Multi-constrained QoS Routing and Energy Optimization for Wireless Sensor Networks

Evangelia Tsiontsiou

▶ **To cite this version:**

Evangelia Tsiontsiou. Multi-constrained QoS Routing and Energy Optimization for Wireless Sensor Networks. Networking and Internet Architecture [cs.NI]. Université de Lorraine, 2017. English. NNT : 2017LORR0340 . tel-01735239

HAL Id: tel-01735239

https://theses.hal.science/tel-01735239

Submitted on 15 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Multi-constrained QoS Routing and Energy Optimization for Wireless Sensor Networks

THÈSE

présentée et soutenue publiquement le 15 Decembre 2017

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Evangelia TSIONTSIOU

Composition du jury

Rapporteurs :	Karine Deschinkel	MCF (HDR)	- Université de Franche-Comté
	Pascal Lorenz	Professeur	- Université de Haute Alsace
Examinateurs :	Thomas Noël	Professeur	- Université de Strasbourg
	Ye-Qiong Song	Professeur	- Université de Lorraine
	Bernardetta Addis	MCF	- Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Remerciements

In one of the most famous quotes, Alexander the Great expresses his recognition to his mentor Aristotle: *Στους γονείς οφείλομεν το ζην, στους δε διδασκάλους το ευ ζην* (To our parents we owe living, to our teachers we owe living well).

I would like to express my most sincere gratitude to my supervisors, Ye-Qiong Song and Bernardetta Addis, whose guidance and contributions have been essential to my work. They suggested interesting research topics and were always eager to spend their time listening and providing help in every step of my work. I thank them for their constant encouragement, patience and support.

I would also like to thank the members of the jury who devoted their time to studying my work: Karine Deschinkel and Pascal Lorenz who acted as referees and Thomas Noël as an examiner.

The research presented in this manuscript was carried out within the Madynes team at Loria. One of the key factors behind this work was the friendly atmosphere at the institute where I had the chance to meet great people. I would like to express my gratitude to all the members of Madynes and Optimist team and also the rest of my friends who were working in Loria and made the day-to-day working conditions very pleasant: Iordan Iordanov, Daishi Kondo, Zia Alborzi, Meihui Gao, Younes Rahmani, Tayeb Oulad Kouider, Magda Krzaczkowska, Jirka Marsik, Kevin Rousselle, Elian Aubry and Anthea Mayzaud.

I would also like to express my special thanks to all my friends outside the laboratory who made my experience abroad very unique: Marianna Grigoriou, Vassilis Vassiliadis, Christos Siskos, Vitalis Ntombrougidis, Stella Tsomora, Paolo Pigato, Eric Biagioli and Eleonora Carocci. A special mention to Dimitris Voliotis, Dimitris Tsabalas and Bernardette Tsabalas who were my second family in Nancy.

I would like to thank my lovely friends in Greece for their constant support in distance.

Τέλος, ένα πολύ μεγάλο και θερμό ευχαριστώ οφείλω στην οικογένεια μου για την αμέριστη αγάπη και συμπαράστασή τους: τους γονείς μου Αλεξάνδρα και Δημήτρη, τα αδέρφια μου Σμαράγδα, Δάφνη και Στέλιο.

To Stella, Dimitris and Konstantinos

Contents

List of Figures

List of Tables

Résumé

La cadre de cette thèse est la conception de protocoles de routage pour les réseaux de capteurs. Les problèmes de recherche du routage de données dans un réseau multi-sauts sont d'une part l'optimisation de l'énergie et d'autre part le routage sous contraintes de la qualité de service (QoS) multicritères (e.g., énergie, fiabilité, délai , ...). Cette thèse apporte deux contributions par rapport à l'état de l'art: une optimisation d'un protocole de routage probabiliste pour l'équilibre de l'usage d'énergie et un protocole de routage capable de prendre en compte simultanément des métriques de QoS multiples.

En effet, pour équilibrer la consommation de l'énergie du routage lorsque des chemins multiples existent, les protocoles de routage probabiliste existants affectent une probabilité de choix à chaque chemin, soit de façon empirique, soit proportionnelle au niveau de l'énergie disponible du chemin. Nous ne savions pas quelles sont les probabilités optimales qui permettent d'avoir la durée de vie maximale du réseau. Cette thèse a permit d'établir ces probabilités optimales à l'aide de la modélisation sous forme d'un problème d'optimisation linéaire.

Quant au problème du routage multicritères, bien que des métriques multiples soient définies par RPL (un standard d'IETF), les protocoles existants choisissent la route soit sur une métrique, soit sur une fonction de coût combinant plusieurs (qui introduit par conséquent un biais de pondération), mais jamais plusieurs simultanément. Dans cette thèse, nous avons d'abord évalué numériquement les performances de l'approche « operator calculus algebra » introduit par R. Schott et S. Staples qui définit un algorithme efficace permettant de trouver tous les chemins satisfaisant les contraintes multiples dans un graphe, puis dérivé une version distribuée sur laquelle nous avons conçu un protocole de routage multi-métriques.

Ces deux contributions ont été implémentées dans l'environnement Contiki et émulées/simulées sous Cooja (un logiciel permettant de simuler des protocoles des réseaux de capteurs).

Mots-clés: Routage QoS, Rèseau de capteurs sans fil, Optimisation de l'Energie, Durèe de vie du rèseau

Abstract

In this thesis, we focus on routing protocols for Wireless Sensor Networks (WSNs). The main research problems in the domain of routing data packets in a multi-hop network are the optimisation of the energy and the routing under multi-criteria QoS constraints (e.g., energy, reliability, delay, ...). To address these problems, this dissertation proposes two contributions. Firstly, an optimal probabilistic routing protocol which balances the usage of energy and secondly, a routing protocol which is able to simultaneously take into account multiple QoS metrics.

In fact, for balancing the energy consumption between the multiple existing links, the existing probablistic routing protocols assign a probability to each link, either in an empirical way or depending on proportional energy level of the path. We did not know what are the optimal probabilities which lead to the optimal network lifetime. Our first contribution proposes optimal probabilities by modeling and solving a linear programming problem.

As for the multi-constrainted QoS routing problem, multiple metrics are defined by RPL (a standard of IETF) but the existing protocols chose paths either according to only one metric or using a single aggregated function with multiple metrics, but never all the metrics simultaneously. In this dissertation, we first evaluate the performance of the « operator calculus algebra » introduced by R. Schott and S. Staples which defines an efficient algorithm allowing to find all the paths which satisfy the multiple constraints in a graph, and secondly we proposed a distributed version of this algorithm based on which a routing protocol has been designed.

Both contributions are implemented in Contiki environment and simulated/emulated under Cooja (a software designed for simulating protocols of WSNs.)

Keywords: QoS Routing, Wireless Sensor Networks, Energy Optimization, Network Lifetime

Chapter 1

General Introduction

Sommaire

1.1 Background and motivations

The rapid development of communication and information technology has led to a hyper connected society in which everything is connected to mobile devices and the Internet and strong communication is present among different services. The core component of this hyper connected society is called Internet of Things (IoT). IoT is enabled by the latest developments in radio-frequency identification (RFID), smart sensors, communication technologies, and Internet protocols. The basic premise is to have smart sensors collaborate directly without human involvement to deliver a new class of applications. In the coming years, the IoT is expected to bridge diverse technologies to enable new applications by connecting physical objects together in support of intelligent decision making.

Wireless Sensor Networks (WSNs) is one of the enabling technologies for IoT. WSNs have many applications from air-quality monitoring to home automation, security surveillance, health monitoring, etc. In this thesis we mainly focus on WSN protocol design for its use in smart home/building applications for healthcare.

Nowadays, the aging population is constantly increasing so that monitoring the behavior of the elderly and the disabled has become a major public health issue. It is of paramount importance to maintain a good quality of life to allow them to live and keep a good autonomy. Nevertheless, that autonomy can quickly turn into dependence in case of an accident such as a fall or faintness. The solution to solve this dilemma between the autonomy and monitoring is to instrument the environment of the person. Indeed, by automatic monitoring the main environmental characteristics of their living space, it is possible to get a lifestyle pattern of the person. For example, measuring temperature, humidity, luminosity, noise levels, presence, in many strategic areas at home can provide useful data to interpret a physical activity in space and time without direct human intervation.

Such kind of environment can be set up using a WSN. WSNs are composed of a large number of battery-powered sensor nodes that have the ability to sense the physical environment, compute

the obtained information and communicate using the radio interfaces. A specific node, called sink, is in charge of collecting and processing the information.

The sensor nodes have many advantages such as flexibility, no need for fixed infrastracture and low implementation cost, but also several limitations, such as the power source, processing capability, bandwidth, uncertainty of sensed data and the vulnerability of sensor nodes to the physical world. On the other hand, there is a demand for high quality of service (QoS) in WSNs for high sensitive data such as health care applications, that require low end-to-end delay, high reliability in communication and low power consumption. Energy is consumed at node level and routing/MAC level. Transceivers consume a high portion of energy for route selection, but also duty-cycle and mesh connections from the sensors to the sink are important factors in energy consumption. Thus, routing is one of the main sources of energy consumption in a WSN where a sensor datum should be transmitted to the sink in multiple hops by using other sensor nodes. Therefore, the routing in a WSNs is a multi-constrained routing problem (MCRP), satisfying all those constraints (delay, reliability, energy consumption). This decision problem associated to WSN routing is NP-Complete [13]. Several solutions propose to tackle the MCRP considering different metrics, but what is still missing is a multi-constrained routing protocol that takes into account metrics of any nature (but not only additive or multiplicative).

Another important issue is the network lifetime that depends on sensor lifetime. The power depletion represents the most significant part when designing a WSN routing protocol due to the limited capacity of the sensor batteries. Recharging or replacing operations are not usually easy because of unpractical environments or cost reasons. One of the main challenges is how to extend the network lifetime, taking into consideration the energy source limitations. Several energy efficient approaches have been suggested to minimize the energy consumption. In this dissertation, we focus on network flow-based routing protocols. The major goal is to propose a simple but effective strategy to route traffic prolonging the network lifetime and taking into account the MAC layer as well (i.e. combining both network and MAC layers).

1.2 Main Contributions

There are two major contributions in this thesis:

1. The first contribution is focusing to the maximization of network lifetime. We make use of flow based programming models using energy constraints, allowing to balance the energy consumption and thus prolonging the network lifetime. We propose an optimal probabilistic energy aware routing protocol (OPEAR) [10] which makes use of the flows extracted by the optimization model. We compare OPEAR with EAR [33], an existing probabilistic routing protocol, proving that our protocol outperforms the existing one.

2. Second, we propose the Operator Calculus based Routing Protocol (OCRP)[6]. OCRP is a multi-constrained routing protocol which takes into account metrics of any nature. It is based on the Operator Calculus (OC) algebra [11] which is proposed as an offline algorithm. We further extended it to an online version. The offline version of OC is compared with SAMCRA, showing the low complexity of the algorithm [7]. OCRP is based on the online version of OC. It is compared with broadcast, tree routing and RPL proving the benefits of the proposed approach.

We have validated the feasibility and performance of all proposed schemes through detailed emulations and evaluation. We decided to use an emulator and not a simulator, as in many

previous works. This allows to validate our protocols in a realistic environment taking into account the effects of software and hardware of a WSN. We have used the Contiki operating system together with the duty-cycle contikiMAC, which was designed for WSN motes. We have run emulations in Cooja, a software that emulates both the software and hardware of sensor nodes.

1.3 Manuscript Organization

This manuscript is organized in five chapters. The second chapter presents an introduction to Wireless Sensor Networks, and it also gives the reader the necessary elements for understanding the rest of this manuscript. Chapter 3 presents the state of the art including related work on routing protocols in WSNs, classyfing them according to the routing metrics that they use. Chapter 4 presents our first contribution OPEAR and chapter 5 our second contribution OCRP. The thesis is terminated with the conclusion and some final remarks which present motivations for further possible research directions that could stem out from our work.

Part I

State of the Art

Chapter 2

Context

Sommaire

2.1 Wireless Sensor Networks and Internet of Things

With the advance of numerous technologies including sensors, actuators, embedded computing and cloud computing, and the emergence of a new generation of cheaper, smaller wireless devices, many objects, or things in our daily lives are becoming wirelessly interoperable with attached miniature and low-powered or passive wireless devices. The Wireless World Research Forum predicts that by the end of 2017, there will be 7 trillion wireless devices serving 7 billion people [59] (i.e., 1000 devices/person). This ultra large number of connected things or devices will form the Internet of Things (IoT).

The IoT term has been defined in different ways by different authors. Vermesan et al. [57] define the IoT as simply an interaction between the physical world and digital one. The digital world interacts with the physical world using a plethora of sensors and actuators. Another definition is proposed by Peña-López et al. [58] where the IoT is proposed to be a paradigm in which computing and networking capabilities are embedded in any kind of conceivable object. We use these capabilities to query the state of the object and to change its state if possible. Shortly, the IoT refers to a new kind of world where almost all the devices and appliances that

we use are connected to a network. We can use them collaboratively to achieve complex tasks that require a high degree of intelligence.

While IoT does not assume a specific communication technology, wireless communication technologies play a major role, and in particular, WSNs are the key technologies for many applications and many industries. The small, rugged, inexpensive and low powered wireless sensors will bring the IoT even to the smaller objects installed in all kind of environments, at reasonable costs. So, WSNs are like the "eyes and ears" of the IoT, being the bridge that connects the real world to the digital world.

2.2 Applications in WSNs

WSNs are currently being deployed in a variety of applications ranging from medical to military and from home to industry. A WSN can use various types of sensors such as: thermal, seismic, magnetic, visual, infrared, acoustic and radar. These sensors are capable of observing different physical conditions such as: temprerature, humidity, pressure, speed, direction, movement, light, soil makeup, noise levels, presence or absence of certain kinds of objects, and mechanical stress levels on attached objects. To give an idea of possible applications, we shortly describe some of them:

1. Environmental Applications

   It is one of the earliest applications of sensor networks. It is used to monitor the enviromental parameters like air pollution, temperature, humidity, forest fire detection, landslide detection, water quality monitoring and natural disaster prevention [78].

2. Health-care Applications

   WSNs can be used to monitor and track elders and patients at hospitals, clinics, and their home for health care purposes, providing interfaces for diagnosing, drag administration and monitoring of human physiological data such as oxygen measurement, blood pressure, heart rate etc. For example, sensors can be deployed in a patient's home to monitor the behaviors of the patient. It can alert doctors when the patient falls and requires immediate medical attention. With this technology, the health care expenditures can be reduced significantly, while improving significantly the quality of life of patients [79].

3. Home and building automation

   Wireless sensor networks can be used to provide more convenient and intelligent living environments for human beings. For example, wireless sensors can be used to remotely read utility meters in a home like water, gas, electricity and then send the readings to a remote centre through wireless communication for achieving confort while saving energy or other resources [80].

4. Public safety and military Applications

   WSNs are becoming an integral part of military command, control, communication and intelligence systems. Sensors can be deployed in a battle field to monitor the presence of forces and vehicles, and track their movements, enabling close surveillance of opposing forces[82].

5. Industry Applications

In industry, WSNs can be used to monitor manufacturing process or the condition of manufacturing equipment. For example, chemical plants or oil refiners can use sensors to monitor the condition of their miles of pipelines. These sensors are used to alert in case of any failure. Some industrial stadards are WirelessHART [83], IEEE802.15.4e [84] and ISA 100.1b [85].

2.3 WSN Architecture and Protocols

In this section we present the main WSN architectural issues.

2.3.1 The Sensor Architecture

A sensor node is generally composed of four main components. These components are the processing unit, the power unit, the transceiver, and the actual sensors.

- The processing unit is the device entity that executes all the tasks performed by the sensor node. The processing unit is generally a micro-controller. It processes the sensed data and controls the hardware components of the sensor node. To each processing unit is associated an external memory. Two categories of memory based on the purpose of storage can be distinguished: the user memory is used for storing application related and personal data, and the program memory is used for programming the unit. Program memory also contains the identification data of the device.

- The power unit is the power source of the sensor. Power is stored either in batteries or capacitors. Batteries, both rechargeable and non-rechargeable, are the main source of power supply for sensor nodes. The sensor node consumes power for sensing, communicating and data processing. In general, more energy is required for data communication than any other process. An important aspect of a wireless sensor node is to ensure that there is always energy available to power the system. Energy harvesting is a complementary solution[75].

- The transceiver unit is the single communication device of the sensor node and combines both transmitter and receiver. The transceiver is used by the sensor node to send and receive data and communicate with the different neighboring sensors and to the outside of the network. In general cases, the sensor nodes communicate on a wireless transmission media to avoid the installation of heavy wired connections. The wireless transmission media can be Radio Frequency (RF) or optical communication (laser) or infrared. Lasers consume less energy but need direct line-of-sight for communication. Infrared, like lasers, do not need antenna but their broadcasting capacity is limited. The most used wireless media for the sensors is the RF based communication, because it satisfies the requirements of most of the sensors based applications.

  The transceiver can operate in one of four operational states which are transmit, receive, idle, and sleep. The energy consumption level differs from one state to another. For a better utilization of the power, a sensor operates either in idle or sleep mode to consume less energy, if it is not sending or receiving data.

- Sensors are the key hardware devices in a wireless sensor node. The processing unit, transceiver and power unit are important devices since they are needed to support the main functionality of the sensor nodes, which is the detection of events. The sensing unit measures and controls particular events or parameters and produces a measurable response

to a change in a physical condition like temperature, pressure or also objects presence. Each sensor has a certain coverage area in which it can sense efficiently and report the observed events. Sensors measure physical data of the parameter to be monitored and generates an analog signal. The analog signal is digitized by an Analog-to-Digital Converter (ADC) and sent to the processing unit for further data processing. Some wireless sensor nodes contain more than one sensor to be able to detect various types of events using the same sensor node. The evolution and development of sophisticated sensor units have improved and increase the possible sensor node utilizations.

Figure 2.1 depicts the architecture of a wireless sensor node including the most common included hardware components.

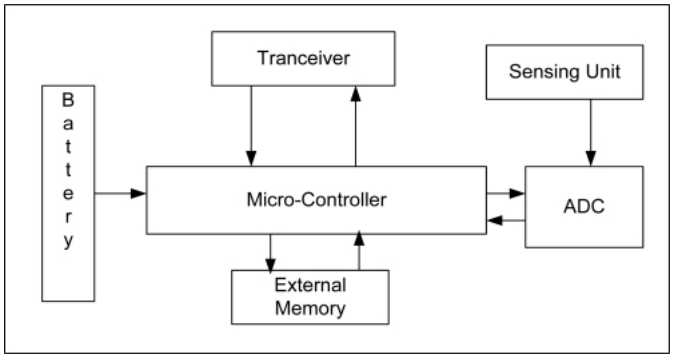

Figure 2.1: The sensor node architecture [73].

The individual nodes in a WSN are inherently resource constrained: they have limited battery or energy harvesting capacity, processing speed, storage capacity and communication bandwidth. The working mode of the sensor nodes may be either continuous or event-driven. Global Positioning System (GPS) and local positioning algorithms can be used to obtain location and positioning information.

2.3.2 The WSN Network Architecture

WSNs can be defined as a self-configured and infrastructure-less wireless network able to monitor physical or environmental conditions and to cooperatively pass their data through the network to a main location or sink where the data can be observed and analysed. A sink or base station acts like an interface between users and the network. A generic WSN architecture is depicted by Figure 2.2.

WSN differs from the traditional wireless networks due to its limited resources (power, processing and memory), low node reliability and dynamic network topology. Thus, the routing between the sensors and the sink node, as well as the radio duty cycle play a very important role in WSNs. From a layered view, MAC determines the channel access delay and utilization and also the energy consumption (active-sleep mode timing) through a duty cycle mechanism. On the other hand, the network layer should provide a protocol with low end-to-end multi-hop transmission time, which also reflects the enery consumption. Thus, QoS in WSNs is provided through a set of measurable service parameters such as delay, jitter, available bandwidth, and packet loss. Beside these metrics, network lifetime is also to be included. Cross-layer design is adoped to achieve a joint optimization. It aims to improve the performace of a communication protocol by taking into account parameters of other layers.

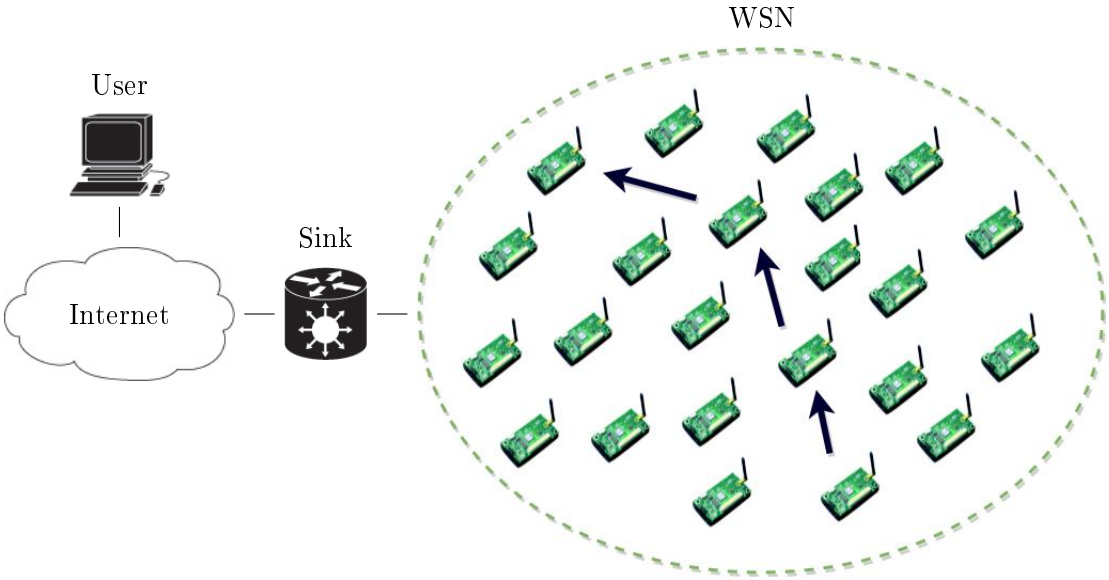

Figure 2.2: Wireless Sensor Network Architecture

2.4 QoS Metrics in WSNs

2.4.1 Network Lifetime

The most challenging issue in WSNs is the network lifetime since the sensor nodes are energy constrained. In the literature there is a great number of lifetime definitions. The definition found most frequently is that the network lifetime T ends as soon as the first node fails, thus

$$T = min_{v \in V} T_v \tag{2.1}$$

with T_v being the lifetime of node v. The sink nodes are excluded from the node set V since they are power plugged [39].

Another variant of network lifetime is the time until the fraction of alive nodes falls below a predefined threshold β, or the time during which at least k out of n nodes are alive. This definition lacks accurancy, in fact in case that $k' < k$ nodes at strategic positions die and the remaining nodes do not have the possibility to transmit any data to the sink the network would not recognize this problem until another $k - k'$ nodes have failed [40].

Hellman and Colagrosso [41] improved the above definition dividing the nodes to critical and non-critical and they allow no fail from critical nodes and k fails from non critical nodes. Other definitions are cluster based, assuming that each cluster has a cluster head and the lifetime is the time till the first cluster head fails [42].

Lastly, it is possible to define the lifetime as the time until all nodes have been drained of their energy [44]. This metric is not used frequently since it is too optimistic assuming that useful information can be sent till the last node dies.

In this thesis, we adopt the first network lifetime definition, since it is the most general and coservative one. We do not have to deal with topology changes, the nodes have equal importance and they are all critical to the network operations. This definition ideally requires a complete uniform depletion of energy to allow the utilization of the network for the longest possible time.

The advantage is that it is a conservative measure, in fact the WSN can be still available even if one sensor failed, therefore it is well adapted to health-care applications where information is sensible.

2.4.2 Type of QoS Metrics

Let's consider a routing path from the source sensor to the sink. The quality of service of this path is measured by using metrics as delay, delay variation (jitter), hop count, energy, reliability, throughput etc. We can distinguish three types of QoS metrics:

- Additive (e.g. delay, jitter, cost, hop count) in which the weight (the weighted graphs associate a value (weight) with every edge in the graph - the words cost or length are also used instead of weight) of that measure equals the sum of the QoS weights of the links defining that path.
- Minimum/Maximum (e.g. energy, available bandwidth and policy flags) is the maximum or the minimum of the metric over all the links in the path.
- Multiplicative (e.g. loss probability or probability of successful transmission) which is the multiplication of the metric for all the links constituting the path. This form can be transformed into additive form by using its logarithmic form.

2.5 Energy Efficient Techniques at MAC and Network Layers

2.5.1 Energy Consumption Model

From the view of upper layer protocols, the sensors are usually assumed to have four states and no cost for transition between the states [66]:

- Transmission: processing for address determination, packetization, encoding, framing, queuing.
- Reception: Low-noise amplifier, filtering, detection, decoding, error detection, address check, reception even if a node is not the intended receiver.
- Idle listening: Similar to reception except that the signal processing chain stops at the detection.
- Sleeping: a low power level allowing the sensor node to stay alive.

Other researchers try to take more realistic factors into consideration [67] and some assume that since the nodes are expected to receive, transmit and listen at full power, for which current demands are approximately the same, two states can represent efficiently the radio operation [68]: active and sleep mode.

2.5.2 Sources of Energy Waste

In WSNs, sensors dissipate energy while sensing, processing, transmitting or receiving data to fulfill the mission required by the application. The sensing subsystem is devoted to data acquisition. It is obvious that minimizing generated data will save energy of very constrained sensors. Redundancy inherent to WSNs will produce huge similar reporting that the network is

in charge of routing to the sink. Therefore, considering that transmission provokes high energy consumption, the communication subsystem is a greedy source of energy dissipation.

