 5, pp. 525–541, 2010.

[72] A. Bokar, M. Bozyigit, and C. Sener, "Scalable energy-aware dynamic task allocation," in *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*. IEEE, 2009, pp. 371–376.

[73] C. Comito, D. Falcone, D. Talia, and P. Trunfio, "Energy efficient task allocation over mobile networks," in *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*. IEEE, 2011, pp. 380–387.

[74] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "Opportunistic iot: exploring the social side of the internet of things," in *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*. IEEE, 2012, pp. 925–929.

[75] A. Kapadia, D. Kotz, and N. Triandopoulos, "Opportunistic sensing: Security challenges for the new paradigm," in *Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International*. IEEE, 2009, pp. 1–10.

[76] D. Zhang, B. Guo, and Z. Yu, "The emergence of social and community intelligence," *Computer*, vol. 44, no. 7, pp. 21–28, 2011.

[77] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "Bikenet: A mobile sensing system for cyclist experience mapping," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 1, p. 6, 2009.

[78] H. Lu, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Bubble-sensing: Binding sensing tasks to the physical world," *Pervasive and Mobile Computing*, vol. 6, no. 1, pp. 58–71, 2010.

[79] M. Shin, P. Tsang, D. Kotz, and C. Cornelius, "Deamon: Energy-efficient sensor monitoring," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09. 6th Annual IEEE Communications Society Conference on*. IEEE, 2009, pp. 1–9.

[80] P. Kostelnik, M. Sarnovsk, and K. Furdik, "The semantic middleware for networked embedded systems applied in the internet of things and services domain," *Scalable Computing: Practice and Experience*, vol. 12, no. 3, 2011.

[81] L. M. S. De Souza, P. Spiess, D. Guinard, M. Köhler, S. Karnouskos, and D. Savio, "Socrades: A web service based shop floor integration infrastructure," in *The internet of things*. Springer, 2008, pp. 50–67.

[82] D. Massaguer, S. Mehrotra, R. Vaisenberg, and N. Venkatasubramanian, "Satware: A semantic approach for building sentient spaces," in *Distributed Video Sensor Networks*. Springer, 2011, pp. 389–402.

[83] K. Aberer, M. Hauswirth, and A. Salehi, "The global sensor networks middleware for efficient and flexible deployment and interconnection of sensor networks," *Ecole Polytechnique Fdrale de Lausanne (EPFL), Tech. Rep. LSIR-REPORT-2006-006*, 2006.

[84] N. Kefalakis, N. Leontiadis, J. Soldatos, and D. Donsez, "Middleware building blocks for architecting rfid systems," in *Mobile Lightweight Wireless Systems*. Springer, 2009, pp. 325–336.

[85] R. J. C. BENITO, D. G. MÁRQUEZ, P. P. TRON, R. R. CASTRO, N. S. MARTÍN, and J. L. S. MARTÍN, "Smepp: A secure middleware for embedded p2p," *Proceedings of ICT-MobileSummit*, vol. 9, 2009.

[86] V. Terziyan, O. Kaykova, and D. Zhovtobryukh, "Ubiroad: Semantic middleware for context-aware smart road environments," in *Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference on*. IEEE, 2010, pp. 295–302.

[87] S. Haller, "The things in the internet of things," *Poster at the (IoT 2010). Tokyo, Japan, November*, vol. 5, p. 26, 2010.

[88] S. E. Chen, "Quicktime vr: An image-based approach to virtual environment navigation," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 29–38.

[89] M. Zyda, "From visual simulation to virtual reality to games," *Computer*, vol. 38, no. 9, pp. 25–32, Sept 2005.

[90] CONVERGENCE, "Convergence," 2010. [Online]. Available: http://www.ict-convergence.eu/

[91] K. Römer, T. Schoch, F. Mattern, and T. Dübendorfer, "Smart identification frameworks for ubiquitous computing applications," *Wireless Networks*, vol. 10, no. 6, pp. 689–700, 2004.

[92] F. Carrez, "Td 3.2–reference architecture. sensei, public deliverable d. 3.2, 2009."

[93] COMPOSE, "Collaborative open market to place objects at your service," 2012. [Online]. Available: http://www.compose-project.eu/

[94] iCore, "Empowering iot through cognitive technologies," 2011. [Online]. Available: http://www.iot-icore.eu/

[95] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, 2013.

[96] K. Malhi, S. C. Mukhopadhyay, J. Schnepper, M. Haefke, and H. Ewald, "A zigbee-based wearable physiological parameters monitoring system," *Sensors Journal, IEEE*, vol. 12, no. 3, pp. 423–430, 2012.