With regard to communication, there is also a great amount of energy wasted in states that are useless from the application point of view, such as:

- *Collision*: when a node receives more than one packet at the same time, it is said that packets collide. All packets involved in the collision have to be discarded and retransmitted.
- *Overhearing*: when a sender transmits a packet, all nodes in its transmission range receive this packet even if they are not the intended destination. Thus, energy is wasted when a node receives packets that are destined to other nodes.
- *Control packet overhead* : a minimal number of control packets should be used to enable data transmissions.
- *Idle listening*: when a node is listening to an idle channel in order to receive possible traffic, it is one of the major sources of energy dissipation.
- *Routing level*: data broadcast communication, topology maintenace and route maitenance require lot of "hello" packets.

2.5.3 Energy Saving Approaches

We can identify different energy saving mechanisms which are categorized as follows:

- *Energy efficient routing at Network layer*: routing protocols can be designed with the target of maximizing network lifetime by minimizing the energy consumed by the end-to-end transmission and to avoid using nodes with low residual energy. Some protocols consider energy as a metric for path selection. By doing so, routing algorithms can select the next hop by focusing not only on the shortest path but also on its residual energy [33], [34].
- *Duty cycling at MAC layer*: duty cycling is the fraction of time nodes are active during their lifetime. The periods during which nodes sleep or are active should be coordinated and accommodated to the specific application requirements. Duty cycle techniques can be further subdivided. High granularity techniques focus on selecting active nodes among all sensors deployed in the network. Low granularity techniques deal with switching off (respectively on) the radio of active nodes when no communication is required (respectively when a communication involving this node may occur). They are highly related to the medium access protocol [23].
- *Multipath routing*: single path routing rapidly drains energy of nodes on a selected path and when the node drains out of power, a new route must be reconstructed. Multipath routing in contrast, alternates forwarding nodes thereby balancing energy among the nodes. It enables the network to recover faster from failure and enhances the network reliability [18].
- *Relay node placement*: the early stage depletion of nodes can be avoided by the even distribution of nodes by placing a few relay nodes, i.e. nodes that do not have sensor functions but they are used only for routing. This improves the energy equilibrium between nodes, coverage, and capacity and avoids sensor hot spots [77], [76].

- *Coverage Optimization*: it is mainly related to the minimization of the number of active nodes while still ensuring the connectivity and coverage of the monitored zone. Taking advantage of the overlapping sensing regions in a high-density WSN by activating certain nodes and turning off others can prolong significatly the network lifetime [70].
- *Cluster architectures*: some WSNs are organized as clusters of sensors. Each cluster has a cluster head (CH) that takes the responsibility of coordinating the communication activities of the members, like gathering sensed data from the cluster nodes and defining a scedule for assigning timeslots to the nodes for transmission. CHs communicate with other CHs or to the base station using aggregated data. Cluster architecture enhances energy efficiency by limiting energy consumption of the nodes mainly by data gathering, possibility of transmitting only specific functions of sensor measurements to the base station and switching CHs according to their residual energy, having a more balanced energy dissipation[86].
- *Sink mobility*: a huge workload is concentrated on the nodes closer to the sink (base station) since all the traffic is directed towards the sink through them. Hence their battery gets depleted faster than the one of other sensor nodes. The load can be balanced by allowing a mobile base station which collects node information by moving in the network. Sink mobility improves connectivity, reliability and reduces collision and message loss[87].
- *Topology control*: it focuses on reducing energy consumption by adjusting transmission power while maintaining network connectivity. A new reduced topology is created based on local information [2].

2.6 Conclusion

WSNs are one of the enabling technologies for the IoT. However, the resource constrained nature of sensors raises the following crucial problem: how to maximize network lifetime despite a very limited energy budget, while keeping a high QoS?

In this chapter, we have summarized the main WSN architecture features, the QoS metrics in WSNs and we described the different techniques to tackle the energy efficiency challenge in WSNs.

In this thesis we focus on the energy saving technique based on routing at network layer. In the next chapter we will present the existing routing protocols for WSNs. We classify the protocols according to different categories of routing metrics and we describe them in detail.

Chapter 3

Routing Protocols in WSNs

Sommaire

3.1 Introduction

In this chapter we survey QoS routing algorithms and routing protocols for WSNs which deal with two main problems: maximizing the network lifetime and the multi-constrained routing. As we discussed at Chapter 2, the lifetime of WSNs is very limited, since the most of the devices are energy constrained. Thus, the most challenging issue in WSNs is how to maximize the network lifetime. This problem has been widely studied in the literature, however, finding an optimal solution for maximizing the network lifetime is still an open problem. Moreover, many applications in WSNs and IoT require delay QoS, high reliability links and high throughput. Therefore, the routing problem in WSNs becomes a multi-constrained routing problem.

In this thesis, we consider both routing algorithms in graphs and routing protocols in networks. The routing algorithms use the routing metrics to determine the paths that a data packet will follow from the source to the destination. A routing protocol uses the routing algorithm to route the data packets, plus it defines the format of routing information exchanges and the routing updates. We present both routing algorithms and routing protocols (also the routing protocols which make use of the presented routing algorithms), and we classify them according to their routing metrics.

We can identify two main categories according to how QoS metrics are used to determine routing: the single-metric routing protocols where the forwarders are selected according to one metric and the multiple-metric routing protocols where multiple metrics are considered. The multiple-metric routing protocols can be further classified as multiple-combined metric routing protocols where the cost function is defined by a single aggregated function of multiple metrics, and the multiple separate-metric routing protocols, which consider more than one metric in a not aggregated form. Protocols using network flow and modeling are also presented. In Figure 3.1 a schematic view of routing algorithms and routing protocols is shown.

3.2 Single-Metric Routing Protocols

3.2.1 Hop Count

Hop count is one of the most frequent metrics and it is used in several protocols as Ad-hoc On-Demand Distance Vector (AODV)[35] and Dynamic Source Routing (DSR) [36]. Every link count as one equal unit independent of the quality or other characteristics of the link. This metric basically brings loop free paths with the minimum number of links. Even if it is considered as a simple metric, it may not result in good performances. In fact, shortest-hop based routing neglects completely energy or QoS measures.

AODV

AODV is a distance vector routing protocol that has been used for ad-hoc wireless networks and Zigbee, the low-cost and low-power wireless mesh network standard. It is an on demand protocol where routes are found when it is required to send a data packet. It mainly uses the hop count to define paths between the source and the destination. When a source node intends to communicate with a destination node whose route is unknown, it broadcasts a ROUTE REQUEST packet that contains the hop count. Each recipient of the ROUTE REQUEST packet that does not maintain a route to the destination rebroadcasts the same packet after incrementing the hop-count. Such intermediate nodes also create and preserve a REVERSE ROUTE to the source node for a certain interval of time. A ROUTE REPLY packet is generated and unicasted back to the source as

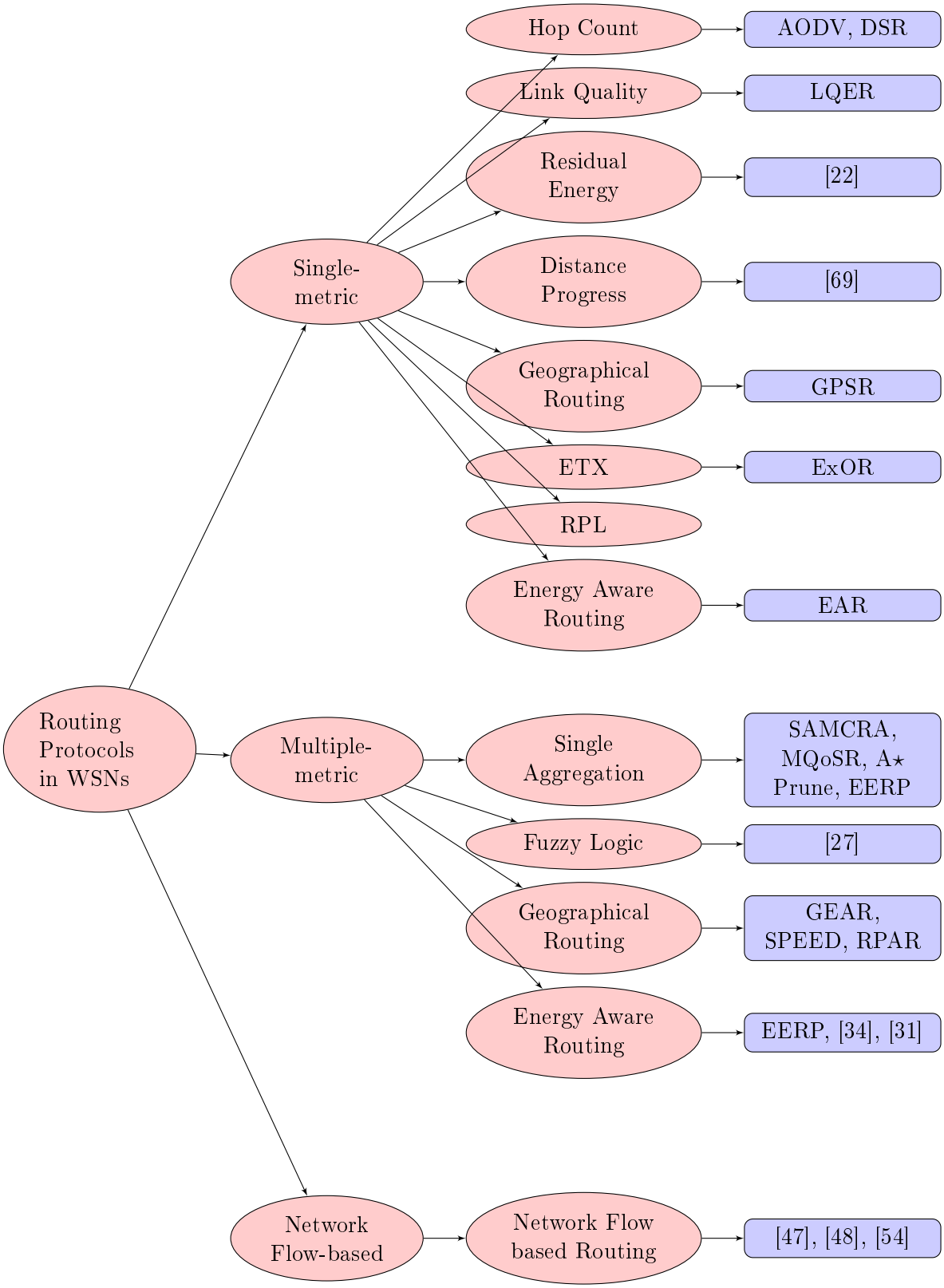

Figure 3.1: Routing Protocols in WSNs.

soon as the ROUTE REQUEST packet has reached the destination, following the same route upwards. AODV maintains these routes as long as they are needed by the source.

The main drawbacks of AODV is that broadcast wastes a big amount of energy and hop count based routing is not enough in WSNs, since links can be unreliable and asymmetric.

DSR

DSR is a distance vector, on demand routing protocol and the way it finds routes is actually similar to AODV (ROUTE REQUEST, ROUTE REPLY etc.). The main difference is that DSR sets the whole route to the destination for each packet, indicating the addresses of the nodes that it has to pass through (DSR header), while AODV does not require to maintain routes to destination that are not in active communication.

3.2.2 Link Quality

This metric is related to the signal strength which is a good indicator for measuring the link quality since a packet can be transferred successfully when the signal strength is more than the threshold value.

Chen et al. [51] propose a Link Quality Estimation based Routing protocol (LQER) which considers the link quality to avoid poor link connectivity and reduce the possibility of retrasmissions to be able to increase the lifetime of WSNs and data reliabilility. LQER estimates the link quality before making the routing decisions, by creating a connectivity graph based on hop count field. The link quality is estimated under the concept of dynamic window (m, k): m data packets successfully transmitted out of window of k data packets.

3.2.3 Residual Energy

Routing protocols which aim at improving the network lifetime, mainly use the metric of residual energy for forwarding data packets. In [22], a routing algorithm is proposed which makes use of $A^{\star}$ algorithm to find the optimal path from the source node to the sink node using as metric the nodes' energy level. Nodes with higher residual energy are prioritized, leaving the weak nodes out of the routing operation.

EAR

The Energy Aware Routing protocol [33] is a reactive protocol that aims to increase the lifetime of the network by using also sub-optimal paths, that provide longer connectivity, in order to avoid lowest-energy ones. The main idea of this protocol is to spend the energy in the nodes more equitably than finding one single path and then fully consume the energy of the nodes composing it. A set of paths is chosen and maintained in a probabilistic fashion.

The optimization objective pursued by EAR is network survivability which is mainly the lifetime of the network under which full connectivity is provided. The authors argue that good solutions cannot be obtained by simple energy optimizing protocols, finding just optimal paths, as these leave a wide disparity in the energy levels of the nodes, and eventually lead to disconnected subnetworks. Indeed, the authors of [33] prove that EAR outperforms Directed Diffusion routing in the sense that it better spreads the traffic over the network and increases the time till the first node fails.

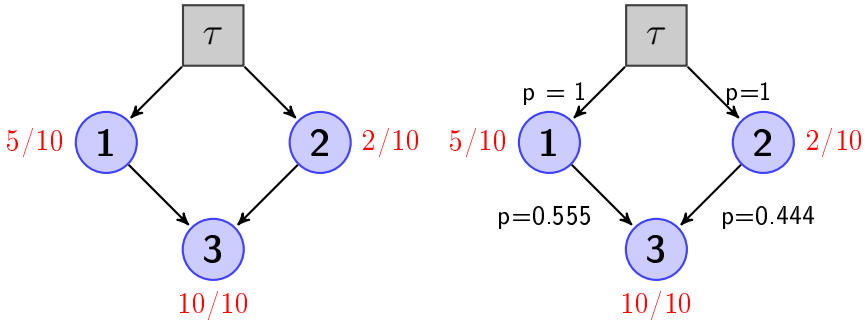

Figure 3.2: a) EAR - Graph Example, b) EAR - Graph Example with assigned probabilities

The EAR protocol is implemented in three phases. First, a flooding process is used to discover all the routes between source/destination pairs with respective costs and routing tables are built-up. Second, a data communication phase is carried out during which data packets are sent from source to destination through preferred links which are chosen according to a probability. Third, a maintenance phase is executed infrequently where a localized flooding is performed again for updating the routing probabilities.

The EAR protocol assumes that the (normalized) residual energy R_i of each sensor i, the energy $e_{i,j}$ required to route a packet on each link (i,j), and the set of sensors F_i that can be reached by sensor i are known. For each link (i,j), an energy cost measure is defined as follows:

$$c_{i,j} = e_{i,j}^{\alpha} R_i^{\beta},$$

where α and β are two parameters to be defined by the planner. A vector of costs is computed for each sensor i, having one component for each $j \in F_i$:

$$C_j^i = \tilde{C}^i + c_{j,i},$$

where $\tilde{C}^i$ is the average of the C_j^i values for sensor i

$$\tilde{C}^i = \sum_{j \in F_i} C_j^i / |F_i|.$$

Assuming $\tilde{C}^{\tau} = 0$ for a sink sensor τ, all C_j^i values, and therefore the $\tilde{C}^i$ values, can be computed by a simple recursion on network links.

Finally, a routing probability

$$P_{i,j} = \frac{1/C_j^i}{\sum_{k \in F_i} 1/C_k^i}$$

is computed for each link (i,j), and routing is then performed according to these probabilities.

Let us consider the graph depicted in Fig 3.2. Node τ is the sink node and broadcasts its cost initialized in zero ($\tilde{C}^i = 0$) to its neighbors. The nodes 1 and 2 are calculating the cost of the path to the sink node:

$$C_1^{\tau} = \tilde{C}^{\tau} + c_{1,\tau} = 0 + \frac{5}{10} = \frac{1}{2}, \quad P_{1,\tau} = 1, \quad \tilde{C}^1 = \frac{1}{2}$$
$$C_2^{\tau} = \tilde{C}^{\tau} + c_{1,\tau} = 0 + \frac{2}{10} = \frac{1}{5}, \quad P_{2,\tau} = 1, \quad \tilde{C}^2 = \frac{1}{5}$$

assuming a simple energy model of spending one unit of energy for each transmission and each reception on the link, so the parameter $e_{i,j}$ is equal to 1.

After receiving the costs from node 1 and 2, node 3 performs similar simulations.

$$C_3^1 = \tilde{C}^1 + c_{3,1} = \frac{1}{2} + 1 = \frac{3}{2}, \quad P_{3,1} = \frac{5}{9}$$

$$C_3^2 = \tilde{C}^2 + c_{3,2} = \frac{1}{5} + 1 = \frac{6}{5}, \quad P_{1,\tau} = \frac{4}{9}$$

$$\tilde{C}^3 = \frac{5}{9} * \frac{3}{2} + \frac{4}{9} * \frac{6}{9} = \frac{41}{30}$$

As we can observe, the probability of node 3 to send a packet to the sink through node 1 is higher than node 2 because node 1 has higher residual energy normalized to the initial energy. Since the flooding process is performed frequently the routing tables are updated frequently. After the first flooding, the probabilities are all uniform even if some nodes have higher initial energy than others. This happens because the residual energy is normalized to the initial energy, so it is 1. This may cause problems since in applications like smart-homes the energy of sensor nodes may vary since the sensors have different nature and different capacities.

3.2.4 Geographical Routing: GPSR

Geographical routing uses location information to formulate an efficient route search toward the destination. It requires only the propagation of single hop topology information, like the best neighbor, to make forwarding decisions.

GPSR

The Greedy perimeter Stateless Routing (GPSR) [45] is one the earliest works on geographical routing. To calculate a path, GPSR uses a greedy forwarding algorithm that sends the packets closer to the destination, supposing that nodes know their location and their neighbors location. If the greedy forwarding fails, the perimeter forwarding is used which routes the packets around the perimeter of the region, allowing thus to bypass network topology holes.

3.2.5 Distance Progress

Distance Progress (DP) [69] is an opportunistic routing metric. In opportunistic routing the function of broadcast is used for transmitting the data packets. A candidate set of next-hop forwarders are selected according to the routing metric. Any of the candidates of a node that have received the transmitted packet forward it towards to destination.

The closeness of nodes to the destination is considered as the measurement for the selecting and prioritizing of candidates. The definition of DP is the following:

$$DP_{c_i}^{s,d} = D(s,d) - D(c_i,d) \tag{3.1}$$

where $D(i,j)$ is the Euclidean distance between nodes i and j. s is the source node, d the destination and c_i the candidate node for forwarding. A node can estimate the remaining distance needed to reach the destination d, when the packet is delivered to candidate c_i. Each node can send a beacon message to provide its position information. The geographic position of the destination is needed, therefore it may be provided by a location-based service.

3.2.6 ETX

Expected Transmission Count (ETX) [29] is one of the most common routing metrics. It is associated to the minimum hop count and reliability parameters like Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI). More specifically, ETX is the expected number of transmissions required to successfully transmit a packet from a node to its neighbor.

Its computation is based on the delivery ratio d_f which is the probability of a data to be delivered. And the reverse delivery ratio d_r represents the probability that the ACK packet is successfully received back. The expected probability that a message is successfully received is $d_f x d_r$. If we consider a packet transmission as a Bernoulli trial (success or fail), ETX is calculated as following:

$$ETX = \frac{1}{d_f x d_r} \tag{3.2}$$

d_r and d_f are computed by sending broadcast probe packets at an average period τ. A jitter parameter is added up to $\pm\tau$ per probe to avoid synchronization. A node receiving a probe must rebroadcast it and allows for the originator of a probe to compute the d_r probability. To compute d_r each node remembers the packet probe sent and triggers a timer. If this probe is received back within a period (i.e. the window size) it is considered as received, otherwise it is considered as lost. The d_f is computed by counting incoming probes native from neighbours every $\pm\tau$ time.

ExOR: Extreme Opportunistic Routing

ExOR [30] is an opportunistic routing protocol for WSNs. Opportunistic routing differs from traditional routing techniques because the route is typically generated on the fly and any set of candidate nodes that have received the transmitted packet may forward it. It uses broadcast nature of wireless networks to forward the packets through the network. ExOR uses a potential forwarders' list to reach the destination. The forwarding decision is made on the receivers' side. All potential forwarders have their own transmission time slot sorted by routing progress and forward only if they have not overheard transmissions from others. This process selects the best available next hop in a distributed way and avoids duplicate forwarding.

3.2.7 RPL

RPL [19] is a routing protocol for Low Power and Lossy networks (RPL) and it has been standarized by IETF. It makes use of a unique forwarding metric among several possible choices, like node state, energy, hop count, throughput, latency, link quality and ETX. The network topology is organized as a Destination Oriented Directed Acyclic Graph (DODAG), rooted at a sink node. Each node can have multiple parents directed to the sink, but only one parent is chosen according to the selected metric.

A network can operate with one or more RPL instances which consist of multiple DODAG graphs and each RPL instance can be related to a different objective function according to which the best path from the source to the sink is calculated. Each RPL instance can be identified by the *instance-id*.

There are three types of Internet Control Message Protocol (ICMP) messages which are sent frequently between the nodes.

- DODAG Information Object (DIO): it is used for the creation and maintance of upward routes. The root first broadcasts a DIO message, informing the RPL nodes with the

required parameters to find out their related to the sink position and the RPL instance. Upon receiving the DIO message, the RPL nodes add the sender in the parent list and each node calculates its own rank taking into account the objective function. At the end of this process, all nodes have a default parent towards the root of the DODAG.

- DODAG Destination Advertisment Object (DAO): it is used to install downward routes. DAO messages are sent by the nodes periodically to notify the parents about the routes of downward nodes, the valid lifetime etc. Nodes aggregate the DAO messages received by various nodes and forward them to the parent.
- DODAG Information Solicitation (DIS): it is used by a node which solicits graph information from the neighboring nodes or wishes to join the network.

RPL nodes receive DIO messages from different neighbours and they have to choose the preferred parent. This decision is taken according to the objective function. IETF has introduced several metrics for the objective function like node state, energy, hop count, throughput, latency, link quality and ETX. However, if more than one metric has to be considered to increase the QoS, there is no indication about the way that this can be achieved. So although several metrics are designed, only one metric can be used. RPL is not designed to solve a multi-constrained routing problem. Another drawback of RPL is that the DODAG maintenance requires high energy consumption.

ORW

Opportunistic routing in WSNs (ORW) [52] applies opportunistic routing in duty-cycled data collection networks. Instead of using unicast communication, nodes anycast any packet until any valid forwarder receives and acknowledges it. Opportunistic routing originally was developed to improve throughput in multihop mesh networks, considering that radios are always on and can overhear messages with no additional cost. But, WSNs are duty cycled and the most critical metric is the energy consumption. Therefore, the authos of [52], trying to adapt opportunistic routing to the demands of WSNs, proposed a metric which estimates the expected number of wakeups required for the end-to-end packet delivery, named Expected Number of Duty-Cycled Wakeups (EDC) which is calculated as follows:

$$EDC_i(S_i) = \frac{1}{|S_i|} + \frac{\sum_{j \in S_i} EDC_j}{|S_i|} \tag{3.3}$$

where S_i represents the neighborhood of node i. The first term is the expectd wait time of the node, which is inversely proportional to the number of neighbors. The second term is the average expected wait time after the packet is forwarded to the next hop. The sum of these two terms is the expected wait time experienced from the node to the sink.

While traditional opportunistic routing choose next hop forwarders according to ETX metric to increase the throuphput, ORW focus on energy efficiency choosing paths that reduce both the radio-on time and the delay.

ORPL

Opportunistic RPL (ORPL) [53] is an extension of RPL, where packets are forwarded opportunistically towards their destination. ORPL replies on anycast at the MAC-layer, so nodes route towards the network root using all available parents as potential parents. ORPL uses EDC

metric. It performs better compared to RPL in terms of reliability , delay and energy consumption in dense networks, but when a few nodes are connected to the network at the same time the impact of opportunistic strategies are limited.

3.2.8 Discussion

As we have seen above, single-metric routing protocols rely on different performance metrics to be optimized, such as hop count, residual energy, link quality etc. However, in practice these metrics often conflict with each other, e.g. paths composed by nodes with high residual energy may not include reliable links. Hence, there is need of using multi-metric routing protocols, which can more carefully balance of the trade-offs among different metrics and thus the overall performance of WSNs can be optimized and offer the required QoS in real applications.

3.3 Multiple-Metric Routing Algorithms

Some routing algorithms handle the problem of having multiple metrics through the use of cost functions which regroup all metrics into a single function. This has the advantage of transforming the problem into a single-objective problem which is easier to solve. Nevertheless, this approach is not possible when the metrics have different natures. Moreover, if constraints are taken individually, using cost functions may lead to paths which violate some constraints.

Constraints are metrics with a threshold of upper/lower value and objectives are desired attributes of the network. QoS metrics can be either constraints or objectives. But, in many protocols there is a maximum threshold for certain metrics but the protocol works trying to minimize these metrics, so there is no guarantee that this constraint will be under a threshold.

3.3.1 SAMCRA

SAMCRA is an exact, centralized, multi-constrained algorithm [14]. It terminates as soon as it finds one feasible path between two nodes. It works with additive QoS metrics like delay, jitter, link cost etc. SAMCRA does not consider explicitly min(max) QoS metrics because they can be treated by omitting all links which do not satisfy the requested min(max) constraints. This "topology filtering" is performed before the execution of the algorithm.

Let us consider a graph G in which each link $u \to v$ from node u to node v is characterized by a m dimensional weight vector $\overrightarrow{w}(u \to v) = [w_1(u \to v), w_2(u \to v), \ldots, w_m(u \to v)]$ where the component $w_i > 0$ is a QoS measure. A path P obeys multiple constraints $w_i(P) \leq L_i$ for all $1 \leq i \leq m$, where $\overrightarrow{L}$ is the constraint vector.

While SAMCRA proposes a non linear path length, first Jaffe [43] proposed a linear path length

$$l(P) = \sum_{i=1}^{m} d_i W_i(P) \tag{3.4}$$

where d_i are positive real numbers. In that case, the m-parameter problem is transformed to a single parameter problem enabling the use of Dijkstra's shortest path algorithm. When minimizing a linear fuction of the weights, solutions outside the constraints area may be returned. An important corollary of a nonlinear path length is that the subsections of shortest paths in multiple dimensions are not necessarily shortest paths themselves. So, SAMCRA returns the

path which has the minimum non-linear path length:

$$l(P) = max_{1 \leq i \leq m}(\frac{\sum_{j=1}^{k} W_i}{L_i}) \tag{3.5}$$

If the shortest path computed with the above length definition has length larger than 1, then it violates at least one of the constraints, and no other path will satisfy the constraints (and respectively if $\frac{W_i}{L_i} \leq 1$ the constraint is satisfied). Thus, SAMCRA solves the multiple constrained problem using the definition 3.5.

SAMCRA is based on two more principles: the $k-$shortest path approach and the non-dominance. The k-shortest path approach is used when more than one possible paths between a source and a destination is needed to be found. So, the algorithm does not stop when the destination has been reached, but it continues till the destination has been reached k times. Therefore, the intermediate nodes store not only one but multiple sub-paths from the source to each intermediate node, but only when these sub-paths are non-dominated[74]. Two paths Q and P are non dominated if there is at least one weight out of several in Q's weight vector which has a more promising value that the respective weight in P's weight vector. This basically means that no sub-path is better or worse than the others and each sub-path has a better value at least in one metric. This property allows to efficiently reduce the search-space without compromising the solution. In each step, SAMCRA chooses the sub-path with the minimum path length and this process lasts till the path has reached the destination and it is also a feasible path.

Although SAMCRA is an efficient exact algorithm for solving the multi-constrained routing problem, it has important limitations. SAMCRA works only with additive and non-negative constraints which is not always the case in WSNs.

3.3.2 A* Prune

The A* algorithm [16] is an heuristic algorithm which finds k-shortest paths. Each node n keeps a cost function $f(n)$ which is the combination of the path cost function $g(n)$ (the cost from the starting node to the current node n) and a heuristic estimate of the distance from n to the destination $h(n)$.

$$f(n) = g(n) + h(n) \tag{3.6}$$

The algorithm is capable to find feasible paths satisfying a set of either additive or boolean constraints and at the same time optimizing the function $f(n)$.

The algorithm consists of two major steps: pre-computation and path expanding/pruning. To deal with the additive constraints, it performs a pre-computation of their associated Dijkstra distances from the current node to the destination. This pre-computation can be done in the background and stored for use by multiple path computations as long as the topology remains unchanged.

In expanding/pruning stage, A* keeps a priority queue of feasible paths found up to now. Initially, it contains the path $p(s, s)$, which consists of the single node s. At every step the paths are expanded one hop, pruning those which create a loop or violate the boolean constraints. For the additive constraints, it combines the current path cost with the associated Dijkstra cost to the destination and it is compared against the additive constraints.

The algorithm finds the shortest feasible paths in advance, so the whole process continues till k-shortest paths are found.

3.4 Multiple-Metric Routing Protocols

3.4.1 Geographical Routing: GEAR, SPEED, RPAR

GEAR

The Geographic and Energy Aware Routing (GEAR) [46] is an energy-efficient routing protocol for WSNs, where the sensors are supposed to have localization hardware equipment, like GPS, so that they know their current position. Furthermore, sensors are aware of their residual energy and also of the location and the residual energy of nodes in their neighborhood. The selection of next hops is done according to a cost function, which is related to the residual energy and the location of the neighbors. Morevoer, the network is divided in some regions and the idea is to restrict the requests to certain regions, not to the whole network. In that way, packets are routed to a "target region" instead of a particular node. After reaching the region, geographical routing or flooding is performed to reach the destination. At node N the estimated cost $c(N_i, R)$ of neighbor N_i is the following:

$$c(N_i, R) = ad(N_i, R) + (1 - a)e(N_i) \tag{3.7}$$

where R is the target region, a is a tunable weight, $d(N_i, R)$ is the distance from N_i to the region R normalized by the largest distance among all neighbors of N, and $e(N_i)$ is the consumed energy at node N_i normalized by the largest consumed energy among neighbors of N.

GEAR considers also the case of having holes, if there is no neighbor closer to the destination. In this case, the nodes propagate the packets one hop backwards and try to follow other paths towards the destination.

SPEED

SPEED[1] is another geographic routing protocol for sensor networks that provides soft real-time end-to-end guarantees.

The protocol requires each node to maintain information about its neighbors and uses geographic forwarding to find the paths. Each node i keeps a neighbor set NS_i which includes the neighbors that are inside the radio range of node i and a forwarding candidate set FS_i which includes the nodes from set NS_i that are closer to the destination. Packets are forwarded only to the nodes that belong in FS_i.

In addition, SPEED strive to ensure a certain speed for each packet in the network so that each application can estimate the end-to-end delay for the packets by dividing the distance to the sink by the speed of the packet before making the admission decision.

Nodes in FS_i are categorized according to their speed to reach the destination. Higher speed nodes have higher probability to be chosed as forwarding nodes. The speed of each node is the fraction of the advance in distance from the next hop node j by the estimated delay to forward a packet to node j.

$$Speed_i^j(Destination) = \frac{L - L_{next}}{HopDelay_i^j} \tag{3.8}$$

where L is the distance from node i to the destination and L_{next} is the distance from the next hop forwarding candidate j to the destination.

Moreover, SPEED can provide congestion avoidance when the network is congested as it considers delay.

RPAR

Real Time Power Aware Routing Protocol (RPAR) [2] is an extension of SPEED and has been designed for real time routing for WSNs. It adapts the transmission power and the routing decisions depending the workload and packet delay deadlines for satisfying the application delay requirements. The key feature of this protocol is adaptability, e.g. for tight deadlines it trades energy and capacity to meet the desired delay constraint and for loose deadlines, it lowers the transmission power to increase the throughput.

Like SPEED, RPAR uses the velocity assignment module, which maps a packet deadline to a required packet velocity (equivalent to packet speed of SPEED). In other words, velocity is an estimation of the distance between the source and the destination and the end-to-end delay. When a node wants to forward a packet, it uses the velocity assignment module to calculate the required velocity based on the remaining distance between the present node and the destination and the Time-To-Live (TTL).

There is also a delay estimator module, which calculates the one-hop delay of different forwarding choices e.g. the time that a node takes to deliver a packet to the neighbor N at power level p. RPAR forwards the packet to the most energy efficient forwarding choice among the choices that meet the required velocity of the packet.

The velocity provided by (N, p) is calculated as follows:

$$u(s, d, (N, p)) = \frac{D(s, d) - D(N, d)}{delay(s, (N, p))} \tag{3.9}$$

where $D(s, d)$ and $D(N, d)$ are the distance between the source s and the destination d and between the current node and the destination respectively. The progress made toward the destination by forwarding the packet to N is $D(s, d) - D(N, d)$. The estimated delay of forwarding choice (N, p), $delay(s, (N, p))$, approximates the time interval from the time that a packet becomes the head of the tranmission queue until it is received at the next hop.

RPAR estimates the energy cost of all eligible forwarders which is given by the following formula:

$$E(s, d, (N, p)) = E(p) \cdot R(s, (N, p)) \cdot \frac{D(s, d)}{D(s, d) - D(N, d)} \tag{3.10}$$

where $E(p)$ is the energy consumed for transmitting a packet at power level p and $R(s, (N, p))$ is the expected number of transmissions before s successfully delivers a packet to N when transmitting at power level p. $D(s, d) - D(N, d)$ represents the progress towards d when N is selected as next hop.

3.4.2 Single Aggregation of multiple metrics

MQoSR

In [17] the authors suggest the Multi-Objective QoS routing protocol (MQoSR) for WSNs. Multiple metrics are taken into account such as energy, delay, reliability and hop count. The next hop forwarder is the one with the minimun total cost C_{total}.

$$C_{total} = C_{link} C_{req} \tag{3.11}$$

where C_link is the link cost function and C_req is the required QoS funtion. C_{link} is calculate as following:

$$C_{link} = \frac{1}{d_{s,d} - d_{N,d}} \tag{3.12}$$

where $d_{s,d}$ is the distance between the current node to the destination and $d_{N,d}$ is the distance between the possible forwarding node to the destination. This cost basically tries to minimize the hop count between the source and the destination.

C_{req} divided into delay, reliability, and energy consumption and it is calculated after assigning the weighting factors C_E, C_D and C_R that are related to each other by the formula $C_E + C_D + C_R = 1$. The required QoS funtion, C_{req}, is the following:

$$C = \frac{E_{req}}{E_{ava}} C_E + \frac{D_{link}}{D_{req}} C_D + \frac{R_{req}}{R_{link}} C_R \tag{3.13}$$

where E_req, D_{req}, R_{req} are application-specific parameters which reflect the required end-to-end energy, delay and reliability respectively for data delivery ,E_{ava} is the residual energy of the current node, D_{link} is the delay required to send a packet to a certain neighbor and R_{link} is the reliability between the current node and a certain neighbor.

MQoSR applies a different selection policy for each QoS requirement, so while C_{req} is stable, C_{req} can be varied according to QoS requirement. The node selects the next hop node based on the requested requirements and the link conditions for improving the QoS provisioning.

EERP

In [50] an Energy-Efficient Routing Protocol (EERP) is proposed for WSNs using A$\star$ algorithm 3.3.2. The authors modify the value of $f(n)$ by adjusting the parameters contained in the sensor such as the value of the energy used, the minimum hop count and the free buffers. Thus, instead of using $g(n)$ which can be the link cost or hop count or any other additive metric, they use a combination of many metrics through an aggregated weight. This weight is composed of the residual energy, the link quality and the number of free buffers of the neighbor. The proposed $g(n)$ is the following:

$$g(n) = Max\{\alpha(\frac{E_{res}(n)}{E_{ini}(n)}) + \beta(\frac{N_r(n)}{N_t(n)}) + \delta(\frac{B_f(n)}{B_{ini}(n)})\} \tag{3.14}$$

where $E_{res}(n)$ and $E_{ini}(n)$ are residual and initial energy of node n respectively. Moreover, $N_t(n)$ and $N_r(n)$ are the number of transmitted and received packets respectively. $B_f(n)$ and $B_{ini}(n)$ referred to the number of free and initial buffer of node n respectively. The sum of the weight parameters α, β and δ is $\alpha + \beta + \delta = 1$.

As we discussed before routing algorithms which use aggregated cost functions help to solve problems easier but they may lead to paths which violate some constraints.

3.4.3 Fuzzy Logic

Fuzzy Logic combines a set of metrics to select best neighbours and forward data towards the destination. The values obtained from the metrics were used as inputs to the fuzzy logic controller in measuring quality of neighbour nodes. A fuzzification process is performed in several stages according to the number of metrics that are taken into account. In every fuzzification step two metrics are combined and the QoS comes out. A value is assigned to each metric which indentifies its quality. The defuzzification process defines that all metrics have been considered

and an average quality value is assigned to each neighbor. In [27], the authors use the fuzzy logic to combine 3 metrics: delay, ETX and energy. Although fuzzy logic is an interesting approach, assigning different values to different metrics in a fuzzy logic controller, whice defines the importance of each metric, is however a difficult point.

3.4.4 Energy Aware Routing

The energy aware routing protocols aim at maximizing the network lifetime and balancing the energy consumption.

EEPR

The authors of [38] propose an Energy-Efficient Probabilistic Routing algorithm (EEPR) for the IoT. EEPR is implemented under the context of the AODV protocol. In the typical AODV protocol, each node that receives a RREQ packets forwards it to their one-hop neighbors. On the other hand, EEPR is calculating the forwarding probability and decides stocastically the next hop.

The routing metrics for probabilities is a combination of the ETX metric and the residual energy of a node. Defining the $ETX_{i-1,i}$, as the ETX value between node $i-1$ and node i, ETX_{max} the maximum ETX value that a link may have, and E_i the residual energy of node i and E_{max} the maximum residual energy of node i, the forwarding probability p of node i is determined by:

$$p = \Big[p_{min} + E_i A \Big[1 + \frac{(ETX_{i-1,i} - ETX_{max})}{(1 - ETX_{max})}\Big]\Big]^{1/a} \tag{3.15}$$

where p_{min} is the predefined minimum probability, a the weighted factor for variation of the forwarding probability and A is described as:

$$A = \frac{1 - p_{min}}{2E_{max}} \tag{3.16}$$

where E_{max} is the maximum residual energy of node i.

Using the above probability formulation, nodes with lower residual energy and ETX have lower forwarding probability.

Energy Efficient Routing

The authors of [34] proposes an Energy Efficient routing protocol based on AODV considering two different energy cost metrics, the remaining energy capacity and the transmission power. During route discovery from the source to the destination the energy values along the route are accumulated in the RREQ packets. At the destination or the intermediate nodes these values are copied into the RREQ packets which are transmitted back to the source. The source alternates between the maximum remaining energy capacity route and minimum transmission route every time it performs route discovery.

Energy Aware Routing

On the other hand, the authors of [31] propose an energy aware routing protocol which uses the remaining energy capacity and the link quality to prolong the network lifetime. It uses the mechanism of route discovery for path setup as AODV and route request and route reply

messages. The suitable forwarder is the one with remaining energy above the *Required Energy Threshold* and with the maximum Link Quality, which decreases the retransmissions. In the same time, the proposed protocol minimizes packet collisions through a back-off delay scheme, where route request messages are not broadcasted immediately, but after a back-off timer.

3.4.5 Discussion

Routing algorithms are major factors in the performance of routing protocols. The purpose of a routing algorithm is to make decisions for the router concerning the best paths for data and the routing protocol determines which routing algorithm is used in the network. Since the routing algorithms have so big impact on the overall performance of the network, we should focus on using efficient routing algorithms. Although there are many routing protocols in the literature, what it lacks is a theoretical background or efficient routing algorithms behind them. Many protocols are based on shortest path routing algorithms and their proposals are based on changing some parameters e.g. link or path costs. Other protocols propose transmission power control, choosing higher reliability links or maximum energy capacity paths. What is important is to bridge the gap between theory and practice, in other words between proved efficient routing algorithms and routing protocols.

3.5 Network Flow-based Routing

Some existing routing protocols are using different approaches such as network flow and QoS. Route setup is modeled and solved as a network flow problem. The objectives can be maximizing the network lifetime, energy balancing between the nodes or finding the bounds of network lifetime.

3.5.1 Maximum lifetime energy routing

Chang and Tassiulas [47] model the maximum lifetime problem. Through a network flow model they find out the number of packets that can arrive to the sink until the first node fails, maximizing the network lifetime. Then they solve the maximum residual energy path problem. The link cost is defined as a function of remaining energy E_i of node i in V, where V is a set of nodes. The required trasmission energy using the link $\{ij\} \in I$ is e_{ij}, where I is a set of arcs between the nodes. Authors use two different versions of link cost c_{ij}:

$$c_{ij} = \frac{1}{E_i - e_{ij}} \tag{3.17}$$

$$c_{ij} = \frac{e_{ij}}{E_i} \tag{3.18}$$

The least cost paths to the destination are found using the Bellman–Ford shortest path algorithm with the link costs c_{ij}. Those least cost paths have the largest residual energy among all the paths.

3.5.2 Maximum lifetime data gathering

Kalpakis & al. [48] define the problem as a maximum lifetime data gathering problem and model the data routes in sensor networks through an integer linear program. The lifetime of the system is defined as the number of rounds or periodic data readings from sensors until the first

sensor dies. The data gathering schedule specifies for each round how to route the data to the sink. A tree is constructed from the sink to all the nodes for each schedule and the aim is to maximize the lifetime of the schedule. The flow network with maximum lifetime subject to the energy constraints of sensor nodes is proposed and it is called an optimal admissible flow network. Then, the schedule is constructed by using this admissible flow network. This is done using an algorithm that translates the flows into capacity of links. Data aggregation is also proposed in this work to reduce transmitions and thus the energy consumption.

3.5.3 Bounds of lifetime

Several papers deal with the problem of finding the bounds of network lifetime. In [54], a flow optimization model is proposed finding upper bounds of the lifetime of a sensor network that collects data from a specific region using some energy-constrained nodes, assigning also roles to the sensors. A similar approach is proposed in [55] where the authors maximize the lifetime imposing hop count constraints.

3.5.4 Hop-Constrained Energy-Aware Routing

In [49] the authors propose a linear programming model to solve the hop-constraint energy-aware routing in WSNs. They split the lifetime into equal periods of time, named rounds and they solve the problem in each round. The objective function minimizes the maximum energy spent by any node. They propose a flow-based routing protocol which makes use of the flow information. The flow is translated as capacity on the links. Before every round, the base station solves the linear program and transmits the routing information to the sensor nodes.

3.5.5 Discussion

Flow-based routing is an interesting approach since it allows to consider multiple objectives through a polynomially solvable network optimization problem to determine the routing information. Nevertheless, most of the above proposals do not specify the way that this flow information can be used in a routing process. Even if [49] do propose a routing protocol, none of the above papers consider a MAC duty-cycle mechanism, which however allows to save the main part of energy.

3.6 Conclusion

In this chapter, we have presented existing routing protocols for WSNs, classsifying them according to their QoS metrics. We have considered three main categories: single metric, multiple metric and network flow-based routing protocols.

Although there are many routing protocols in the literature as we have seen above, which consider many different QoS metrics, what is really missing behind these protocols is the theoretical background. Most of the protocols propose techniques or change network parameters and they prove their efficiency compared to existing approaches only through experimental results (mainly simulations). But, when theory is applied into practice it is always a safe way to propose efficient solutions. A routing protocol which is based on an efficient routing algorithm is more stable and likely to succeed. This is what we will present in the next chapters, our contributions, which try to brigde the gap between theory and practice or between routing algorithms on graphs and routing protocols in networks.

Part II

Contributions

Chapter 4

Optimal Probabilistic Energy Aware Routing Protocol

Sommaire

4.1 Introduction

In the previous chapter we have discussed about the importance of network lifetime in WSNs. Sensors nodes have often limited battery, and recharging operations, if at all possible, substantially increase the cost of service.

WSNs have many applications on healthcare. Figure 4.1 shows a typical scenario of a smart home equipped with different wireless sensor nodes, eventually enriched by an assistant robot, for permanently monitoring and assisting the person's daily life activities and his/her personal health state evolution [28]. Data are collected to a sink either periodically (e.g. temperature, humidity, luminosity) or following a specific event (e.g. door opening/closing, chair and bed pressure or movement detection). The data transmission reliability is enhanced thanks to a multi-hop mesh topology.

When deploying a WSN in such a scenario, one of the most important requirements is to be able to estimate the network lifetime. To this aim it is necessary to have an estimate of the data

transmission demands, as the frequency of periodic data sources or the average packet generating rate of the event-triggered sensor sources in a controlled environment like a smart home. Many issues intervene in the design of good policies to manage these WSNs, such as the ability to extend the expected lifetime of the full network, to take into account worst-case scenarios and identify the most critical sensors, to efficiently use batteries, to balance sensor usage and so on.

In this chapter, we propose an energy-aware flow-based routing protocol for maximizing the network lifetime, while still fulfilling the data collection requirements. We propose the Optimal Probabilistic Energy Aware Routing (OPEAR) protocol that combines a mathematical model to propose optimal probabilities in an offline computation phase and a simple, yet energy efficient, routing policy during realtime operations. We implemented a prototype of the OPEAR protocol in the Contiki emulation environment above ContikiMAC duty-cycled protocol, using a stress-test computational campaign. We compared OPEAR with the popular EAR [33]. We implemented EAR over ContikiMAC to have a fair comparison. The computational tests reveals that OPEAR clearly outperforms EAR, yielding substantially higher network lifetimes.

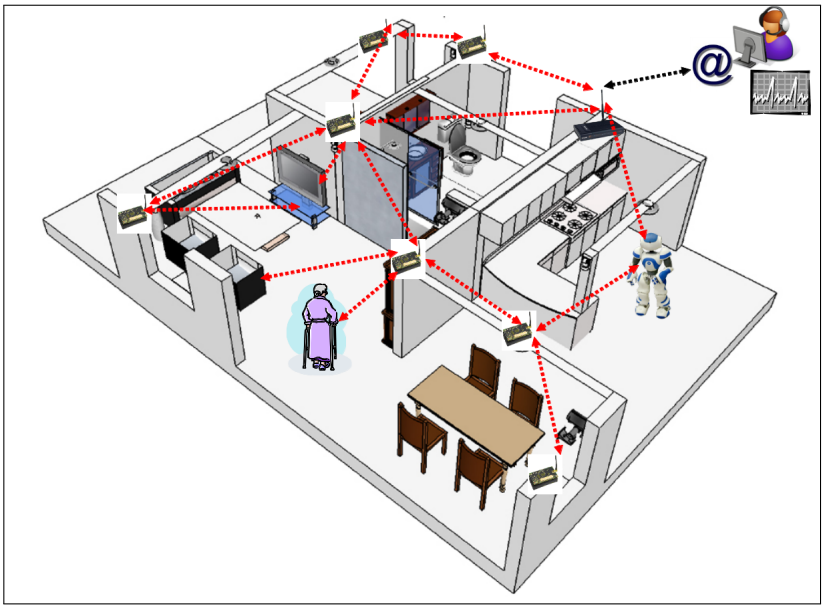

Figure 4.1: A smart home scenario.

The rest of the chapter is organised as follows: The problem description is presented in section 4.2. In section 4.3 the mathematical model is detailed and in section 4.4 the OPEAR protocol is outlined. Then, a realistic stress-test dataset is introduced and our experiments, simulations and results are described in Section 4.5. Conclusions and perspectives are briefly presented in Section 4.6.

4.2 Problem description

Let $G(N, A)$ be the graph representing a WSN, where the set of nodes N represents the sensor and relay nodes, and the set of arcs A represents the wireless connections between the sensor nodes: the arc (i, j) exists if node i can communicate with node j. We denote by $\tau \in N$ the sink node, collecting sensor data from all the network. For any node $i \in N \setminus \tau$ we denote by q_i

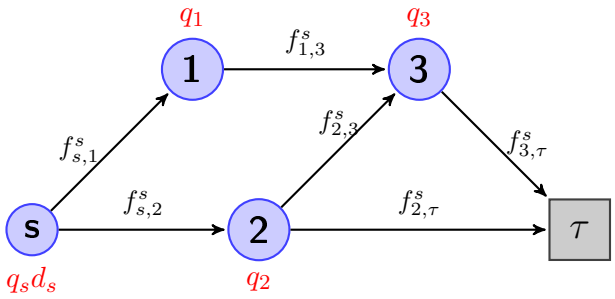

Figure 4.2: An illustrating example.

the energy of sensor i at the beginning of planning horizon. We distinguish two types of nodes except the sink: source nodes and relay nodes. Relay nodes have only routing capability. We denote $N_s \subset N$ the set of source nodes, that are the sensors that collect data to be sent to the sink node. A source node can also receive and forward packets from another source or relay node, so we assume that source nodes perform also like routers. We represent data sent from sensors as a set of demands D'. For each demand $i \in D'$, o_i represents the sensor collecting the data (origin of the demand) and d_i the expected amount of data collected by sensor within the planning horizon. The destination of all the demands is the sink τ. We denote T as the planning horizon.

We denote the decision variables $f^k_{i,j}$ which represent the number of packets whose source is k to be sent from i to j (and symmetrically, the number of acknowledgment messages returned from j to i).

A single-source example is reported in Fig. 4.2, with our notation. In red are reported the initial energy and (for sources only) the demand of each node.

We deal with the problem of lifetime optimization in WSNs. We focus on network and MAC layer energy saving techniques, so we solve the routing problem with energy constraints through a probabilistic approach. We consider a concrete duty-cycle MAC protocol (contikiMAC) in our optimization model, while the existing works only focus on energy balancing at routing level, without explicitly consider the duty-cycle, which however allows to save the main part of energy.

4.2.1 Assumptions

Our strategy is composed of two steps: an offline planing phase and a real-time routing phase.

In the offline planning phase, probabilistic routing tables, one for each sensor, are computed by a centralized device. In details, in such a planning phase we assume that:

- the network connecting sensors is static in terms of links (this is almost always the case for a small size home network where nodes are static and external interference is mitigated by the physical layer's channel hoping technique),
- a unique sink exists in the network (the sink is main-powered without energy limitation), representing the destination of all packets. This assumption can be relaxed without any impact on the model if we assume that multiple sinks can communicate with a wired network or the information gathered by one sik are used only by this sink,
- other sensor nodes are battery-powered with a given initial energy budget,

Symbol	**Meaning**
G	finite graph
N	set of nodes
N_s	set of source nodes
A	set of arcs
D'	set of demands
(i, j)	arc between node i and j
τ	sink node
q_i	initial energy of node i
r_i	residual energy of node i
d_i	demand of node i
o_k	origin of demand k
T	planning horizon
$f(i, j)$	flow from node i to j
$f_{i,j}^k$	flow from node i to j from source k
$f_{i,j}^{k\star}$	optimal flow from node i to j from source k
ρ_1	transmission energy
ρ_2	reception energy
ρ_3	energy spent for sleep mode and CCA
z	difference of residual energy
v	minimal residual energy
y	total consumed energy

Table 4.1: Summary of symbols of Chapter 4

- the planning module has full knowledge of the network (battery level of each sensor and radio links among sensor nodes. This information can be obtained for instance using MPIGate gateway [28].),
- each sensor has local knowledge of the network, consisting in two sets of links, leading to forward and backward sensors respectively, that is those sensors which are respectively nearer to and farther from the sink in terms of minimum number of hops to reach it (this can be the case of a DODAG as defined in IETF RPL protocol, where each node has its own rank),
- the estimated demand is known, representing for each sensor the number of packets that must be routed in the given planning horizon.