[97] H. Viswanathan, B. Chen, and D. Pompili, "Research challenges in computation, communication, and context awareness for ubiquitous healthcare," *Communications Magazine, IEEE*, vol. 50, no. 5, pp. 92–99, 2012.

[98] F. Viani, F. Robol, A. Polo, P. Rocca, G. Oliveri, and A. Massa, "Wireless architectures for heterogeneous sensing in smart home applications: Concepts and real implementation," *Proceedings of the IEEE*, vol. 101, no. 11, pp. 2381–2396, 2013.

[99] R. Costa, D. Carneiro, P. Novais, L. Lima, J. Machado, A. Marques, and J. Neves, "Ambient assisted living," in *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*. Springer, 2009, pp. 86–94.

[100] A. Mahmood, H. Fakhar, S. Ahmed, and N. Javaid, "Home energy management in smart grid," *arXiv preprint arXiv:1311.5385*, 2013.

[101] L. Li, W. Xie, Z. He, X. Xu, C. Chen, and X. Cui, "Design of smart home control system based on zigbee and embedded web technology," in *Artificial Intelligence and Computational Intelligence*. Springer, 2012, pp. 67–74.

[102] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization," *Computer Networks*, 2012.

[103] S. Karpischek, F. Michahelles, F. Resatsch, and E. Fleisch, "Mobile sales assistant-an nfc-based product information system for retailers," in *Near Field Communication, 2009. NFC'09. First International Workshop on*. IEEE, 2009, pp. 20–23.

[104] A. Resch and T. Blecker, "Smart logistics–a literature review," *Pioneering Supply Chain Design: A Comprehensive Insight Into Emerging Trends, Technologies and Applications*, vol. 10, p. 91, 2012.

[105] C. Hipp, T. Sellner, J. Bierkandt, and P. Holtewert, "Smart factory: System logic of the project epik," in *SMART 2012, The First International Conference on Smart Systems, Devices and Technologies*, 2012, pp. 105–111.

[106] Z. Xiong, H. Sheng, W. Rong, and D. E. Cooper, "Intelligent transportation systems for smart cities: a progress review," *Science China Information Sciences*, vol. 55, no. 12, pp. 2908–2914, 2012.

[107] S. Djahel, M. Salehie, I. Tal, and P. Jamshidi, "Adaptive traffic management for secure and efficient emergency services in smart cities," 2013.

[108] J. Ye, B. Chen, Q. Liu, and Y. Fang, "A precision agriculture management system based on internet of things and webgis," in *Geoinformatics (GEOINFORMATICS), 2013 21st International Conference on*. IEEE, 2013, pp. 1–5.

[109] N. Bui, A. P. Castellani, P. Casari, and M. Zorzi, "The internet of energy: a web-enabled smart grid system," *Network, IEEE*, vol. 26, no. 4, pp. 39–45, 2012.

[110] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2m: From mobile to embedded internet," *Communications Magazine, IEEE*, vol. 49, no. 4, pp. 36–43, 2011.

[111] M. Vecchio, S. Sasidharan, F. Marcelloni, and R. Giaffreda, "Reconfiguration of environmental data compression parameters through cognitive iot technologies," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on*. IEEE, 2013, pp. 141–146.

[112] J. Pascual Espada, Ó. Sanjuán Martínez, G. Pelayo, C. Bustelo, and J. M. Cueva Lovelle, "Virtual objects on the internet of things," *IJIMAI*, vol. 1, no. 4, pp. 23–29, 2011.

[113] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog *et al.*, "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, 2012.

[114] B. Vatant and M. Wick, "Geonames ontology (2006)," *Online at http://www. geonames. org/ontology*.

[115] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33 – 46, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731506001808

List of Publications Related to the Thesis

- G Colistra, V Pilloni, L Atzori. ”The problem of task allocation in the Internet of Things and the consensus-based approach,” Computer Networks, 2014, v 73, pp.98-111
- R Cherchi, G Colistra, V Pilloni, L Atzori. ”Energy consumption management in Smart Homes: An M-Bus communication system,” International Conference on Telecommunications and Multimedia (TEMU), 2014, pp.80-85
- G Colistra, V Pilloni, L Atzori. ”Task allocation in group of nodes in the IoT: a Consensus Approach,” IEEE International Conference on Communications (ICC), 2014, pp. 3848-3853
- G Colistra, V Pilloni, L Atzori. ”Objects that agree on task frequency in the IoT: A lifetime-oriented consensus based approach,” IEEE World Forum on Internet of Things (WF-IoT), 2014, pp. 383-387