At the end of the offline planning phase the routing tables are transmitted to all sensors by means of a broadcasting operation. In particular, each sensor will receive a vector p, having one entry p_i for each of its forward neighbors. In the real-time routing phase each sensor has only a local view, holding its own routing table and forwarding packets to its forwarder neighbors according to the probabilities specified therein.

4.3 System Modeling

Now we present the core model used in the planning phase. Firstly, we consider just the problem of finding a feasible routing for all demands D'.

The *routing energy aware feasibility problem* can be formulated as follows:

$$\sum_{\substack{j\in N:\\(i,j)\in A}} f^k_{i,j} = \sum_{\substack{j\in N:\\(j,i)\in A}} f^k_{j,i} \qquad \forall k \in D', \forall i \in N \setminus \{\tau, o_k\} \tag{4.1a}$$

$$\sum_{\substack{j\in N:\\(i,j)\in A}} f^k_{i,j} = d_k \qquad \forall k \in D', i = o_k \tag{4.1b}$$

$$\sum_{\substack{i\in N:\\(i,\tau)\in A}} \sum_{k\in D'} f^k_{i,\tau} = \sum_{k\in D'} d_k \tag{4.1c}$$

$$r_i = q_i - \sum_{k\in D'} \Big(\sum_{\substack{j\in N:\\(i,j)\in A}} \varrho_1 f^k_{i,j} + \sum_{\substack{j\in N:\\(j,i)\in A}} \varrho_2 f^k_{j,i} + \varrho_3 T \Big) \qquad \forall i \in N \setminus \{\tau\} \tag{4.1d}$$

$$f^k_{i,j} \geq 0 \qquad \forall k \in D', \forall (i,j) \in A \tag{4.1e}$$

$$r_i \geq 0 \qquad \forall i \in N \setminus \{\tau\} \tag{4.1f}$$

where we added the convenience variable r_i to represent the residual energy of sonsor node i. Constraints (4.1a), (4.1b), (4.1c) impose flow conservation. Each received packet must be forwarded (4.1a). Constraints (4.1b) and (4.1c) impose respectively that all packets of a given sensor/demand are forwarded by its source node and all packets are received by the sink node. Constraints (4.1d) link the energy consumption variable values to the number of packets sent and received; ϱ_1 represents the energy spent in each data packet transmission, ϱ_2 the energy spent in each data packet reception and ϱ_3 represents the fixed energy spent per second during the sleep mode and the CCA mechanism (the node performs periodically two quick wake ups

to detect if there is an incoming transmission). Since the planning horizon T is known, the expected number of CCAs and the sleep mode time can be calculated and the respective energy can be subtracted. So, the remaining energy of each node is equal to the initial energy minus the energy spent for transmission, the energy spent for reception and the energy spent during the sleep mode and CCA mechanism. The parameters ϱ_1, ϱ_2 and ϱ_3 give flexibility to the model for being used by any radio duty cycling protocol. Furthermore, ϱ_1 and ϱ_2 allow to incorporate the energy consumed by acknowledgements.

It can be argued that the planning horizon must not exceed the lifetime of the network. We will show in section 4.5 how a good estimation of the lifetime can be obtained and used to set the value T.

4.3.1 Objective Function

The key objective in WSNs is to increase the network lifetime and the energy is consumed by sensor or relay nodes. Many works address the problem of minimizing the total energy consumption. We will show that this is not always the best strategy to achieve this goal.

Let us consider different objective functions and study their effect on routing and energy distribution on a small numerical example. First we complete the above model with the following equations.

$$r_i - r_j \leq z \qquad \forall i, j \in N \setminus \{\tau\} \tag{4.2a}$$

$$v \leq r_i \qquad \forall i \in N \tag{4.2b}$$

$$\sum_{k \in D'} \Big(\sum_{\substack{j \in N: \\ (i,j) \in A}} \varrho_1 f^k_{i,j} + \sum_{\substack{j \in N: \\ (j,i) \in A}} \varrho_2 f^k_{j,i} + \varrho_3 T \Big) \leq y \qquad \forall i \in N \setminus \{\tau\} \tag{4.2c}$$

where variable z represents the maximum difference of residual energy between two sensors, variable v represents the minimum residual energy of a sensor in the network and variable y represents the maximum consumed energy. Constraints (4.2b) and (4.2a) ensure consistency between r_i and v and r_i and z variables, respectively.

We can now introduce different objective functions:

1. minimize total consumed energy (TE) - $min\ y$
2. maximize the minimal residual energy of nodes (mRE) - $max\ v$
3. minimize the difference between the residual energy of nodes (ΔRE) - $min\ z$

Example 4.3.1. Let us consider the graph in Figure 4.3, where there are 10 nodes, node 1 is the sender and τ is the sink. The sender sends 8 units of demand, using the solution of the routing problem 4.1a - 4.1f, using the 3 different objective functions. Nodes 2-9 are relay nodes and they all have the same initial energy equal to 5. Let us suppose that the sender has enough energy to send the whole demand. For simplicity we use an energy model in which one unit of energy (equal to one unit of flow) is spent for every tranmission/packet.

In Figure 4.4 we can see the residual energy of all nodes after sending all 8 units of demand. In table 4.2 the total consumed energy and the maximal difference of the residual energy between the nodes for every different objective function are reported.

We observe that, as expected, the objective function TE brings the minimal total consumed energy, but after sending all demands some nodes have run out of battery (residual energy equal

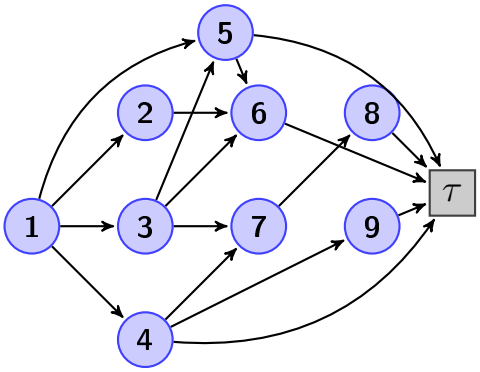

Figure 4.3: Graph Example

Objective Function	TE	mRE	ΔRE
Total Cons. Energy	28	32	33
Max. Res. Energy Diff.	4	0	0

Table 4.2: Total consumed energy and maximal residual energy difference.

to 0). On the other hand, objective functions mRE and ΔRE bring similar results, consuming more energy than objective function TE but we observe that at the end of the planning all nodes are still alive. We recall that the lifetime is defined as the time period where all the nodes are still alive. Therefore, objective functions mRE and ΔRE seem to be more suitable for our goals, leading to a more distributed consumption of energy in the network.

Example 4.3.2. Let us consider another numerical example, where energy on nodes is not uniformly distributed. In case one node is "sensitive" having low initial energy compared to the others, it is interesting to evaluate objective functions mRE, ΔRE and the combination of them to see if they can provide robust solutions. Let us consider the graph of Figure 4.3 where all nodes have initial energy equal to 5 except node 6 which has initial energy equal to 1. The demand is equal to 6. As we can observe in Figure 4.5, objective functions mRE and ΔRE provide the same solution, keeping the network alive after the whole planning. The combination of both of them having a weighted factor 0.5 gives slightly better results reducing the consumed energy. Both objective functions are partially able to increase the lifetime but their combination seems the safest way to provide a more robust solution. A certain level of robustness of the offline solution calculation is a key factor for any probabilistic protocol. In fact, the routing solutions obtained are not statically applied, but they are used to compile routing tables which are used in a probabilistic manner. In Section 4.5 we will show that the intuition we got from these small numerical examples is supported by numerical experiments on larger instances.

For that reason, our objective function will consider a convex combination of the two potentially conflicting objectives: maximize the minimum residual energy and minimize the maximum difference of residual energy between sensors.

Objective Function	mRE	ΔRE	mRE+ΔRE
Total Cons. Energy	24	24	20
Max. Res. Energy Diff.	4	4	4

Table 4.3: Total consumed energy and maximal residual energy difference.

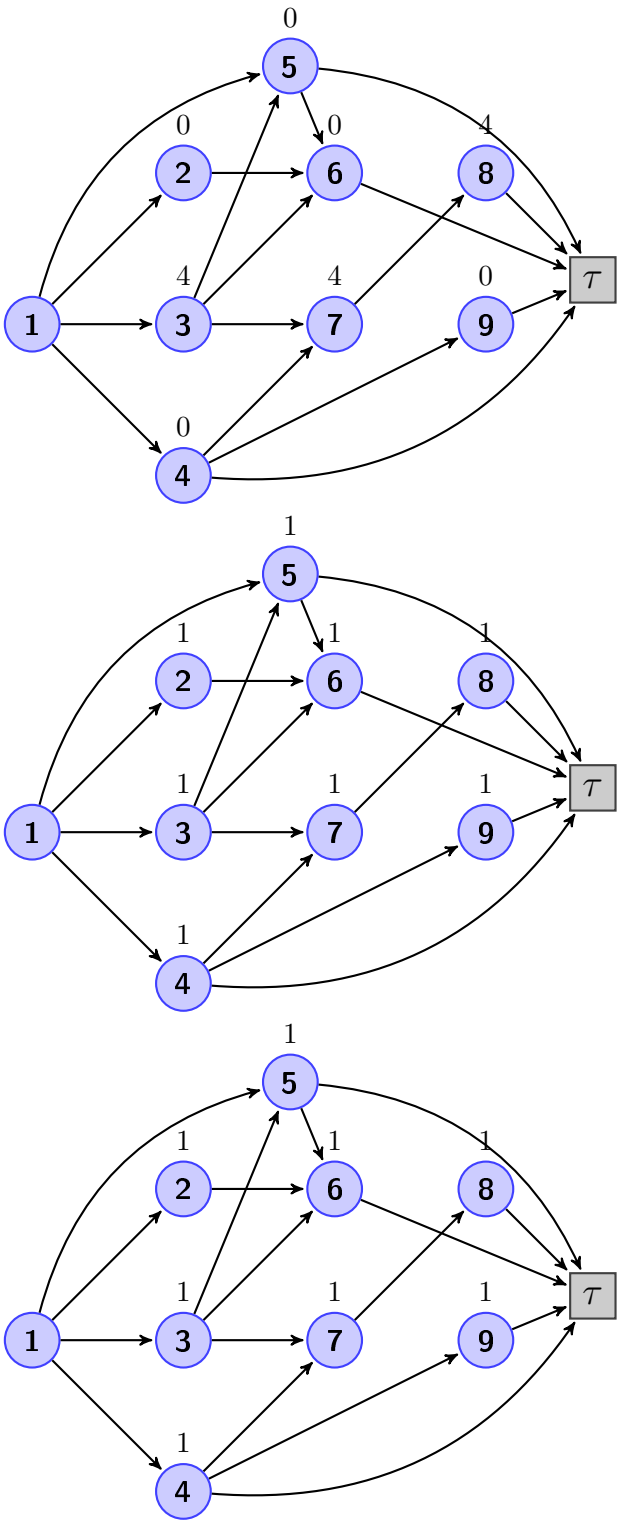

Figure 4.4: Residual Energy - case with uniform initial energy: a) Obj. Function TE, b) Obj. Function mRE, c) Obj. Function ΔRE

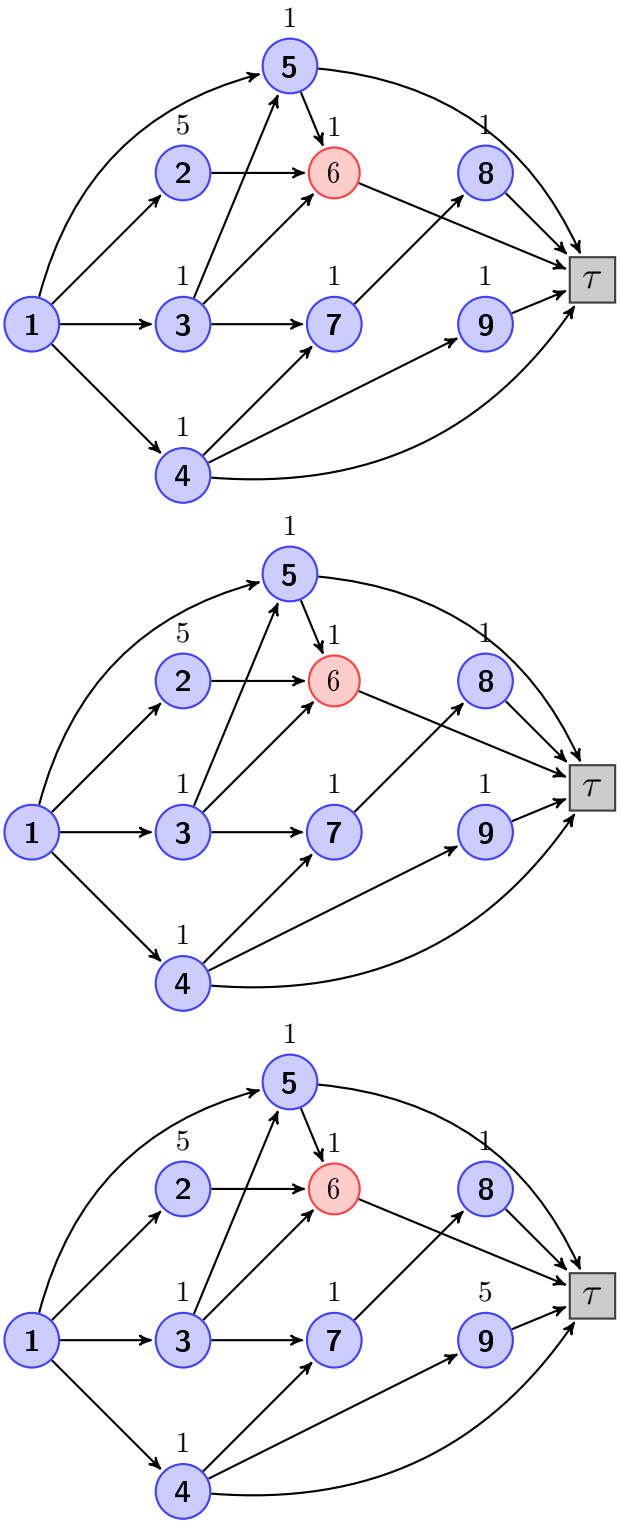

Figure 4.5: Residual Energy - case with not uniform initial energy: a) Obj. Function mRE, b) Obj. Function ΔRE, c) Combination of Obj. Functions mRE and ΔRE

We obtain the *routing energy aware optimization problem*, adding the equations 4.2a and 4.2b to the *routing energy aware feasibility problem* 4.1a - 4.1f, following the objective function:

$$\textbf{max.ze } \gamma(-z) + (1-\gamma)v \tag{4.3}$$

4.4 OPEAR: An Optimal Probabilistic Energy-Aware Routing protocol

The main idea of our OPEAR protocol is to maximize the lifetime of the network by (a) building optimized probabilistic routing tables using a linear programming model and the forecast transmission demands (in terms of number of packets) and (b) limiting the realtime activity of sensors to basic operations, therefor further reducing energy consumption.

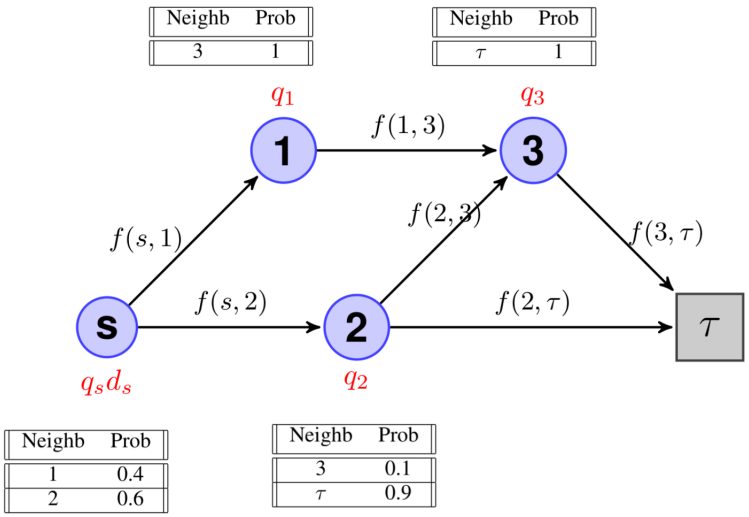

Figure 4.6: An illustrating example.

As an overview, let consider again the network presented in section 4.2. We report in Figure 4.6 a solution in form of routing tables.

The OPEAR protocol can be summarized in two main phases (Fig 4.7).

1. *Routing Table Calculation Phase:* The sink node having full knowledge of the network and an estimate of sensor transmission demands, calculates optimized routing tables. Then, it broadcasts a "hello" packet to the neighbors containing such information, so that each node can suitably fill its forwarding table. The "hello" packet is recursively transmitted from each sensor to its neighbors, eventually reaching the terminal sensors collecting data.

2. *Data Forwarding Phase:* Each source/relay node receives and forwards data packets to a neighbor, that is randomly chosen according to the probabilities in its routing table. Each node receives packets and forwards them according to the same probabilistic policy. For increasing the reliability of the network, each node that receives a packet sends back an acknowledgment message. This procedure is iterated until the data packet reaches the sink node. Retransmissions in case of no-ACK is not taken into account explicitly in the

optimization model, as we assumed that it is negligible in our setting. If an estimation of the packet loss is available, such information can be integrated in the coefficients ϱ_1 and ϱ_2 to take into account retransmission without changing the model. The energy consumption for the acknowledge message is taken into account (the consumed energy of one packet transmission includes the consumed energy for the ACK as well).

Optionally, either at fixed intervals or due to a triggering event, the routing table calculation phase is repeated. The sink gathers information about the remaining energy of all nodes and recomputes optimized routing tables. Then it broadcasts again hello packets for updating the sensor routing tables.

In Figure 4.6 the source node s will send packets with probabilities 0.4 and 0.6 to nodes 1 and 2, respectively. It means that ideally the source sends 40% of packets to node 1 and 60% of packets to node 2. At the operational phase the receiver is chosen using random generation, therefore the distribution of sent packets can be slightly different from the ideal (and optimal) one. Therefore, performing re-optimization, after re-collecting the state of the network, will allow a better behaviour.

In an ideal case the calculation phase would happen after sending each packet, thus the routing tables would be updated according to the new energy distribution. But, in that case the energy consumption would be very high, because of the energy spent in collecting the status of the network and updating the routing tables. Therefore, a trade-off exists between the consumed energy in routing table calculation phase and the energy gained using new optimization routing tables. The demands of the application can define the interval of calculation phase.

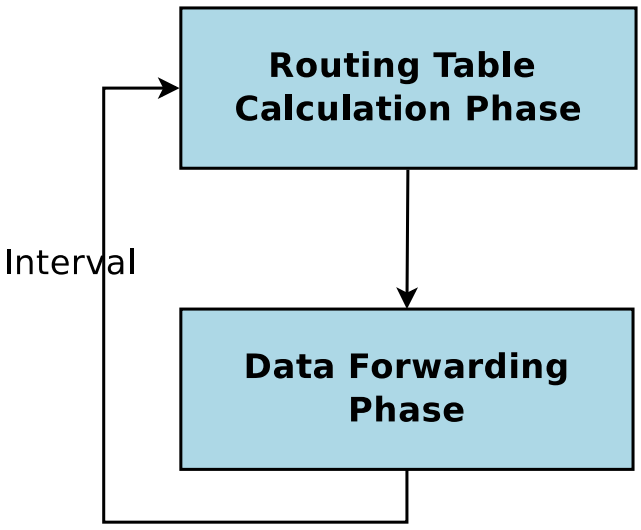

Figure 4.7: OPEAR protocol phases.

4.4.1 Optimal Probabilities

Here we explain how to derive routing probabilities from the solution of the offline optimization phase. Let $f^\star$ be an optimal solution. Routing variable $f_{ij}^{k\star}$ represents the quantity of demand k that uses connection ij, therefore we can calculate the probability transmission of node i to

Mode	Current (mA)
TX	17.4
RX	18.8
Listen	18.8
Sleep	0.001

Table 4.4: Radio current thresholds

node j as follows:

$$P_{ij} = \begin{cases} \sum_{k \in D'} \frac{f_{i,j}^{k,\star}}{\sum_{l:(i,l) \in A} \sum_{k \in D'} f_{i,l}^{k,\star}} & \text{if } (i,j) \in A \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

The probability P_{ij} corresponds to the fraction of demand forwarded to node j by node i.

4.4.2 Energy Model

The radio duty cycle is a critical component of the system with respect to power consumption. From the view of upper layer protocols, the sensors are usually assumed to have four states, i.e. transmitting, receiving, idle listening and sleeping and no energy "cost" for transitions between states [66].

For our experiments we used ContikiMAC radio duty cycle [64], therefore 4 states are taken into account: transmitting, receiving, idle listening and sleeping. We assume that the nodes are synchronized through the phase-lock optimization mechanism. According to the CC2420 datasheet [71], we consider the current thresholds presented in Table 4.4. The energy spent in each state is equal to the current threshold multiplied to the duration of corresponding state. The energy of overhearing (when a node is in the vicinity of a sender may receive and process the message in order to know if it is the destination or not) is a small share of the total consumed energy, and is therefore not taken into account in this work. Nevertheless, it can be easily added to the optimization model modifying ϱ_3 parameter and using overhearing probabilities.

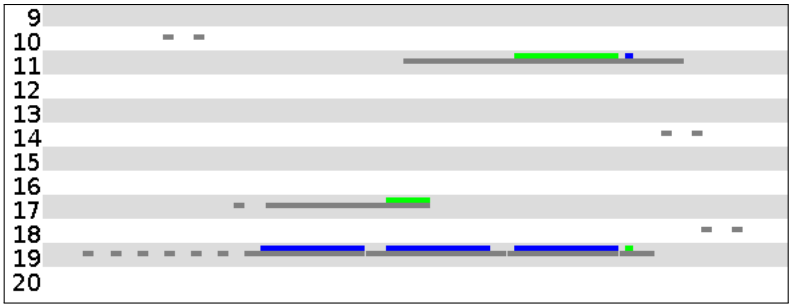

Figure 4.8: Cooja Timeline during Tx, Rx, Idle and Sleep mode.

ContikiMAC

ContikiMAC is a radio duty cycling protocol that uses efficient wake up mechanisms to listen for packet transmissions and allows devices to keep their transceivers off for 99% of the time [64]. If

the packet transmission is detected during a wake-up, the receiver is kept on to be able to receive the packet. The receiver sends then the corresponding acknowledgment (ACK). To transmit a packet the sender repeatedly sends its packet until it receives the corresponding ACK.

The above description is illustrated in Figure 4.8, where a Cooja timeline that depicts all the communication events in the simulation over time is showed. Each line represents a node. Node 19 needs to send a packet to node 11. The small gray lines are ContikiMAC periodically waking the radio up, the blue lines are the transmissions (node 19), the green lines are the receptions (node 11) whereas the ACK is illustrated by the shorter packet. Node 19 sends repeatedly to node 11 till the reception of the ACK. Node 17 overhear while the others inside the range do not wake up during the transmission. Node 17 is not the destination, so it enters in sleep mode. Nodes periodically wake up and check the Received Strength Indicator (RSSI) to give an indication of radio activity on the channel. This is done with the Clear Channel Assessment (CCA) mechanism. Two subsequent CCAs are activated for successful packet transmission. More details about precise timing in ContikiMAC can be found in [64].

ContikiMAC uses also a transmission phase-lock optimization to allow run-time optimization of the energy-efficiency of transmissions. The sender can learn of a receiver's wake-up phase by making note of the time at which it saw a link layer acknowledgment from the receiver. Since the receiver must have been awake to be able to receive the packet, the sender can assume that the reception of a link layer acknowledgment means that the sender has successfully transmitted a packet within the receiver's wake-up window and thus that the sender has found the receiver's wake-up phase.

4.5 Experimental evaluation

We performed an experimental campaign to assess the performances of our OPEAR protocol, comparing it with a EAR [33] by means of tests on emulated networks. We perform an emulation using the Contiki operation system, under Cooja simulator. In the following, we first detail the dataset, we describe our simulation setting and finally we present the results.

4.5.1 Dataset design

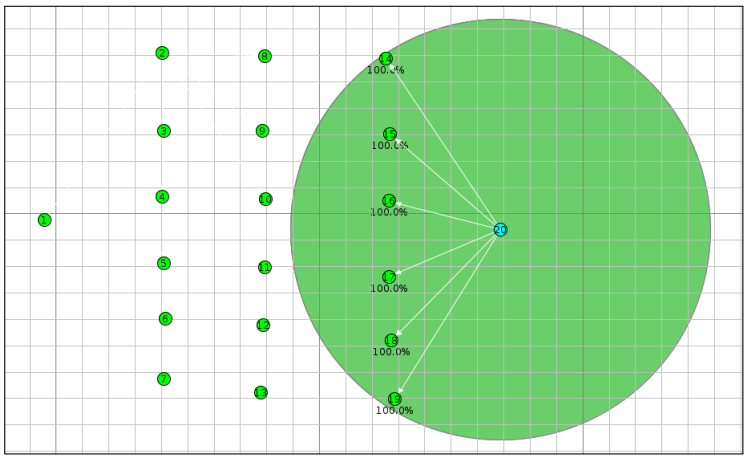

Figure 4.9: Benchmark- 20-node graph.

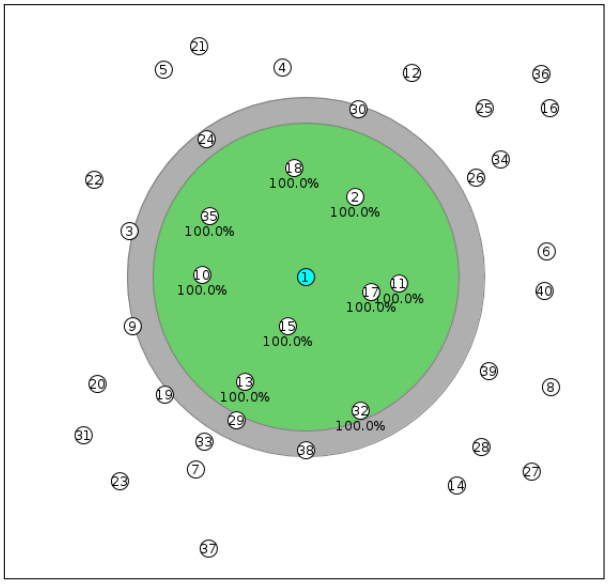

Figure 4.10: Benchmark- Random graph of 40 nodes.

We considered traffic patterns that are periodic. As discussed in subsection 4.2.1 we assume that for a specific time period we can forecast the number of sensor packets that each sensor will send for this specific time period, that may depend on the type of sensor. In a practical setting for the majority of sensor types, like periodic temperature monitoring, estimate the total demand is quite straightforward. For sensors like event-triggered movement sensors, this can be more complicated, but using application-dependent statistics it can be taken into account.

As far as the network architecture is concerned, we use a centralized approach as the majority of existing deployments do. That is each sensor communicates with a central server, either by direct connection or through a communication path of a certain number of hops including relay sensors, that controls the network. This server can in turn be controlled remotely from external entities as well.

We considered different topologies. First, a grid-like topology which is depicted in Figure 4.9. It corresponds to a high network density, in order to run tests in a worse-case scenario. We fixed the number of nodes to 20, which is a common value for a smart-home application. Then we considered three cases: 1) only node 20 is a source, 2) nodes 20, 19 and 14 are sources, 3) all nodes are sources. Node 1 is set as sink and all the remaining nodes that are not sources are relay nodes. Secondly, we considered the random graph depicted in Figure 4.10 of 40 nodes. This allows to test our protocol in a bigger graph considering just the case with all source nodes, which seems to be the most challenging one.

As a stress test, we assumed to be in a scenario in which energy is scarce. To reproduce such a condition, we designed our test instances as follows. First, we generated random energy levels in the ranges [1000-2500] and [2000-3500] Joules for single-source and multiple-source scenarios respectively, drawing values from uniform distributions. Then we computed the maximum demand that can be routed until the first sensor has zero residual energy. To perform such a

computation we use the following linear programming model:

$$\textbf{max } \delta_T$$

$$\sum_{\substack{j\in N:\\(i,j)\in A}} f^k_{i,j} = \sum_{\substack{j\in N:\\(j,i)\in A}} f^k_{j,i} \qquad \forall k \in D', \forall i \in N \setminus \{\tau, k\} \tag{4.5}$$

$$\sum_{\substack{j\in N:\\(i,j)\in A}} f^k_{i,j} = \delta_T \phi_k \qquad \forall k \in D', i = o_k \tag{4.6}$$

$$\sum_{\substack{i\in N:\\(i,\tau)\in A}} \sum_{k\in D'} f^k_{i,\tau} = \delta_T \sum_{k\in D'} \phi_k \tag{4.7}$$

$$r_i = q_i - \sum_{k\in D'} \Big(\sum_{\substack{j\in N:\\(i,j)\in A}} \varrho_1 f^k_{i,j} + \sum_{\substack{j\in N:\\(j,i)\in A}} \varrho_2 f^k_{j,i} + \varrho_3 \delta_T \Big) \qquad \forall i \in N \setminus \{\tau\} \tag{4.8}$$

$$f^k_{i,j} \geq 0 \qquad \forall k \in D', \forall (i,j) \in A \tag{4.9}$$

$$r_i \geq 0 \qquad \forall i \in N \setminus \{\tau\} \tag{4.10}$$

δ_T is the total lifetime, then it is now a variable and it is maximized. Instead of having demand d_k we consider Φ_k as a frequence, therefore $\Phi_k \delta_T$ corresponds to demand in the planning horizon δ_T. Constraints (4.5) - (4.10) have in turn the same meaning of constraints (4.1a) - (4.1f).

Indeed that stress-test model shares features with that of [54]. However, objectives are different: in [54] the demands are known, and the network lifetime is maximized, while to build our instances we aim at finding the maximum amount of packets that the network can transmit before an out of battery failure occurs ($\delta_T \Phi_k \quad \forall k \in D'$).

We created four instances, one for each of the following scenarios:

A random energy levels, single-source sending D' packets

B random energy levels, multiple homogeneous sources, each sending D' packets

C random energy levels, multiple heterogeneous sources, each sending a number of packets uniformly drawn in the range $[0.9D', 1.1D']$.

D random energy levels, all sources heterogeneous, each sending a number of packets uniformly drawn in the range $[0.9D', 1.1D']$.

All the strategies are tested on the grid like graph. Strategy D is used also on the random graph.

4.5.2 Emulation Scenario

We emulated both protocols in Contiki 2.7 [72] using TelosB (also known as TMote Sky). This approach allows us to deploy in sensors the same computer code that would be executed in a real environment. Furthermore, we used the COOJA simulator provided by Contiki [60], whose behavior is close to real hardware, thus yielding very reliable results together with the possibility of rapidly testing multiple scenarios. In Table 4.5 we report our COOJA settings.

Settings	Value
Contiki version	2.7
Wireless channel model	UDG Model with Distance Loss
Communication range	60m
Mote type	Tmote Sky
Communication profile	Rime
MAC Layer	CSMA
Duty Cycle	ContikiMAC

Table 4.5: Contiki OS and Cooja Parameters

4.5.3 Results

We first made a round of tests to assess the scalability of our OPEAR protocol. In particular, we measured the CPU time required to optimize the network flow models for creating the probabilistic routing tables as the size of the network increases. We verified that even problems on large networks can be optimized easily. This was expected, as only Linear Programming optimization is involved in the definition of the probabilistic routing tables. Then, as main performance measure we considered the network lifetime a protocol achieves. As we refered in Section 2.4.1 the network lifetime is defined as the time between the starting of sensor data transmission and the first sensor failure, considering a sensor to fail when its energy reaches zero.

Both protocols compute and assign probability tables during the Routing Table Calculation phase. During the main process, EAR calls the flooding process with a given frequency, while OPEAR can call the flooding process or not. In order to have a fair comparison between EAR and OPEAR, we considered two use settings: *no flooding* (nf), that is the probability tables are computed only once at the beginning of the simulation and never changed, and *with flooding* (kf, where k is the number of flooding steps performed), i.e. new probability tables are re-computed with a certain frequency. This is kept the same for both protocols.

As far as parameters tuning is concerned, according to preliminary experiments we found EAR to perform better in our dataset with $\alpha = \beta = 1.0$. The γ parameter in OPEAR was set to 0.5. Indeed, we could observe that with this setting, utopia values for both objective functions simultaneously could be attained; that was not the case by fixing either $\gamma = 1.0$ or $\gamma = 0.0$.

In Tables 4.6 and 4.7 the parameters for the different simulation experiments are summarized for the grid topology of 20 nodes and for the random topology of 40 nodes respectively. Each column represent an emulation experiment and its use settings. In the second line the cases with different flooding cases are presented. A(nf), B(nf), C(nf) and D(nf) for case of no flooding, A(2f), B(2f), C(2f) and D(2f) for case of 2 flooding processes and A(4f), B(4f), C(4f) and D(4f) for case of 4 flooding processes. In the third and fourth line are reported the sink node and the source nodes, respectively. In the fifth the distribution of demands for the sources nodes. In the sixth and seventh line the number of sent packets and the frequency of flooding (expressed as the number of times the flooding process proceeds during the whole planning period) are reported. We tested the experiments twice using different random initial energy q_1 and q_2 to prove that randomness does not affect the efficiency of our approach.

In Figures 4.11 - 4.14 and 4.15 we report the network lifetime (vertical axis) obtained in each simulation (indicated in horizontal axis), expressed in msec. In each graph the first set of bars corresponds to the test without flooding (nf), the second corresponds to test with two flooding processes (2f), while the third se t corresponds to test with four flooding processes (4f).

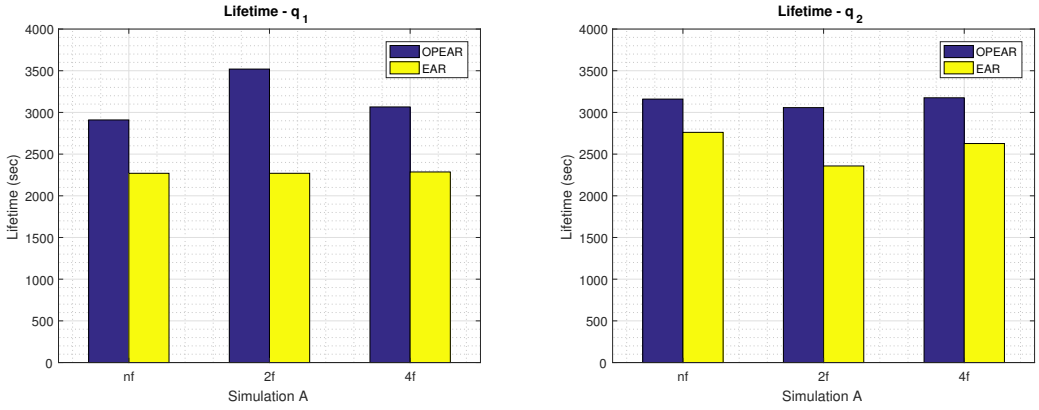

Figure 4.11: Network Lifetime - Grid topology - Exper. A a) q_1, b) q_2

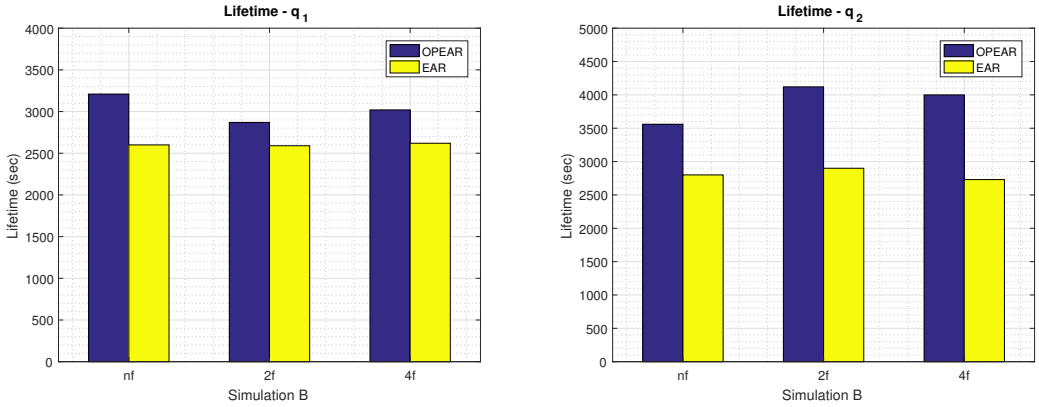

Figure 4.12: Network Lifetime - Grid topology - Exper. B a) q_1, b) q_2

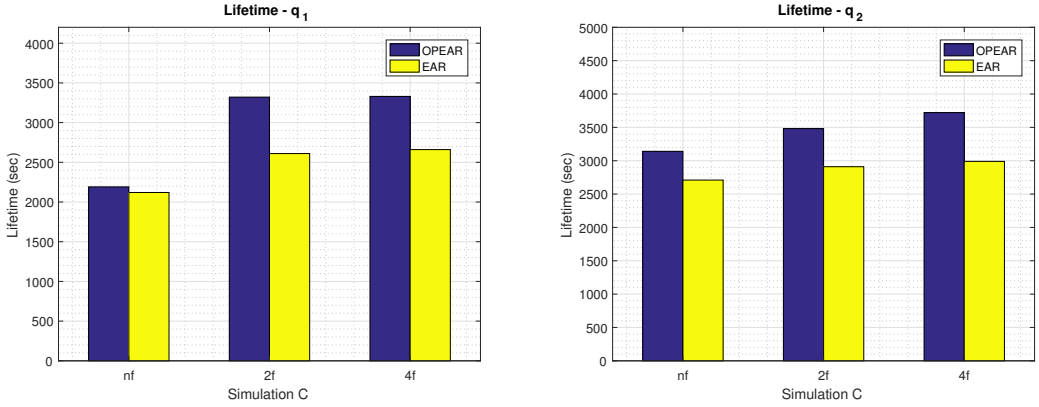

Figure 4.13: Network Lifetime - Grid topology - Exper. C a) q_1, b) q_2

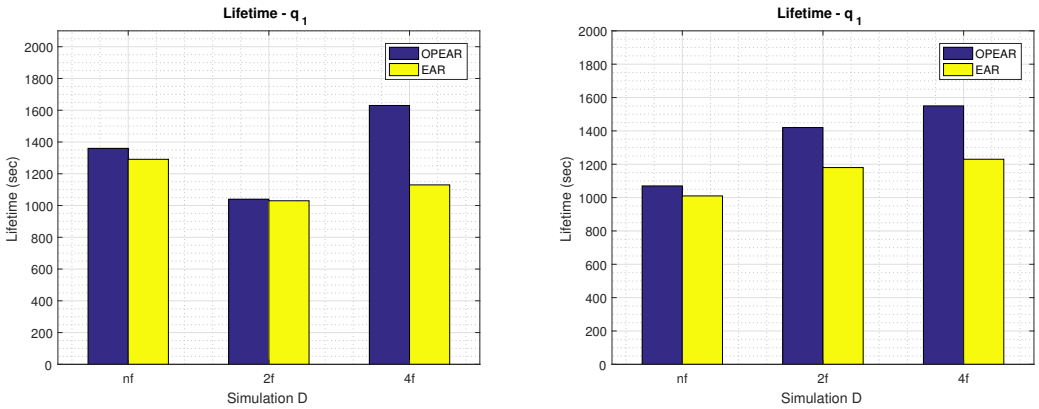

Figure 4.14: Network Lifetime - Grid topology - Exper. D a) q_1, b) q_2

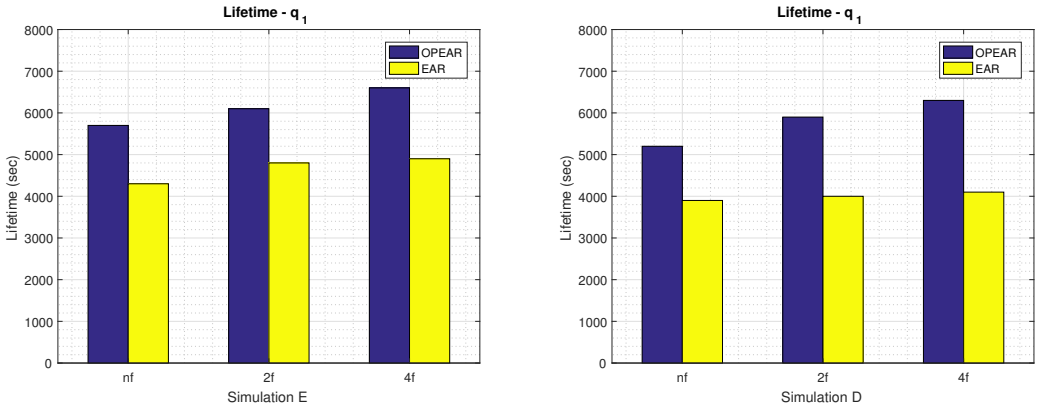

Figure 4.15: Network Lifetime - Random topology - Exper. D a) q_1, b) q_2

Table 4.6: Simulation details - Grid topology - 20 nodes

Exper.	A			B			C			D		
Cases	(nf)	(2f)	(4f)	(nf)	(2f)	(4f)	(nf)	(2f)	(4f)	(nf)	(2f)	(4f)
Sink	1			1			1			1		
Source(s)	20			20,19,14						All		
Demand	Uniform			Uniform			Random			Random		
Packets	5542 - 4495			5965 - 7118			$[0.9, 1.1] \cdot 5832$			$[0.9, 1.1] \cdot 5832$		
Flooding Freq.	—	2	4	—	2	4	—	2	4	—	2	4
Packet Freq. (p/sec)	1.25			1.25			1.25			1.25		
Energy Range	[1000-2500]			[2000-3500]								
Prob. Distribution	Uniform			Uniform			Uniform			Uniform		

Table 4.7: Simulation details - Random topology - 40 nodes

Exper.	A		
Cases	(nf)	(2f)	(4f)
Sink	1		
Source(s)	All		
Demand	Random		
Packets	$[0.9, 1.1] \cdot 5832$		
Flooding Freq.	—	2	4
Packet Freq. (p/sec)	1.25		
Energy Range	[2000-3500]		
Prob. Distribution	Uniform		

Figure 4.11 depicts our results in networks where one single source sends packets to the sink (Experiment A), Figure 4.12 the case where multiple sources send the same number of packets to the sink (Experiment B), Figure 4.13 the case where several sources send different number of packets to the sink (Experiment C), Figure 4.14 the case where all nodes are sources and send different number of packets to the sink (Experiment D), while Figure 4.15 the case where all nodes are sources and send different number of packets to the sink for the larger random graph of 40 nodes.

As a general consideration, we observe that OPEAR outperforms EAR in all simulations. OPEAR generates optimal probabilities, trying to minimize in part the difference of the residual energy of the nodes. This means that the nodes are used in that level so that at the end of the planning horizon they have similar residual energy. Therefore, our objective function prolongs the lifetime of the network. EAR chooses next hops by taking into account the residual energy of the next hop neighbor itself and the residual energy of nodes that may be part of the path till the packet arrives to the sink. In other words, EAR does not find paths with the most energy but sub-optimal paths which may include nodes with low energy. A drawback of EAR is that since the residual energy is normalized by the initial energy (that is a part of the cost function), the first round brings completely uniform probabilities among all the links, so nodes with lower energy are used at same level as nodes with higher energy. Furthermore, EAR has a local vision and do not perform a global optimization on the offline step of OPEAR.

Flooding seems to have a positive influence in OPEAR, proving the capability of OPEAR of re-adapting routing tables to the changed panorama of sensor residual energy. Optimal solution

Table 4.8: Variance of Residual Energy

Exper.	A (nf)	A (2f)	A (4f)	B (nf)	B (2f)	B (4f)	C (nf)	C (2f)	C (4f)	D (nf)	D (2f)	D (4f)
OPEAR	177456	153937	132815	199666	85658	128362	127919	292073	165222	263149	282839	251953
EAR	191083	192443	191684	273326	275197	303051	294316	307627	329706	277622	313203	318152
% reduction with respect to EAR	7.13	20	30.71	26.95	68.87	57.64	56.53	5.06	49.89	5.21	9.69	20.8

let some sensors be used more than others with less energy. On the other hand, EAR does not seem to be improved significantly.We have to remark that the energy spent for flooding is not considered in our experiments for both protocols. By that way, we are able to evaluate the mechanism of probability assignment to the links.

Table 4.8 present the variance of residual energy for both protocols. We can observe that OPEAR provides lower variance than EAR which proves the fact that OPEAR is able to keep more uniformly the residual energy of nodes.

OPEAR aims to balance the usage of sensors along a given planning period, thus improving lifetime. Moreover, OPEAR aims to reduce the possibility of a negative cascode effect due to the complete consumption of the energy of some sensors, therefore enhancing reliability. Our method is based on an approximation of ContikiMAC, predicting the energy that is only spent in CCA and transmitting-receiving phases by assuming ideal phase-lock mechanism provided by ContikiMAC (which is not always the case during the tests). OPEAR was tested in a real setting using Contiki emulation. In the tests, as we used an approximation, some performance degradation with respect to the ideal case is measured, and therefore the real lifetime is shorter then the one computed by the offline model. Nevertheless, OPEAR prolongs significantly the network lifetime compared to the benchmark protocol.

4.6 Conclusion

In this chapter we presented the Optimal Probabilistic Energy Aware routing protocol for routing in duty-cycled WSNs. It is designed to improve network lifetime (calculated as the span of time that all sensors are alive), and therefore reliability, that features one of central importance in several applications, in particular in sensor networks in smart homes for heath care. We compared the performances of OPEAR to those of the popular EAR by means of emulations of realistic networks, considering different scenarios and different use modes. OPEAR protocol showed to outperform EAR in all settings, allowing higher network lifetime. Athough energy issue is the main challenge is WSNs, with ever spreading use, more and more applications require also QoS guarantees. In the next chapter, we will deal with multi-metric QoS routing problem.

Chapter 5

Operator Calculus based Routing Protocol

Sommaire

5.1 Introduction

In the previous chapter we deal with the problem of energy optimisation in WSNs. Routing in wireless sensor networks is fundamentally a multi-constrained problem. On the one hand, nodes in these networks need to handle energy constraints because of their limited energy capacity, but on the other hand an increasing number of applications require additional guarantees, like minimal delay and reliability. Therefore, a good routing protocol needs to select best paths through several existing ones to balance energy consumption and guarantee the application's constraints. As we refered in Chapter 2.4.2, those QoS metrics could have different nature. They could be additive like delay, multiplicative like reliability, *max* or *min* constrained like energy or more generally obey any function.

As we have seen in Chapter 3 the existing protocols choose paths either according to one metric or using a single aggregated function with multiple metrics. RPL [19] proposes several metrics but there is no indication about the way that those metrics can be combined, solving the multi-constrained problem.

In this chapter we make use of the Operator Calculus (OC) theory introduced by R. Schott and S. Staples [11] to solve path selection in the presence of multiple constraints. To the best of our knowledge, OC is the only algorithm which is able to simultaneously consider those different kind of metrics. Based on OC, we derive a distributed algorithm for path selection in a graph and develop a new routing protocol that makes use of this algorithm: the Operator Calculus based Routing Protocol (OCRP).

The rest of this chapter is organized as follows. In the next section we review the OC theory. In Section 5.3, we illustrate through an example the path selection algorithm using OC and derive a distributive version of it. Section 5.4 details the implementation of OCRP in Contiki OS. Section 5.5 provides a performance analysis of OCRP. We conclude the chapter in Section 5.6.

5.2 Operator Calculus on Graphs

5.2.1 Motivating Example and Nilpotent Adjacency matrix

In WSNs we care about solving the multi-constrained problem dealing with all those different QoS metrics, like energy, delay and reliability.

The principle idea underlying the OC approach is the association of graphs with algebraic structures whose properties reveal information about the associated graphs. For example, an element a of a ring on algebra is said to be *nilpotent* if $a^n = 0$ for some positive integer n. By constructing the "nilpotent adjacency matrix" associated with a finite graph, information about self-avoiding structures (paths, cycles, trails, etc.) in the graph are revealed by computing powers of the matrix. Cycles are removed from consideration automatically by the algebra itself.

OC constructs the "nilpotent adjacency matrix" related to a finite graph and solves the shortest path problem by the powers of that matrix, removing the cycles of the paths. OC makes also use of constraints algebra that automatically removes from consideration any path whose weight fails to simultaneously satisfy the constraints (Section 5.2.2).

Example 5.2.1. We make use of the graph example of Fig. 5.1 which is composed of 5 nodes and directed edges. Each edge has a weight vector $\mathbf{w}_{ij} = (w_{ij1}, \ldots, w_{ijm}) \in \mathbb{R}^m$. Delay, link quality and node's residual energy are used as constraints. The constraint vector is $\mathfrak{C} = \{9, 30, 60\}$. The link quality in that case is expressed by *min* constraint in each hop, instead of using a multiplicative constraint in a whole path.

The adjacency matrix of the above graph is represented in Table 5.1.

If we have a weighted graph, we can modify the adjacency matrix adding the constraints in the matrix. We denote by $\xi^{\mathbf{x}} : \mathbf{x} \in \mathbb{R}^m$ the constraint vector of each link and we use the notation $\omega_{\mathbf{u}}$ to represent the path $\mathbf{u}$. The associated *path-identifying nilpotent adjacency matrix* Ψ is represented in the matrix of Table 5.2.

The powers of the above matrix bring all the hop-minimal feasible paths of the graph. Each power evolves the paths one hop till no more feasible path exists.

5.2.2 Constraint Algebra

The operator calculus approach is further extended through the introduction of the *constraints algebra* that automatically removes from consideration any path whose weight fails to simulta-

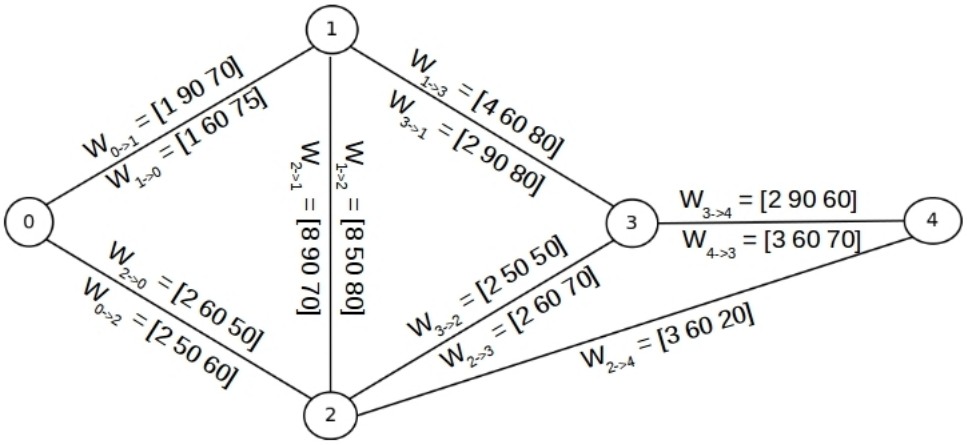

Figure 5.1: A graph example with links having a vector of weights. We observe that links may be unidirectional ($2 \to 4$) and links $i \to j$ and $j \to i$ have different weights.

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Table 5.1: Adjacency matrix associated to the graph of Fig. 5.1

neously satisfy multiple constraints [11]. The resulting constrained path algebra is then used to devise algorithms for computing multi-constrained paths in weighted graphs.

Consider a directed graph $G = (V, E)$ on n vertices such that associated with each edge $(v_i, v_j) \in E$ is a vector of weight $\mathbf{w}_{ij} = (w_{ij1}, \ldots, w_{ijm}) \in \mathbb{R}^m$. For the case $m = 1$, Dijkstra's algorithm finds all single source minimum paths in a directed graph on n vertices with nonnegative edge weights in $\mathcal{O}(n^2)$ time [8]. The Bellman-Ford algorithm finds single source minimal paths in digraphs with nonnegative cycles and arbitrary edge weights and runs in $\mathcal{O}(n\,|E|)$ time [3, 9].

In the more general case $m \geq 1$, Corley and Moon [4] presented an algorithm for finding all Pareto non-negative cycles and paths with computational complexity $\mathcal{O}(mn^{2n-3} + mn^n)$, where m is the constraint vector dimension and n is the maximal hop depth of the graph.

Given an initial vertex v_0 and terminal vertex v_∞ in a weighted graph, the collection of *feasible paths* from v_0 to v_∞ refers to all paths whose associated total costs satisfy some predefined constraints. OC approach aims to find all hop-minimal feasible paths satisfying multiple constraints involving weights that are not necessarily additive.

Definition 5.2.1. Multi-Constrained Path Problem (or MCP problem): Given a finite graph $G(N, A)$, where N is the set of nodes and A is the set of links, each edge $(v_i, v_j) \in A$ is weighted with a vector of m non-negative integers $w_{ij} = (w_{ij1}, \ldots, w_{ijm})$ and a constraint vector $\mathfrak{C} = (c_1, \ldots, c_m)$. The multi-constrained path problem is to find paths $\mathbf{p}$ from source vertex v_0 to

$$\Psi = \begin{bmatrix} 0 & \xi^{1,90,70}\omega_{01} & \xi^{2,50,60}\omega_{02} & 0 & 0 \\ \xi^{1,60,75}\omega_{10} & 0 & \xi^{8,50,80}\omega_{12} & \xi^{4,60,80}\omega_{13} & 0 \\ \xi^{2,60,50}\omega_{20} & \xi^{8,90,70}\omega_{21} & 0 & \xi^{2,60,70}\omega_{23} & \xi^{3,60,20}\omega_{24} \\ 0 & \xi^{2,90,80}\omega_{31} & \xi^{2,50,50}\omega_{32} & 0 & \xi^{2,90,60}\omega_{34} \\ 0 & 0 & 0 & \xi^{3,60,70}\omega_{43} & 0 \end{bmatrix}$$

Table 5.2: Nilpotent adjacency matrix associated to the graph of Fig. 5.1

target vertex v_∞ in the graph G such that

$$\mathrm{wt}(\mathbf{p}) \preceq (c_1, \ldots, c_m) = \mathfrak{C}, \tag{5.1}$$

where $\mathrm{wt}(\mathbf{p})$ is the weight vector of each path $\mathbf{p}$. In other words, MCP problem finds the collection of feasible paths from source to destination whose associated total costs satisfy some predefined constraints.

Definition 5.2.2. Constraints Algebra: The *constraints algebra*, denoted $\mathcal{A}_\mathfrak{C}$, is the real associative unital algebra generated by $\{\xi^\mathbf{x} : \mathbf{x} \in \mathbb{R}^m\}$ with (formal) unit $\xi^\mathbf{0}$ having multiplication defined according to

$$\xi^\mathbf{x}\,\xi^\mathbf{y} := \begin{cases} \xi^{\mathbf{x}*\mathbf{y}} & \text{if } \mathbf{x}*\mathbf{y} \preceq \mathfrak{C}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

Example 5.2.2. Consider $m = 3$: end-to-end delay, reliability and node residual energy. The requirements are the following:

- The total end-to-end delay must be less or equal to $d_{\max}$.
- The link quality of total path must be greater or equal to $lq_{\min}$.
- The remaining energy of each node in the path must be greater or equal to $re_{\min}$.

In this case, $* = (+, \cdot, \min)$, $\mathfrak{R} = (\leq, \geq, \geq)$, and $\mathfrak{C} = (d_{\max}, lq_{\min}, re_{\min})$. If we consider $\mathfrak{C} = (10, 0.6, 50)$, we have the following computations:

$$\xi^{(1,0.7,60)}\,\xi^{(9,0.9,80)} = \xi^{(1+9,0.7\cdot 0.9,\min\{60,80\})} = \xi^{(10,0.63,60)}.$$

On the other hand,

$$\xi^{(1,0.7,30)}\,\xi^{(12,0.9,35)} = 0$$

because $1 + 12 = 13 > 10$ and also because $\min\{30, 35\} = 30 < 50$.

In this example, the multiplicative identity (unit) of$\mathcal{A}_{(10,0.6,50)}$ is formally defined as $\xi^\mathbf{0} := \xi^{(0,1,\infty)}$.

Given a constraint vector $\mathfrak{C} = (c_1, \ldots, c_m) \in \mathbb{R}^m$, properties of the constraints algebra $\mathcal{A}_\mathfrak{C}$ can be used to sieve out the feasible paths from the collection of all paths. The feasible paths can then be ranked and an optimal path can be chosen.

For fixed positive integer n, consider the alphabet $\Sigma_n := \{\omega_i : 1 \leq i \leq n\}$. For convenience, we adopt the following *ordered multi-index* notation. In particular, letting $\mathbf{u} = (u_1, \ldots, u_k)$ for

some k, the notation $\omega_{\mathbf{u}}$ will be used to denote a *sequence* (or *word*) of distinct symbols of the form

$$\omega_{\mathbf{u}} := \omega_{u_1}\omega_{u_2}\cdots\omega_{u_k}. \tag{5.3}$$

Appending 0 to the set Σ_n, multiplication is defined on the words constructed from elements of Σ_n by

$$\omega_{\mathbf{u}}\omega_{\mathbf{v}} = \begin{cases} \omega_{\mathbf{u}.\mathbf{v}} & \text{if } \mathbf{u}\cap\mathbf{v} = \emptyset, \\ 0 & \text{otherwise,} \end{cases} \tag{5.4}$$

where $\mathbf{u}.\mathbf{v}$ denotes sequence concatenation.

One thereby obtains the noncommutative semigroup Ω_n, whose elements are the symbol 0 along with all *finite words* on distinct generators (i.e., finite sequences of distinct symbols from the alphabet Σ_n). Defining (vector) addition and real scalar multiplication on the semigroup yields the semigroup algebra $\mathbb{R}\Omega_n$. This semigroup algebra will be referred to as a *path algebra.*

Theorem 1. Given a multi-weighted graph G on n vertices and a constraint vector $\mathfrak{C} = (c_1, \ldots, c_m)$, let v_0 and v_∞ denote distinct source and target vertices, respectively. If Ψ is the $\mathfrak{C}$-constrained path-identifying nilpotent adjacency matrix associated with G, then the collection of feasible paths $v_0 \to v_\infty$ in G is given by

$$\xi^{\mathbf{0}}\omega_0 \sum_{\ell=1}^{n} \langle v_0|\Psi^\ell|v_\infty\rangle = \sum_{\substack{\text{paths } \mathbf{p}: v_0\to v_\infty \\ \text{wt}(\mathbf{p})<\mathbf{c}}} \xi^{\text{wt}(\mathbf{p})}\omega_{\mathbf{p}}. \tag{5.5}$$

More specifically, feasible paths exist if and only if $\xi^{\mathbf{0}}\omega_0 \sum_{\ell=1}^{n}\langle v_0|\Psi^\ell|v_\infty\rangle$ is nonzero [11]. For the case $v_0 = v_\infty$, one has

$$\langle v_0|\Psi^\ell|v_0\rangle = \sum_{\substack{\text{cycles } \mathbf{p}: v_0\to v_0 \\ \text{wt}(\mathbf{p})<\mathbf{c}}} \xi^{\text{wt}(\mathbf{p})}\omega_{\mathbf{p}}. \tag{5.6}$$

Although the implementation of OC may look like “brute force" if one only thinks in terms of trying all paths and making comparisons before removing infeasible paths, however, the ideal implementation of OC requires no such development and comparisons. In fact, an implementation in an "algebraic architecture" would require none of that. The properties of the algebra automatically kill the cycles and the infeasible paths. Moreover, the approach is flexible – the OC formulations of all problems look alike. We simply change the set of weights and adjust the binary operation. In this thesis, we are mainly interested in finding, among all feasible paths, only the hop-minimal because its practical interest in a WSN.

5.3 Path selection algorithms

In this section, two path selection algorithms are illustrated. In the first one, we suppose that there is a central entity (a server for example) which has a global view of the network and therefore can construct the Ψ matrix of the whole graph. We refer to this version as the centralized path selection algorithm. In the centralized algorithm, the graph is assumed to be static. The second algorithm presented, referred to as the online path selection algorithm, is a distributed version of the first one. In the online algorithm, each node in the graph is assumed to have a partial view of the whole graph (one-hop, two-hop, etc.) and is capable of constructing the Ψ matrix of this sub-graph.

Symbol	Meaning
n	number of nodes of graph G
m	number of metrics/constraints
c_i	weight of metric i
w_{ij}	weight vector of arc (i, j)
$wt(p)$	weight vector of path p
$\mathfrak{C}$	constraint vector
ξ	constraint vector of arc
v_0	initial vertex
v_∞	terminal vertex
u	(partial) path
ω_u	sequence of paths u
Ψ	nilpotent adjacency matrix
$\mathcal{A}_\mathfrak{C}$	constraints algebra
R	operator
Σ_n	alphabet of integer n
d_{max}	maximum delay
lq_{min}	minimum link quality
re_{min}	minimum remaining energy

Table 5.3: Summary of symbols of Chapter 5

5.3.1 Centralized path selection algorithm

The first centralized algorithm considered here is applied to a fixed graph, i.e., the static case. The topology of the entire network is known at all times. Using the result of Theorem 1, Algorithm 1 enumerates all feasible hop-minimal paths from initial vertex v_0 to terminal vertex v_∞. Applying a choice function then allows the user to select a preferred path for the routing.

It is worth noting that OC eliminates at early stage any encountered infeasible path, leading thus to drastically decrease the search space, so largely reduce the total complexity. However, the number of solutions depends on how tight the constraints are. Tighter the constraints, fewer the solutions. Looser constraints give more solutions and require longer runtimes.

Algorithm 1 expresses the centralized algorithm in pseudocode.

Algorithm 1: StaticCentralizedPaths (pseudocode)

input : Source node v_0, target node v_∞, and constrained path-identifying nilpotent adjacency matrix Ψ.

output: Component of row vector $\mathbf{u}$ representing all multi-weighted paths of minimal length from v_0 to v_∞ satisfying the constraint vector.

$\mathbf{u} := \mathbf{v_0}\Psi$
while $[(\mathbf{u} \neq \mathbf{0}\,\mathbf{and}\,\mathbf{u}_{v_\infty} \neq 0)]$ **do**
  $\mathbf{u} := \mathbf{u}\Psi$
end
return $\mathbf{u}_{v_\infty}$

At each iteration of the loop, a partial path is evolved one step. The algorithm terminates

in two cases:

1. The partial paths all ultimately revisit vertices or violate the constraints. In this case the algorithm returns 0.

2. Any feasible path $v_0 \to v_\infty$ is found. At this point, the algorithm returns all existing feasible hop-minimal paths within the frame sequence considered.

Note that replacing the final line of Algorithm 1 by

$$\textbf{return } \beth(\mathbf{u}_{v_\infty})$$

allows one to compute the *preferred* hop-minimal path from v_0 to v_∞. We denote $\beth$ as a choice function to select specific paths.

Given initial vertex v_0, target vertex v_∞, and constrained path-identifying nilpotent adjacency matrix Ψ, the centralized algorithm proceeds as follows.

1. Compute the action of the matrix Ψ on the initial vertex v_0 and prepend the initial vertex to create a row vector, $\mathbf{u}$, containing partial paths of length 1. Simply stated, this is the v_0th row of Ψ with each entry left-multiplied by the element ω_{v_0}. In the pseudocode, this is denoted by $\mathbf{v}_0\Psi$.

2. If target has not been reached (v_∞-component of $\mathbf{u}$ is zero), apply the matrix Ψ to $\mathbf{u}$, thereby appending one step to each partial path.

3. Repeat step 2 until either the target has been reached or all partial paths have self-intersected, in which case $\mathbf{u} = 0$.

4. Return all existing hop-minimal paths to target (zero if no path exists). This information is found in the v_∞ component of $\mathbf{u}$.

Note that as a matrix, Ψ acts on row vectors by right-multiplication.

Example 5.3.1. Applying the algorithm to the graph of Fig. 5.1, the task is to compute all feasible hop-minimal paths from 0 to 4, satisfying the constraint vector $\mathfrak{C} = (d_{max}, lq_{min}, re_{min}) = (9, 30, 60)$, where $* = (+, \min, \min)$ and $\mathfrak{R} = (\leq, \geq, \geq)$. Setting $\mathbf{v}_0 := (\xi^{0,\infty,\infty}\omega_0)$, one-hop feasible paths with initial vertex 0 are recovered by computing the row vector

$$\mathbf{u} = \mathbf{v}_0\Psi = \begin{bmatrix} 0 & \xi^{(0+1),min(\infty,90),min(\infty,70)}\omega_{01} & \xi^{(0+2),min(\infty,50),min(\infty,60)}\omega_{02} & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \xi^{1,90,70}\omega_{01} & \xi^{2,50,60}\omega_{02} & 0 & 0 \end{bmatrix}.$$

All two-hop feasible paths with initial vertex 0 can be found by computing

$$\mathbf{u} * \Psi = \mathbf{v}_0\Psi^2 = \begin{bmatrix} 0 & 0 & \xi^{(1+8),min(90,50),min(80,70)}\omega_{012} & \xi^{(1+4),min(90,60),min(70,80)}\omega_{013} + \xi^{(2+2),min(50,60),min(60,70)}\omega_{023} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \xi^{9,50,70}\omega_{012} & \xi^{5,60,70}\omega_{013} + \xi^{4,50,60}\omega_{023} & 0 \end{bmatrix}.$$

The reader can verify that the second and fifth component of this vector is zero because the paths *{021} and {024}* are infeasible and are removed by the algebra, i.e.,$\xi^{2,50,60}\ \xi^{3,60,20} = 0$ in the constraints algebra. More details about the calculation of $\mathbf{v}_0\Psi^2$ is given below:

- 1^{st} component: $\xi^{1,90,70}\omega_{01}\xi^{1,60,75}\omega_{10} + \xi^{2,50,60}\omega_{02}\xi^{2,60,50}\omega_{20} = 0$. *{010}* and *{020}* are cycles and are set to zero.
- 2^{nd} component: $\xi^{2,50,60}\omega_{02}\xi^{8,90,70}\omega_{21} = \xi^{10,50,60}\omega_{021} = 0$. The delay constraint is not satisfied.
- 3^{rd} component: $\xi^{1,90,70}\omega_{01}\xi^{8,50,80}\omega_{12} = \xi^{9,50,70}\omega_{012}$.
- 4^{th} component: $\xi^{1,90,70}\omega_{01}\xi^{4,60,80}\omega_{13} + \xi^{2,50,60}\omega_{02}\xi^{2,60,70}\omega_{23} = \xi^{5,60,70}\omega_{013} + \xi^{4,50,60}\omega_{023}$.
- 5^{th} component: $\xi^{2,50,60}\omega_{02}\xi^{3,60,20}\omega_{24} = \xi^{5,50,20}\omega_{024} = 0$. The energy constraint is not satisfied.

Respectively, the three-hop paths can be found by

$$\mathbf{v}_0\Psi^3 =$$
$$\begin{bmatrix} 0 & \xi^{(2+4),min(50,90),min(60,80)}\omega_{0231} & 0 & 0 & \xi^{(5+2),min(60,90),min(70,60)}\omega_{0134} + \xi^{(4+2),min(50,90),min(60,60)}\omega_{0234} \end{bmatrix} =$$
$$\begin{bmatrix} 0 & \xi^{6,50,60}\omega_{0231} & 0 & 0 & \xi^{7,60,60}\omega_{0134} + \xi^{6,50,60}\omega_{0234} \end{bmatrix}.$$

In that point, the destination has been reached by the paths *{0234}* and *{0134}* and the algorithm terminates returning the two hop-minimal feasible paths.

The complexity of enumerating all feasible k-paths from v_0 to v_∞ with the operator calculus approach is

$$\mathcal{O}(n \cdot \sharp\{\mathbf{p} \in \mathcal{F}_{v_0} : |\mathbf{p}| \leq k-1\}),$$

where $\mathcal{F}_{v_0}$ denotes the collection of all feasible paths having initial vertex v_0, and $|\mathbf{p}|$ denotes the length (i.e., number of hops) of the path $\mathbf{p}$ [11].

5.3.2 Complexity evaluation of OC algorithm and comparison with SAMCRA

Since the complexity of OC approach depends on the number of feasible paths, we would like to give a numerical insight of the efficiency of the proposed algorthim. For that reason we evaluate OC experimentally and we compared it with SAMCRA, identifying their complexity under the same topology and same demands [7]. SAMCRA is a well-known centralized, multi-constrained algorithm [14]. It terminates as soon as it finds one feasible path between two nodes. Comparing both algorithms in a fair way, OC stops the multiplication process whenever it finds one feasible path.

SAMCRA returns the path which has the minimum non-linear length l that satisfies all the constraints (see Chapter 3.3.1). If we consider a path P with k links: $P = \{e_1, e_2, \ldots, e_k\}$, the length of P is [14]:

$$l(P) = max_{1 \leq i \leq m}(\frac{\sum_{j=1}^{k} W_i^{e_j}}{L_i})$$

where m is the number of weights and L the constraint vector.

SAMCRA is based on two more principles: the $k-$shortest path approach and the non-dominance. The k-shortest path approach is used when more than one possible paths between a source and a destination are needed to be found. In SAMCRA, this approach is applied only to the intermediate nodes, where we store not only one but multiple sub-paths from the source to each intermediate node, only when these paths are non-dominated. In each step, SAMCRA chooses the sub-path with the minimum path length and stops whenever the first feasible path is found.

Example 5.3.2. For comparing SAMCRA with OC, we limit ourselves to additive metrics since SAMCRA considers only this type of constraint. Both algorithms are compared under the randomly generated 128-node graph of Fig. 5.2. A weight vector of 3 additive metrics is associated to each link. 5 different constraint vectors are used that are depicted in Table 5.4 and each of them has three additive constraints, A, B and C. The weight of a path p corresponding to each metric is equal to the sum of weights of its links for this metric and each weight has to be less or equal to the respective constraint ($\mathfrak{R} = (\leq, \leq, \leq)$). The constraint vector C_i represents stricter constraints than the constraint vector C_{i+1}.

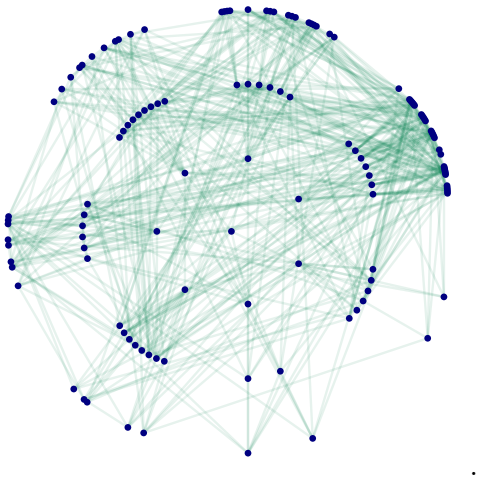

Figure 5.2: Randomly generated 128-node graph.

Table 5.4: Table of constraints

Constraints	A	B	C
$\mathfrak{C}_1$	1.61604	2.41987	68.1328
$\mathfrak{C}_2$	2.83083	4.75579	185.967
$\mathfrak{C}_3$	3.91719	6.85752	330.864
$\mathfrak{C}_4$	4.75373	10.1413	428.139
$\mathfrak{C}_5$	5.07585	14.1619	643.696

Table 5.5: Execution time in msec of 100 (centralized) random requests over a 128-node graph

	$\mathfrak{C}_1$	$\mathfrak{C}_2$	$\mathfrak{C}_3$	$\mathfrak{C}_4$	$\mathfrak{C}_5$
SAMCRA	5262.9	5262.4	5398.3	5449.5	5438.0
OC	278.9	177.6	51.6	44.9	44.6

Table 5.5 depicts the total computation time of 100 random requests of SAMCRA and OC on the graph of Fig. 5.2, using the 5 constraint vectors of Table 5.4. As we can observe, SAMCRA needs more execution time than OC for computing all 100 requests.

Both SAMCRA and OC are exact, so they are able to find a solution, if a solution exists. SAMCRA is Dijkstra-based and this is the main reason that it is not enough efficient in terms of complexity when it is applied in big graphs. Its complexity depends on the number of nodes and the density of the graph. SAMCRA has to explore each unique link of the graph to find the optimal one. On the other hand, OC, using the nilpotent adjacency matrix, needs less time for the graph search process.

Moreover, we can see in Table 5.5 that the execution time of SAMCRA is slightly increased as the constraints are relaxed. Relaxed constraints means that there is more space to explore or more sub-paths can be feasible. However, this is not the case for OC approach. The matrix multiplication process requires almost stable execution time, so more relaxed constraints enables OC to find more quickly a feasible solution. It seems that matrix multiplication is more efficient approach than the exhaustive graph search applied in SAMCRA.

SAMCRA can only work with positive and additive weights. The advantage of OC is its flexibility of the fact that it can work with both positive and negative weights and it can also deal with additive, concave and multiplicative metrics. Moreover, if we take into consideration the numerical results in terms of complexity, OC seems to be a very interesting approach for applications that demand high number of nodes.

Both algorithms have been implemented in Java 1.6.0.27 and computed in Ubuntu 12.04.3 with 2x3 Intel Core 2 Duo processor and 8 GB DDR2.

Example 5.3.3. In this example we evaluate the execution time of OC for finding *all paths* under different constraint vectors for a certain request. We make use of a complete mesh topology, a 10x10 lattice graph [88], so we have a 100-node graph with undirected links. In our graph, node 1 is the source and node 100 the destination. Table 5.6 shows the different constraint vectors $\mathfrak{C}_1$ - $\mathfrak{C}_5$. A, B represent the two different constraints and we consider that they have the same value. Each link has two additive metrics ($m = 2$) and each metric has a weight in the range (0 - 1). The algorithm stops when no more feasible paths exist as all possibilities are explored or excluded.

Table 5.7 shows the execution time of OC in nsec after finding all feasible paths for the request $0 \rightarrow 100$ under the different constraint vectors of Table 5.6.

Table 5.6: Table of constraints - 10x10 Lattice graph

Constraints	A	B
$\mathfrak{C}_1$	9.25	9.25
$\mathfrak{C}_2$	9.5	9.5
$\mathfrak{C}_3$	9.75	9.75
$\mathfrak{C}_4$	10	10
$\mathfrak{C}_5$	10.25	10.25

As expected, more relaxed constraints bring higher number of feasible paths and higher execution time.

Example 5.3.4. In this example we evaluate the execution time of OC for finding *all paths with variable graph size* under the same constraint vector for a certain request. We make use of a 4x4, 5x5, 6x6 and 10x10 lattice graph. Node 1 is the source and node 16 the destination. The 16-node lattice graph is the same for all different size lattice graphs. The other graphs are enlarged as we can see in Figure 5.3 so that the destination node 16 is always in the same position.

Table 5.7: Execution time of OC in nsec, finding all paths for one request over a 10x10 lattice graph

	$\mathfrak{C}_1$	$\mathfrak{C}_2$	$\mathfrak{C}_3$	$\mathfrak{C}_4$	$\mathfrak{C}_5$
Execution Time	972.75	1323.89	2181.26	4438.79	11497.52
Nb. of Paths	4893	9864	15086	20244	84977

Each link has two additive metrics ($m = 2$) and each metric has a weight in the range [0-1]. The constraint vector is $\mathfrak{C} = \{10, 10\}$. Table 5.8 shows the execution time of OC in nsec after finding all feasible paths for the request $0 \rightarrow 16$ under the constraint vector $\mathfrak{C} = \{10, 10\}$.

Table 5.8: Execution time of OC in nsec, finding all paths for one request over different size of lattice graphs

	4x4	5x5	6x6	10x10
Execution Time	14.2	139.62	1055.64	5894.32
Nb. of Paths	184	3883	23933	84246

As expected, bigger graphs bring higher number of feasible paths and higher execution time.

5.3.3 Distributed path selection algorithm

The distributed routing algorithm is implemented at individual nodes. Each node receiving a packet is able to choose a route for passing the packet along, based on some partial information the node has about the current network topology. The distributed algorithm described here makes use of the following assumption.

We assume that each node "knows" its own best-case minimal distance from every other node in the underlying graph. This is the minimum number of hops between a node and every other node in the underlying graph assuming all links are valid. We have to note here that the distributed version of OC is not an online algorithm but a distributed implementation of the centralized OC. Therefore, we can consider that during an initialization phase each node can execute a shortest path algorithm and therefore know the minimal distance to each other node of the network. In practice, a routing protocol sends hello packets with this kind of information and nodes can know the relative distance to the destination (e.g. the RPL rank in a DODAG).

Writing $\mathcal{P}_{ij}$ for the set of all paths $\mathbf{p} : v_i \rightarrow v_j$ in the underlying graph G, let $d : V \times V \rightarrow \mathbb{N}_0 \cup \{\infty\}$ be the function defined by

$$d(v_i, v_j) := \begin{cases} \min\limits_{\mathbf{p} \in \mathcal{P}_{ij}} \{|\mathbf{p}|\} & \text{if } \mathcal{P}_{ij} \neq \emptyset, \\ \infty & \text{otherwise.} \end{cases} \tag{5.7}$$

In other words, $d(v_i, v_j)$ is the minimum number of hops required to reach v_j from v_i among all existing paths in the underlying graph. If no such path exists, $d(v_i, v_j) = \infty$.

In order to determine whether an evolving path is getting closer to the destination, a "distance

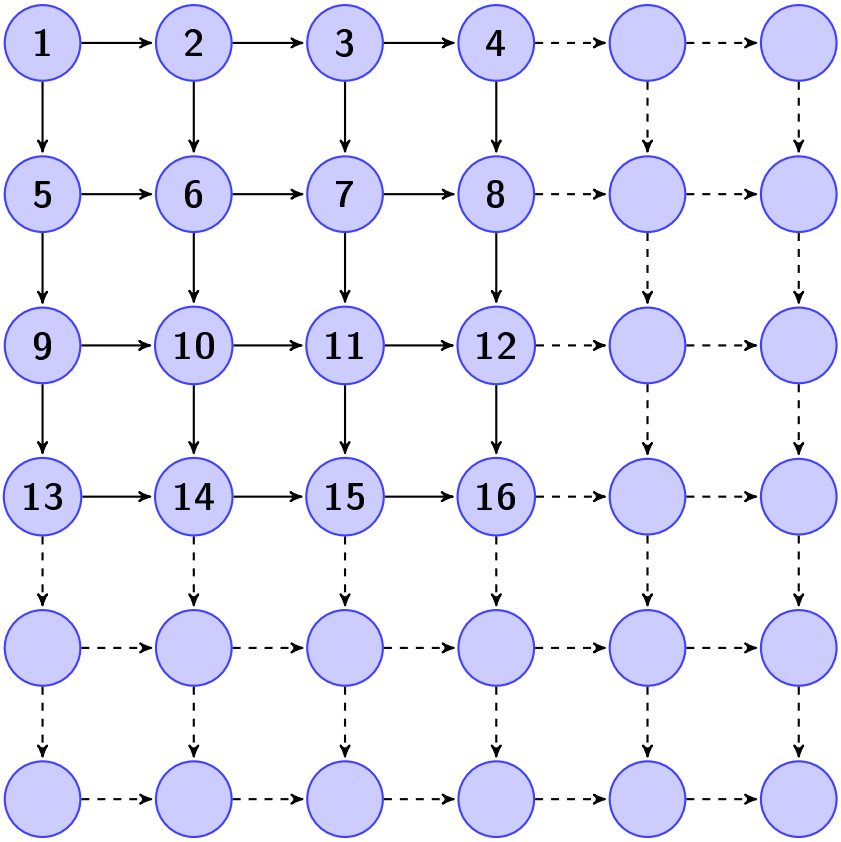

Figure 5.3: Lattice Graph

oracle" function is employed. We define the *distance oracle* $\Delta : (\mathcal{A}_{\mathfrak{C}} \otimes \Omega_n) \times (\mathcal{A}_{\mathfrak{C}} \otimes \Omega_n) \to \mathbb{N}_0$ by

$$\Delta\left(\sum_{\mathbf{p}\in P_1\subseteq\mathcal{P}} \xi^{\mathrm{wt}(\mathbf{p})}\omega_{\mathbf{p}}, \sum_{\mathbf{q}\in P_2\subseteq\mathcal{P}} \xi^{\mathrm{wt}(\mathbf{q})}\omega_{\mathbf{q}}\right) := \min_{\substack{\mathbf{p}\in\mathcal{P}_1\\ \mathbf{q}\in\mathcal{P}_2}} d(\mathbf{p}_{|\mathbf{p}|}, \mathbf{q}_{|\mathbf{q}|}). \tag{5.8}$$

The distance oracle reveals the minimum number of hops required to reach the terminal vertex of a path in $\mathcal{P}_2$ from the terminal vertex of a path in $\mathcal{P}_1$.

Given a finite path $\mathbf{b}$, denote by $\Psi_{(\mathbf{b},t_k)}$ the $\mathbf{b}$-*neighborhood adjacency matrix* at time t_k. This matrix represents the adjacency of the *neighborhood* $\mathcal{N}(\mathbf{b}_{|\mathbf{b}|})$ in the graph at time t_k.

Algorithm 2 is implemented at each node of the underlying graph. It is worthwhile to note that computations are being performed in a sequence of algebras.

For convenience, define the canonical *path projection* $\pi_\Omega : \mathcal{A}_{\mathfrak{C}} \otimes \Omega_n \to \mathcal{P}$ by linear extension of

$$\pi_\Omega(\xi^{\mathrm{wt}(\mathbf{b})}\omega_{\mathbf{b}}) := \mathbf{b}. \tag{5.9}$$

Algorithm 2 simulates the nodewise implementation to choose a preferred multi-constrained path $v_0 \to v_\infty$ in the graph process $(G_{t_k} : k \in \mathbb{N}_0)$. It will be assumed that the time is discretized and the graph sequence is adapted such that topology changes cannot occur during a hop.

Algorithm 2 proceeds by developing multiple paths in parallel with source vertex v_0. At each time step, the partial paths are augmented by one step such that the constraints are still satisfied and the absolute distance (number of hops) from the target is not increased. The algorithm terminates when the time exceeds a final allowable time or when a feasible path $v_0 \to v_\infty$ is found.

Algorithm 2: DistributedChoosePath

input : Source vertex v_0, target node v_∞, initial time t_0, final allowable time t_∞, and neighborhood adjacency matrix $\Psi_{(v_0,t_k)}$ valid at time step t_0.

output: Algebraic element ψ representing a feasible path $v_0 \to v_\infty$ if one is found, zero otherwise.

Initialize frame counter (k), best case distance (in hops) from source to target (δ), constraints, and preferred first step from v_0.

$k := 0$

$\delta := \Delta(\xi^{\mathbf{0}}\omega_{v_0}, \xi^{\mathbf{0}}\omega_{v_\infty})$

$\mathfrak{C} := (\delta, c_1, \ldots, c_m)$

$\psi := \mathfrak{I}\left(\langle \xi^{\mathbf{0}}\omega_{v_0} | \Psi_{(v_0,t_0)} | \mathbf{1}\rangle\right)$

Repeat until maximum frames considered or until remaining distance to target is zero.

while $\left[(t_k \le t_\infty)$ **and** $[\Delta(\psi, \xi^0\omega_{v_\infty}) \neq 0]\right]$ **do**

  Advance frame counter.

  $k := k + 1$

  Extend partial path by one step.

  $\psi := \mathrm{MultiplePathEvolve}[\xi^{\mathbf{0}}\omega_{v_0}, v_\infty, t_k, \Psi_{(\pi_\Omega(\psi),t_k)}]$

end

Return preferred feasible path if one found. Otherwise, return zero.

return ψ

Algorithm 3: MultiplePathEvolve

input : Algebraic monomial $\xi^{\mathrm{wt}(\mathbf{b})}\omega_{\mathbf{b}} \in \mathcal{A}_{\mathfrak{C}} \otimes \Omega_n$, target node v_∞, and discrete time t_k. The element $\xi^{\mathrm{wt}(\mathbf{b})}\omega_{\mathbf{b}}$ represents a multi-weighted partial path (of length $|\mathbf{b}|$) from fixed initial vertex v_0 to vertex $v_{\mathbf{b}_{|\mathbf{b}|}}$. The constraint vector is $\mathfrak{C} = (\delta, c_1, \ldots, c_m)$.

Data: Neighborhood adjacency matrix $\Psi_{(\mathbf{b},t_k)}$ valid at time step t_k is assumed to be available, as this is implemented at each node.

output: Algebraic element ψ representing all feasible multi-weighted paths of length $|\mathbf{b}|+1$ emanating from v_0, satisfying the fixed constraint vector, and whose last steps do not increase the minimal distance to v_∞ over the previous partial path.

Get best-case remaining distance from target.
$\delta' := \Delta(\xi^{\mathbf{b}}\omega_{\mathbf{b}}, \xi^{\mathbf{0}}\omega_{v_\infty})$

Reset constraints so remaining distance cannot increase during path evolution.
$\mathfrak{C} := (\delta, c_1, \ldots, c_m)$

Extend partial path by one step, and return feasible paths.
return $\langle \xi^{\mathrm{wt}(\mathbf{b})}\omega_{\mathbf{b}} | \Psi_{(\mathbf{b},t_k)}$

Example 5.3.5. Let us consider our graph of example 5.1. We assume that each node can see only the one hop neighbors. The neighborhood adjacency matrix of node 0 is equal to:

$$\Psi_{(0,t)} = \begin{bmatrix} 0 & \xi^{1,90,70}\omega_{01} & \xi^{2,50,60}\omega_{02} \\ \xi^{1,60,75}\omega_{10} & 0 & 0 \\ \xi^{2,60,50}\omega_{20} & 0 & 0 \end{bmatrix}$$

The minimum number of hops required to reach the destination is $\delta = 2$ by the path $\{0, 2, 4\}$. The constraint vector of the online algorithm includes δ, so $\mathfrak{C} = \{2, 9, 30, 60\}$, $\mathfrak{R} = (\leq, \leq, \leq, \leq)$, $* = (+, +, \min, \min)$ and $\mathbf{v}_0 := (\xi^{0,\infty,\infty}\omega_0)$. Applying OC in node 0, leads to the following row vector, which represents the extention of the partial path by one step.

$$\mathbf{u} = \mathbf{v}_0\Psi_{(0,t)} = \begin{bmatrix} 0 & \xi^{1,90,70}\omega_{01} & \xi^{2,50,60}\omega_{02} \end{bmatrix}$$

Node 0 knows that 1 and 2 are closer to the destination. Therefore, algorithm 3 selects both nodes as next-hops.

The neighbourhood adjacency operator matrix of node 1 and 2 is given below:

$$\Psi_{(1,2t)} = \begin{bmatrix} 0 & \xi^{1,90,70}\omega_{01} & 0 & 0 \\ \xi^{1,60,75}\omega_{10} & 0 & \xi^{8,50,80}\omega_{12} & \xi^{4,60,80}\omega_{13} \\ 0 & \xi^{8,90,70}\omega_{21} & 0 & 0 \\ 0 & \xi^{2,90,80}\omega_{31} & 0 & 0 \end{bmatrix}$$

$$\Psi_{(2,2t)} = \begin{bmatrix} 0 & 0 & \xi^{2,50,60}\omega_{02} & 0 & 0 \\ 0 & 0 & \xi^{8,50,80}\omega_{12} & 0 & 0 \\ \xi^{2,60,50}\omega_{20} & \xi^{8,90,70}\omega_{21} & 0 & \xi^{2,60,70}\omega_{23} & \xi^{3,60,20}\omega_{24} \\ 0 & 0 & \xi^{2,50,50}\omega_{32} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In the next time step, both node 1 and 2 run OC and choose next valid hops and so on. The final row vector of node 3 is:

$$\mathbf{u} = \begin{bmatrix} 0 & 0 & 0 & \xi^{6,50,60}\omega_{0234} + \xi^{7,60,60}\omega_{0134} \end{bmatrix}$$

We remark that in the distributed version of OC the number of paths may be limited compared to the centralized version because of the parameter δ. In our example, the path $\{0231\}$, presented in the centralized version, is not found by the distributed OC because node 3 cannot send the packet to a node of same δ like node 2. Sender-nodes can forward a packet to nodes which have a strictly less δ and this helps to find paths that are strictly evolved towards the destination. Therefore, we can say that some paths are emitted, but only the "unuseful" ones.

5.4 Implementation of OCRP

OCRP is a multi-constrained routing protocol that applies OC methods to find the set of feasible paths from a source node to a destination. The OC method is implemented in a distributive manner on every node of the network and is applied for every generated and received data packet. OCRP is implemented in Contiki OS [24].

Each path must satisfy three constraints: end-to-end delay, link quality, and nodes' residual energy. The total end-to-end delay must be less than d_{max}, the link quality of each hop must be greater than lq_{min}, and the remaining energy of each node in the path must be greater than re_{min}. Other constraints like throughput can be easily added.

The OCRP implementation is based on the RIME stack as described in the next section.

5.4.1 General description of OCRP

OCRP is implemented on top of the broadcast module and unicast module as illustrated in Figure 5.4. It is independent of the MAC protocol. OCRP implements three different modules, described next.

Neighbor discovery and maintenance This module is responsible for maintaining an up-to-date neighbor table. It is an important feature since all other modules rely on the neighbor table for its functionality. Every node uses either hello messages or data messages to broadcast its internal state to neighbors. A node generates a hello message and broadcasts it if no data is generated or forwarded during the last period of time. A broadcast process is responsible for setting the period's length and handling its expiration within each node. The hello or data packets include the remaining energy of the node. A neighbor node that receives a hello packet computes the link quality using the LQI and RSSI values of the packet.

Delay Estimator This module measures the round trip time (RTT) for every neighbor node. It is called every random amount of time selected within a given interval. This interval is chosen such that the RTT for all neighbors are computed within a reasonable amount of time and

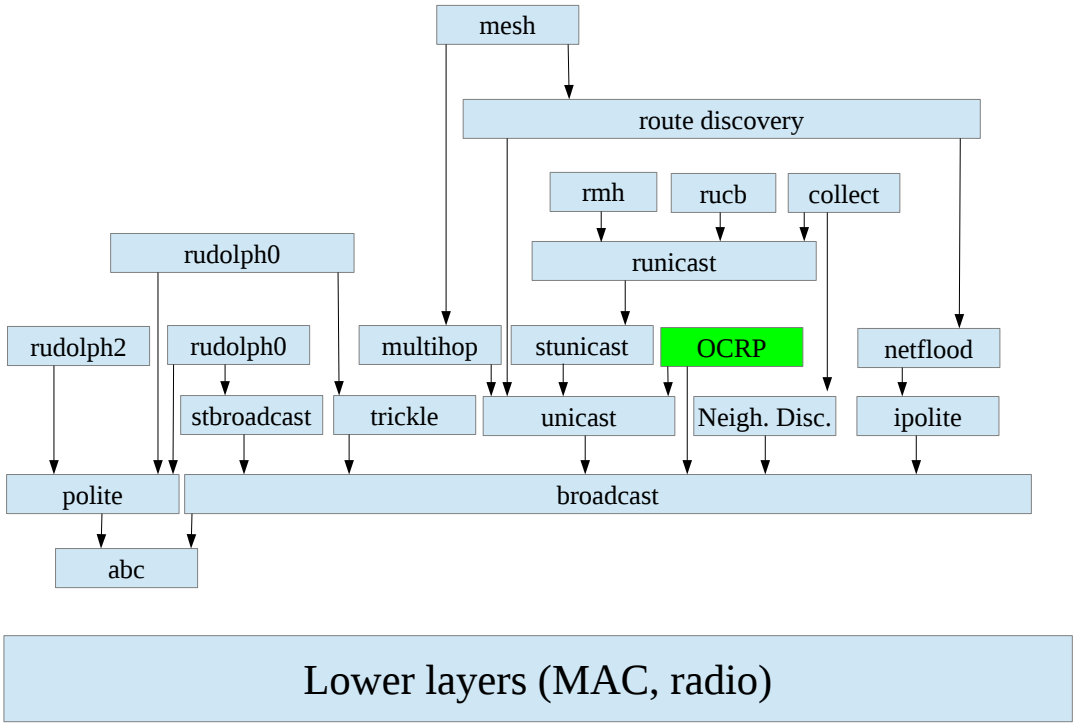

Figure 5.4: Rime stack

for all situations compared to data generation, while limiting its effect on traffic. During each call, a node selects the next neighbor in the neighbor table and sends a unicast PING message to it. On reception of the PING message, the neighbor sends back a PONG message. The PING originating node then calculates the last RTT and updates its weighted moving average according to equation 5.10. We note here that we could not implement delay estimation using data messages if data are forwarded to multiple receivers in a single broadcast; therefore no acknowledgments are used.

$$\overline{RTT}_{n+1} = \alpha \overline{RTT_n} + (1-\alpha) RTT_{n+1} \tag{5.10}$$

Data broadcast protocol This module is responsible for the generation and routing of data packets. It is detailed in the next section.

In addition to the processes described, OCRP uses an implicit time synchronization protocol, called timesynch. "The time synchronization is implicit in that no explicit time synchronization messages are sent: the module relies on the underlying network device driver to timestamp all radio messages, both outgoing and incoming. The code currently only works on the Tmote Sky platform (TelosB0) [26] and the cc2420 driver [25].

Time synchronization eases the calculation of the remaining time before the end-to-end delay deadline expires. A data message is time stamped before being queued by the transmitting node. The receiving node uses this information to calculate the message transfer time and subsequently update the remaining time before deadline expiration.

5.4.2 Data broadcast protocol

The data broadcast protocol (DBP) is responsible for the generation and the routing of data packets to the destination. Hereafter we describe how it works from the generation to the final reception.

After the generation or the reception of a data packet, a node executes the following steps:

1. It executes the OC method to select the set of valid next-hops.
2. It includes the set of valid next-hops in the data packet header.
3. It transmits the packet.

When a packet is received from the MAC layer, a node determines whether it is the destination. If the answer is yes, the data are extracted from the packet and no further actions are taken. Otherwise, it determines whether its address is included in the set of valid next-hops. If the answer is no, the packet is deleted and no further actions are taken. Otherwise, it tests whether the packet is processed for the first time or not. If the answer is no, then the packet is deleted and no further actions are taken. If the node is included in the list of next-hops and the packet is processed for the first time, then it executes stages 1, 2, and 3 as described above.

A node stores the sequence numbers and the source addresses of the last received packets. Therefore, it can verify if a packet has already been received.

The following paragraph details how a valid next hop is selected.

Computing a valid next-hop The neighbor discovery and maintenance module and the delay estimator module ensure that every node in the network knows its list of neighbors, their remaining energy, and the link quality and estimated delay to reach each neighbor.

In addition, each node knows its distance as well as its neighbors' relative distances to a given sink node. In the actual implementation, this information is embedded in hello and data messages. Initially, a sink node advertises a rank equal to zero to itself and other nodes advertise an infinite rank to any sink. Whenever a node receives a rank value strictly lesser than its actual rank for a given sink, it updates its new rank as the received rank plus one and advertises it in its hello and data packets.

For any given packet to transmit, a neighbor is valid if the following two conditions are met. First, the distance of the neighbor to the destination is strictly less than the distance of the actual node to the destination. Second, the progression of a packet to this neighbor does not violate the constraints imposed on this packet.

The OC algorithm is then implemented as follows. First, the node computes the set of nodes that respect the first condition. The node has now a partial view of the network graph (limited to its neighbors) for which it builds its 'nilpotent adjacency matrix', Ψ. The path selection algorithm is finally applied to Ψ and the list of valid next-hops is derived.

5.4.3 Other implementation details

The broadcast and unicast procedures implement a retry counter. Each packet is transmitted up to $MaxRetriesCount$ times until the packet is successfully forwarded to the next-hop. In each attempt, the underlying protocol sends an indication to OCRP to indicate whether or not the message has been successfully transmitted. Note that a successful transmission does not imply that the next-hop has effectively received the packet. This depends on the MAC protocol in use.

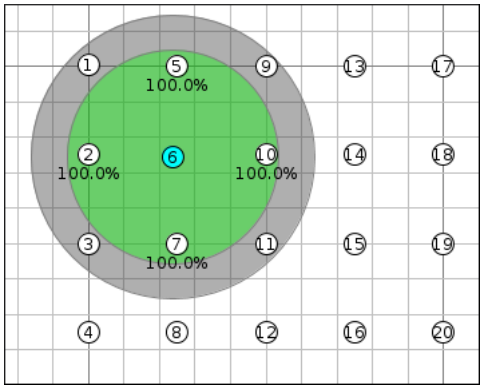

Figure 5.5: Simulated scenario. Nodes are organized in grid. Chequered green circle delimits the transmission radio. Grey circle delimits the interference zone

OCRP uses one transmission queue for broadcast and unicast packets. Therefore, the RTT calculations take the queueing delay into account. This is an important feature since the queueing delay largely affects the per hop packet transfer delay.

5.5 Performance analysis

We have performed two main experiments under 20-node and 40-node graphs. In the first one we compare OCRP with tree routing and directional flooding under noMAC to evaluate only the routing process and prove that OCRP can be applied in a WSN and in the second experiment we compare OCRP and RPL under contikiMAC to evaluate them in a more realistic scenario considering a duty-cycle MAC.

5.5.1 First Experiment

Simulation setup

We consider two multi-hop networks composed of 20 and 40 nodes organized in a grid and distributed in a surface of $100 \times 100m^2$. An example of the 20-node graph is illustrated in Figure 5.5. Node 1, located in the upper left of the network, is the sink node and the remaining nodes are senders. The transmission range is fixed to $20m$ with an interference radius equal to $30m$. Cooja simulator tool was used to compare the performance of OCRP with tree and directional flooding routing. The TelosB sky motes were emulated to obtain higher fidelity results. The radio model is the UDGM radio loss [1].The packet size is 75 bytes. The simulations run under noMAC (100% duty-cycle) in order to only focus on the routing behavior.

A node generates periodically a data packet every 0.25, 0.5, 1, 2 and 10 seconds. This means that each node generates 4, 2, 1, 0.5 and 0.1 packets per second respectively. All the packets are sent to node 1. The nodes' radios are always on. We opted for this choice to measure the intrinsic performance of routing protocols. Nodes have a unique queue for transmitting data, hello messages and unicast PING messages. Its length is fixed to 10 packets in the simulations. We note that increasing queue length penalizes static tree routing and flooding protocols since OCRP selects the nodes which have the lowest RTT. In fact, RTT calculation captures the queueing delay.

[1] Unit Disk Graph Model.

The delay (in ms), link quality (percentage of successful transmitted packets over all sent packets) and nodes' residual energy (percentage) were used as constraints, $\mathfrak{C} = (d_{\max}, lq_{\min}, re_{\min})$. $* = (+, \min, \min)$, $\mathfrak{R} = (<, >, >)$. In the simulations, $\mathfrak{C} = (100, 30, 30)$ for the 20-node graph and $\mathfrak{C} = (200, 30, 30)$ for the 40-node graph.

We measured the *end-to-end delay*, the *rate of packet delivery within deadline* and the *energy consumption*. The *end-to-end delay* is defined as the average end-to-end delay of all received packets, considering only the first reception of the same packet if multiple copies are received. The *rate of packet delivery within deadline* is defined as the number of received packets which respected the deadline divided by the total number of generated packets. The *energy consumption* is defined as the average units of energy consumed per second. Here we assume that a broadcast packet and a unicast packet consume 1 unit and 0.5 units of energy, respectively. We simply assume that the energy cost to transmit a broadcast packet is twice the cost of transmitting a unicast packet. This is approximately true for some duty-cycled MAC protocols like B-MAC [20], X-MAC [21], and Contiki-MAC [23]. In fact, the broadcast is repeatedly transmitted during a whole duty-cycle period to ensure that the packet is received by all neighbor nodes.

We compared the performance of OCRP, tree routing and directional flooding routing. We recall that each node in OCRP selects multiple valid next-hops according to the constraint vector $\mathfrak{C}$. Directional flooding selects all the next-hops that are closer to the destination (lower rank) and tree routing works by sending the packet to the static parent of the node.

Simulation results

Figures 5.6.a and 5.7.a show the ratio of packet delivery within deadline (PDD). According to the results, tree routing and OCRP have similar performance in case of 20-node graph but OCRP is slightly better in case of 40-node network. Directional flooding obtains the worst results. Overall results degrade when the traffic increases.

OCRP removes in advance all the partial paths that do not satisfy the constraints, while the other protocols forward the packets without any limit. This means that the PDD of OCRP totally depends on the deadline. Nevertheless, we can observe that the performance of both OCRP and tree routing is close in terms of PDD. Tree routing looses many packets because it is static. It uses predifined paths and high traffic deteriorates its performance due to the fact that the parents, particularly the ones located in the destination neighborhood, get congested. They are busy forwarding packets, and their queues have higher utilization. Hence, the number of received packets, as well as the number of received packets within deadline, decrease. This behavior results also in higher end-to-end delays as shown in Figure 5.6.c and 5.7.c. OCRP dynamically selects next-hops according to neighbor's state. It thereby dynamically balances load among all its neighbors, which leads to better PDD and end-to-end delay results.

In directional flooding, since every packet is forwarded to all neighbors that are closer to the destination, multiple copies of the same packet are transmitted for each newly generated one. This has the effect of overloading the nodes especially the immediate neighbors of the destination. Another impact is the high number of collisions between the nodes which try to transmit. As a consequence, this high load impacts not only the number of received packets within deadline, but also the energy consumption and the end-to-end delay as shown in all other figures.

Since reliability is one of the most important issues in WSNs, recovery processes are needed in case of failures in the network. OCRP and even directional flooding are more reliable in that case, since they use the multi-path approach and each new generated packet may follow different paths. On the other hand, tree routing keeps predefined static paths and this does not define it as a fault tolerant routing protocol. For instance, if a sensor node is out of battery, all the

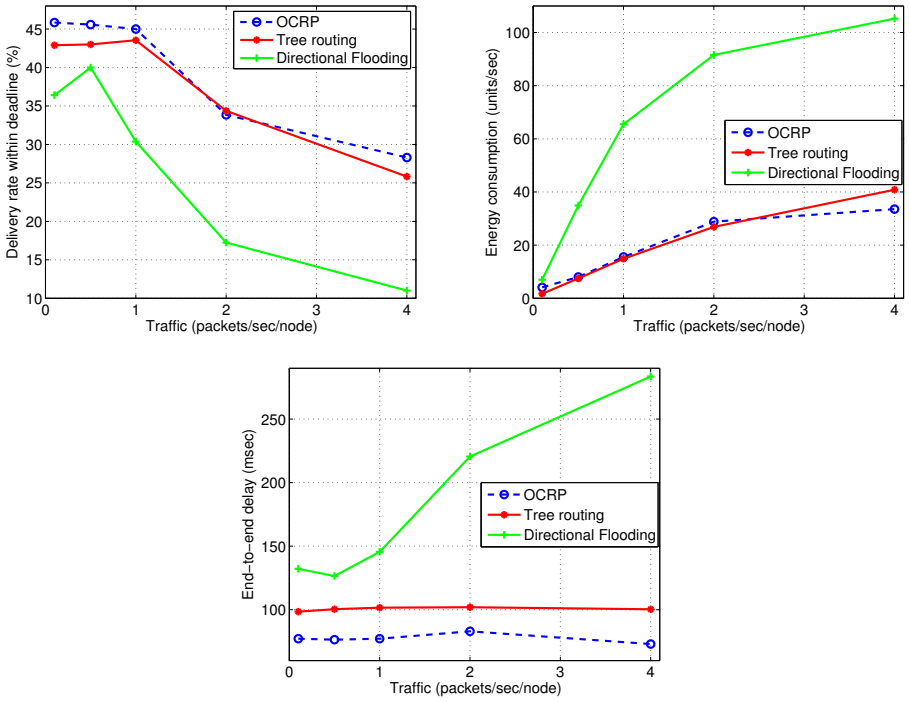

Figure 5.6: a) Ratio of packet delivery within deadline for 20-node graph, b) Energy consumption for 20-node graph, c) End-to-end delay for 20-node graph.

packets that traverse from this node will be dropped.

5.5.2 Second Experiment

In this section we compare OCRP with RPL. RPL is a routing protocol similar to tree routing but with dynamic parent selection according to the used objective function (ETX, hop count). It also specifies how to construct a Destination Oriented Directed Acyclic Graph (DODAG) in a proactive appoach, so that routes are found and maintained by sending frequently control messages. RPL has attracted a lot of interest lately and it has been widely used for the Internet of Things (IoT). For that reason, we found interesting the comparison between OCRP and RPL. We compare both protocols under contikiMAC to evaluate the routing process also over a duty-cycle MAC protocol.

Simulation Setup

In this experiment we apply the same simulation setup, but we use instead of noMAC, the ContikiMAC, which has a dynamic duty cycle. We compare OCRP with ContikiRPL under Contiki operating system and Cooja simulator in terms of delay, packet delivery, packet delivery within deadline (using different constraint vectors) and energy consumption. Every node sends a packet every 0.01, 0.05, 0.1, 0.5 and 1 second. Higher traffic brings too low performance to both protocols. The energy model is remained the same.

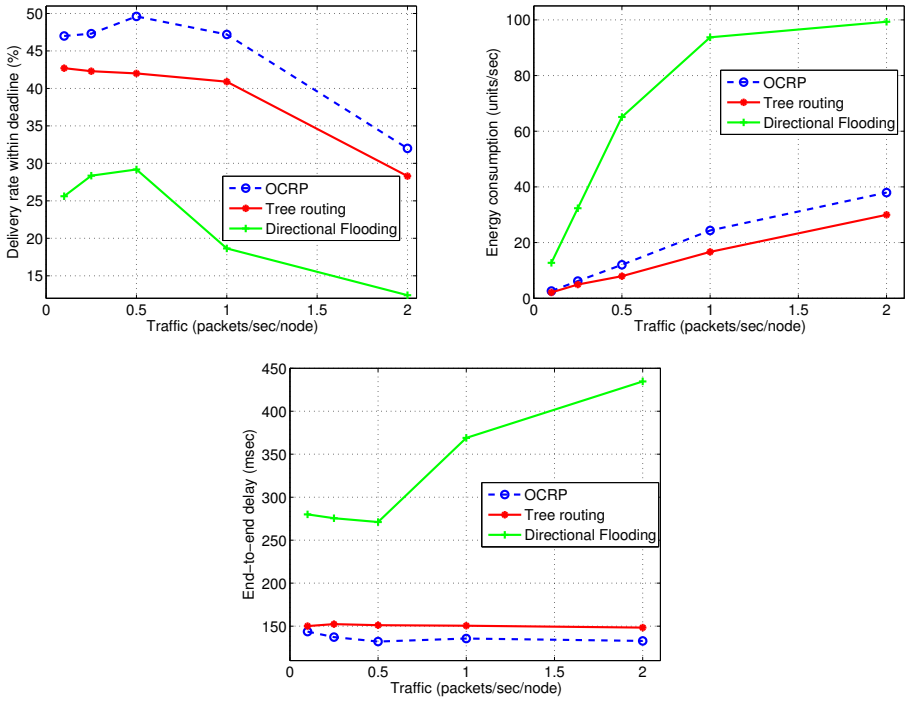

Figure 5.7: a) Ratio of packet delivery within deadline for 40-node graph, b) Energy consumption for 40-node graph, c) End-to-end delay for 40-node graph.

Simulations Results

In table 5.9 we can see the different constraint vectors under which OCRP is evaluated. $\mathfrak{C}_1$ proposes stricter constraints than $\mathfrak{C}_2$ and respectively $\mathfrak{C}_3$ proposes more relaxed constraints than $\mathfrak{C}_2$ and $\mathfrak{C}_1$. The check interval of contikiMAC is 62.5 msec, the packet size is 75 bytes and the queue length is limited to 10 packets.

As a general consideration, OCRP with stricter constraints provide lower delay, lower energy consumption and lower packet delivery ratio. That is expected as stricter constraints allow less packets to be released in the network since OCRP automatically drops the packets that violate the constraints. Moreover, bigger the graph, higher the delay and the energy consumption, while the packet delivery ratio is decreased.

In Figures 5.8.a and 5.9.a we can observe that the delay of OCRP is generally lower than RPL, since RPL continues forwarding packets even with huge delays. For the same reason the delivery rate is higher for RPL, as we can see in Fig. 5.8.c and Fig. 5.9.c. More relaxed constraints, higher the delivery ratio of OCRP. Moreover, OCRP can not reach 100% of delivery rate because not all transmissions can meet the constraints. If we consider the case where OCRP works under very relaxed constraints where basically all packets might meet the constraints, OCRP could not have 100% delivery rate because of some collisions due to broadcast function.

Figures 5.8.d, 5.8.e and 5.8.f depict the delivery ratio within deadline for $\mathfrak{C}_1$, $\mathfrak{C}_2$ and $\mathfrak{C}_3$ respectively for 20-node graph and Figure 5.9.d, 5.9.e and 5.9.f depict the delivery ratio within deadline for $\mathfrak{C}_1$, $\mathfrak{C}_2$ and $\mathfrak{C}_3$ respectively for 40-node graph. OCRP outperforms RPL in all cases. That means that even if the success rate of packet delivery in RPL is higher, the packets which arrive the sink under the time constraint are less than those of OCRP and for delay sensitive applications OCRP seems to be more appropriate than RPL. That is the main concept of OCRP,

to delete the paths that do not meet the QoS requirements automatically such that the traffic load of the network is decreased, which affects the delay and the energy consumption.

RPL consumes less energy than OCRP. This happens because of the fact that OCRP uses broadcast process to send data packets while RPL unicast, so OCRP is considered to consume the double for every data packet. Forwarding to more that one neighbor is another reason of higher energy consumption because it sends higher number of packets, but OCRP is able to find the balance between the multi-path approach and energy consumption using the basis of the algebraic architecture, which kills automatically cycles and paths that exceed the predefined constraints. OCRP continues the forwarding process only if there is a guarantee of high quality, so only "useful" packets are distributed in the network, saving much energy compared to other routing protocols.

Table 5.9: Different constraint vectors

Constraints	Delay	Reliability	Energy
$\mathfrak{C}_1$	300	50	50
$\mathfrak{C}_2$	500	30	30
$\mathfrak{C}_3$	700	10	10

The proposed delay estimation, link quality and delay measurements implementation may be further improved. Nevertheless, these encouraging results show that OCRP algorithm could be effectively implemented in wireless sensor networks. This is a proof of concept for a new class of multi-constrained routing strategies that will be improved in future work. We proved that OCRP works more efficiently in terms of delay and constrained delivery rate but for networks with low traffic RPL may also be suitable, always according to the demands of the application. The ideal case would be a hybrid solution where any routing protocol could be used for low traffic and small topologies while OCRP could be used for higher traffic and larger networks. We note that in addition to the simulations, the code has been successfully uploaded and tested using five TelosB hardware motes.

5.6 Conclusions

Operator calculus method on graphs is a new and innovative approach for solving multi-constrained routing problems. Based on this approach, we developed a new algorithm for solving multi-constrained path selection in graphs. We presented centralized and distributed versions of this algorithm. The centralized version is implemented within a node and assumes a global knowledge of the graph structure. The distributed version is implemented within every node and assumes that each node has a partial view of the network. We did an experimental evaluation of OC, we compared it with SAMCRA and we show that OC is more efficient than SAMCRA in terms of execution time.

In the distributed version, we make use of Δ which represents the related distance of every node towards the destination. This assumption is a limitation of the distributed version of OC, since in practice it is difficult for every node to have this information.

We presented a new dynamic routing protocol, called OCRP, which successfully implements the distributed algorithm within ContikiOS. In addition to the multi-constrained path selection, the RTT estimation, link quality and delay measurement protocols ensure that OCRP adapts its

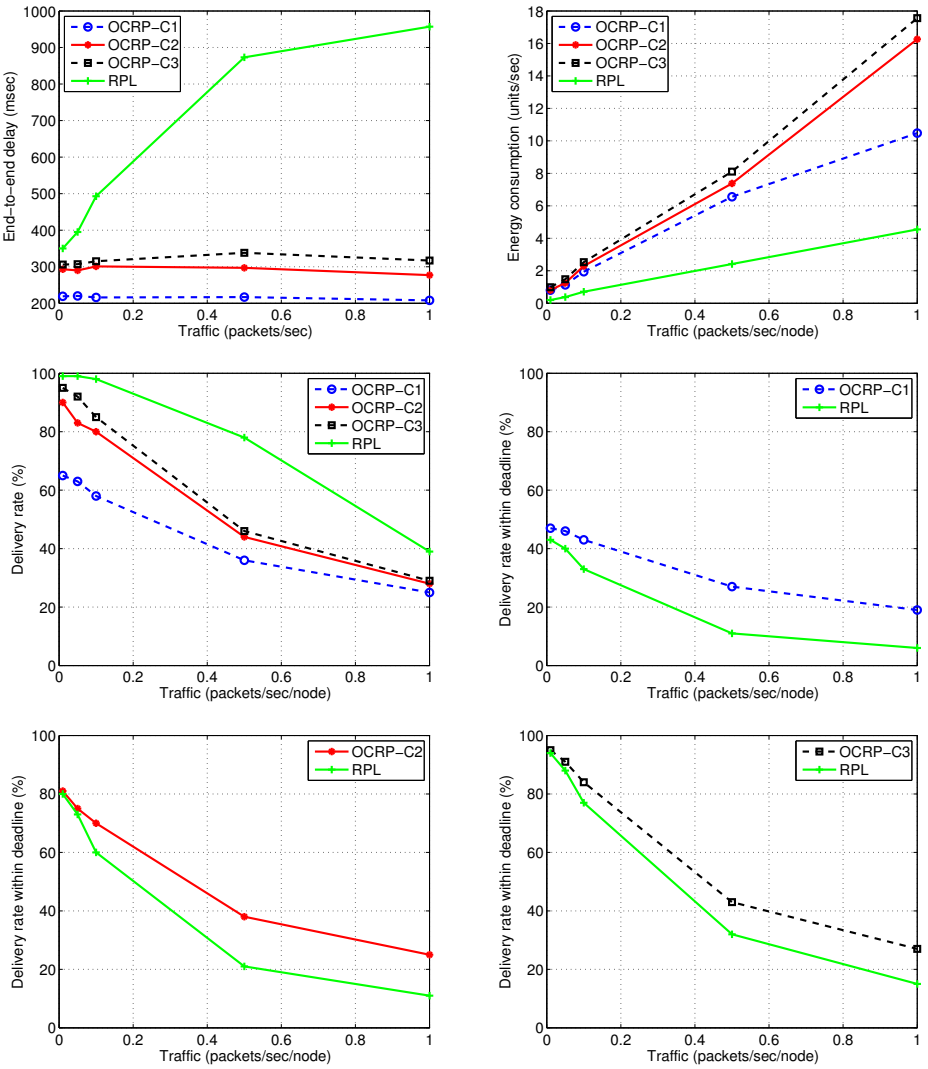

Figure 5.8: a) End-to-end delay for 20-node graph, b) Energy consumption for 20-node graph, c) Ratio of packet delivery for 20-node graph, d) Ratio of packet delivery within deadline for 20-node graph with $\mathfrak{C}_1$, e) Ratio of packet delivery within deadline for 20-node graph with $\mathfrak{C}_2$, f) Ratio of packet delivery within deadline for 20-node graph with $\mathfrak{C}_3$.

behavior according to the network's state. We used a simple energy consumption model because we aimed at evaluating the efficiency of our approach, however a realistic energy consumption model can be easily adapted. We showed through emulation using Cooja and uploading to real nodes that our distributed multi-constrained path selection algorithm is well supported by resource-constrained wireless sensor nodes.

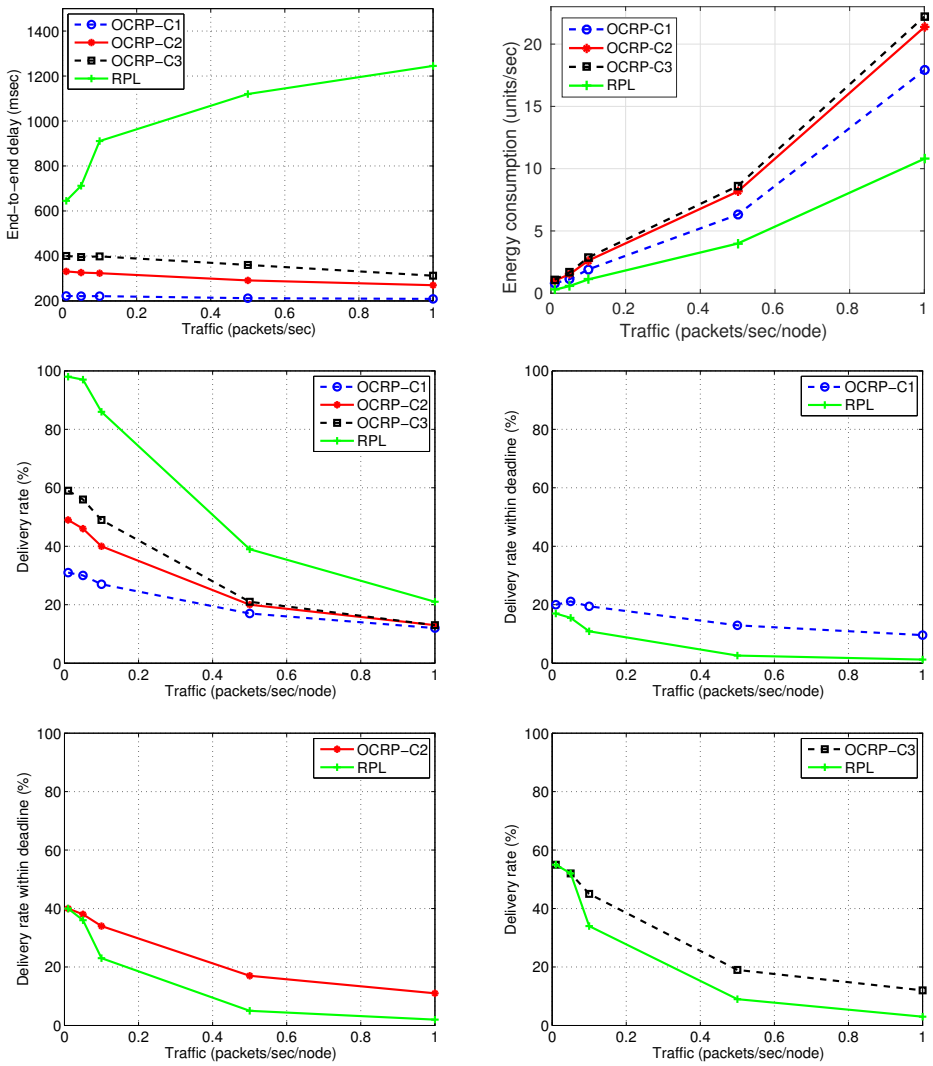

Figure 5.9: a) End-to-end delay for 40-node graph, b) Energy consumption for 40-node graph, c) Ratio of packet delivery for 40-node graph, d) Ratio of packet delivery within deadline for 40-node graph with $\mathfrak{C}_1$, e) Ratio of packet delivery within deadline for 40-node graph with $\mathfrak{C}_2$, f) Ratio of packet delivery within deadline for 40-node graph with $\mathfrak{C}_3$.

Chapter 6

Overall Conclusions and Future Work

The problem of routing has received a lot of attention from the research community and could be considered a kind of "solved problem". However, emerging wireless technologies and applications of WSNs and IoT introduce many new requirements for optimal routing solutions under multiple QoS constraints and energy optimization. The aim of this dissertation is to enrich the state of the art in routing protocols for WSNs.

6.1 Summary of the Results and Final Contributions

Our first contribution consists of OPEAR [10], an optimal probabilistic energy-aware routing protocol that balances the usage of energy between the sensors using probabilities. We have proposed a way to compute optimal probabilities that lead to optimal network lifetime. We model and solve a linear programming model. We consider a concrete duty-cycle MAC protocol (ContikiMAC) in our optimization model, while the existing works only focus on energy balancing at routing level without considering the duty-cycle, which saves an important part of the energy. OPEAR was emulated and tested against a classical proposed solution in many scenarios in Contiki environment. OPEAR balances the energy consumption, thus improving the network lifetime. OPEAR also reduces the possibility of a negative cascade effect due to the complete consumption of the energy of some sensors, thus enhancing reliability.

The second contribution is OCRP [6], a multi-constraint routing protocol which is able to simultaneously take into account multiple QoS metrics. IETF has proposed several metrics for RPL like energy, reliability, delay, ETX etc. but the existing protocols choose paths either according to only one metric or using single aggregated function with multiple metrics, but never all the metrics simultaneously. OCRP is based on the « operator calculus algebra » introduced by R. Schott and S. Staples which defines the efficient algorithm OC, allowing to find all the paths which satisfy the multiple constraints in a graph. We compared OC with existing multi-constrained algorithms and we showed the efficiency of OC in terms of complexity. OCRP is also compared against existing solutions in Contiki and we showed that it works more efficiently in terms of delay and constrained delivery rate.

In general, we observe that there are many routing protocols in the literature that try to address the problem of routing in WSNs, improving network performance in terms of reliability, delay, energy consumption etc. However, there lacks of theoretic basis for some of these protocols that would help to enhance even more the performance of these protocols. On the other hand, some recent theoretic routing algorithms on graph exhibit high practical potentials (e.g., SAMCRA, OC), their use in WSN routing protocol design would be of great benefice.

We have studied EAR, an energy-aware probabilistic routing protocol that proposes heuristic probabilities of the path selection, intending to balance the energy consumption along all feasible paths at each hop. However, the optimality of those probabilities has not been addressed, whilst there exists network flow-based routing algorithms that are proven optimal. To bridge this gap, we formally studied the problem of the existence of the optimal probabilities that lead to maximizing network lifetime. This has led us to propose OPEAR that uses the probabilities based on an optimization model, so comparing both of them; OPEAR does better in terms of network lifetime.

Energy optimization has been the main focus of the WSN protocol design. However with the diversification of both the applications and the protocols themselves in IoT, more and more QoS requirements must also be supported by WSN and IoT protocols. For instance, IETF has specified several path selection metrics for RPL that users can choose. However a network should often provide support for several or even all of them. Concerning routing protocol design, we have been interested in providing solutions for simultaneously supporting both energy saving and multiple QoS metrics (e.g., delay, reliability).

For dealing with this issue, most of the existing routing approaches are mainly based on combining some of the metrics in a global cost function. However optimizing such cost functions may introduce unfairness among the parameters depending on the weight and the way to combine them. It is also interesting to find feasible, but not optimized solutions, since in fact, multi-constrained routing problems induce Pareto fronts. SAMCRA is one of those algorithms in graph theory capable of solving multi-constrained QoS routing problem. However it only considers additive parameter combination, which is not enough for including all practical interested parameters (e.g. reliability probability of links are multiplicative, residual energy could be greater or less than a given threshold, etc.). This is why we have interested in OC (operator calculus) algorithms that can simultaneously consider multiple QoS parameters with any combination of additive, multiplicative and on threshold. Moreover it is not limited to positive path weight (although we didn't find in practice the use case of negative path weight). To verify the interest of OC, we have firstly investigated its complexity for its online use within a routing protocol. Through intensive numerical simulations and comparison with SAMCRA (the best one before OC), we showed that OC has low complexity and can deal with large instances. In a second step, we extended the centralized OC algorithm to a distributed version, and designed a routing protocol based on some theory, OCRP, that bridges the gap between the theory (OC algorithm on graph) and the practice (OCRP for WSN multi-constrained QoS routing). OCRP can be efficiently applied not only in WSNs but also a broader class of networks where multiple constraints must be considered when choosing the path.

6.2 Future Work

- We propose an approximative way to count the energy consumption in ContikiMAC, included in our optimization model. However, we did not take into account the energy of overhearing which is a common problem in WSNs. Moreover, our energy approximation is based on the ideal phase-lock mechanism. Using overhearing probabilities and considering a more realistic phase-lock mechanism can form a more accurate optimization model.

- Combing routing and relay node placement is another interesting extension of our work. The common problem in the literature is how to deploy relay nodes between the sensors and the sink node, providing connectivity while minimizing the number of relay nodes [91], maximizing the network lifetime [92] or throughput [90]. In our future work, we want to

address the problem of routing and relay node placement of a certain number of additional relay nodes in order to provide network lifetime guarantees.

- OCRP is a multi-path approach so it offers the possibility of generating more than one path. One possible extension of OCRP is to solve the two disjoint path problem [93], so that we provide protection against link or node failures and thus, we increase the robustness of the network.
- OPEAR aims to propose optimal probabilities and provide optimal network lifetime. However, the optimization model takes into account only energy constraints and it would be interesting to be extended adding delay and reliability constraints, so that we can have an optimal energy aware multi-constraint protocol.
- Cross layer optimization is a current trend for better optimizing the network performance. In the literature there are many protocols of different layers which have been implemented independently from one another. The combination of two or more layers together brings interesting results. In [89], the authors combine network and link layers, proposing a routing solution using TDMA scheduling, minimizing the delay. In our future work we can further extend our model to consider cross-layer optimization using ContikiMAC.

Chapter 7

List of Publications

- Optimal Probabilistic Energy-Aware Routing for Duty-Cycled Wireless Sensor Networks, Evangelia Tsiontsiou, Bernardetta Addis, Ye-Qiong Song and Alberto Ceselli, *IFIP NTMS 2016*, Larnaca, Cyprus (Rank B conference, best paper award).
- An Operator Calculus Approach for Multi-constrained Routing in Wireless Sensor Networks, B. Nefzi, R. Schott, S. Staples, Y. Q. Song and E. Tsiontsiou, ACM MobiHoc 2015, Hangzhou, China (Rank A conference, selection rate: 14.8%).
- Operator Calculus Algorithms for Multi-Constrained Paths, S. Jamilla, R. Schott, S. Staples, Y. Q. Song and E. Tsiontsiou, *International Journal of Mathematics and Computer Science*, 2015.

Appendix A

Synopsis

A.1 Résumé

La cadre de cette thèse est la conception de protocoles de routage pour les réseaux de capteurs. Les problèmes de recherche du routage de données dans un réseau multi-sauts sont d'une part l'optimisation de l'énergie et d'autre part le routage sous contraintes de la qualité de service (QoS) multicritères (e.g., énergie, fiabilité, délai , ...). Cette thèse apporte deux contributions par rapport à l'état de l'art: une optimisation d'un protocole de routage probabiliste pour l'équilibre de l'usage d'énergie et un protocole de routage capable de prendre en compte simultanément des métriques de QoS multiples.

En effet, pour équilibrer la consommation de l'énergie du routage lorsque des chemins multiples existent, les protocoles de routage probabiliste existants affectent une probabilité de choix à chaque chemin, soit de façon empirique, soit proportionnelle au niveau de l'énergie disponible du chemin. Nous ne savions pas quelles sont les probabilités optimales qui permettent d'avoir la durée de vie maximale du réseau. Cette thèse a permit d'établir ces probabilités optimales à l'aide de la modélisation sous forme d'un problème d'optimisation linéaire.

Quant au problème du routage multicritères, bien que des métriques multiples soient définies par RPL (un standard d'IETF), les protocoles existants choisissent la route soit sur une métrique, soit sur une fonction de coût combinant plusieurs (qui introduit par conséquent un biais de pondération), mais jamais plusieurs simultanément. Dans cette thèse, nous avons d'abord évalué numériquement les performances de l'approche « operator calculus algebra » introduit par R. Schott et S. Staples qui définit un algorithme efficace permettant de trouver tous les chemins satisfaisant les contraintes multiples dans un graphe, puis dérivé une version distribuée sur laquelle nous avons conçu un protocole de routage multi-métriques.

Ces deux contributions ont été implémentées dans l'environnement Contiki et émulées/simulées sous Cooja (un logiciel permettant de simuler des protocoles des réseaux de capteurs).

A.2 Contexte et Motivations

Le développement des technologies de l'information et de la communication est tres rapide et a conduit à une société hyper connectée dans laquelle tout est connecté aux appareils mobiles et à Internet et où une forte communication est présente entre les différents services. L'élément central de cette société hyper connectée s'appelle Internet of Things (IoT). L'IoT est possible grâce aux derniers développements en matière d'identification par radiofréquence (RFID), de capteurs intelligents, de technologies de communication et de protocoles d'Internet. Le principle

idee est que les capteurs intelligents collaborent directement sans intervention humaine pour offrir une nouvelle classe d'applications. Dans les années à venir, l'IoT devrait permettre de relier diverses technologies afin de permettre de nouvelles applications en connectant des objets physiques ensemble pour soutenir une prise de décision intelligente.

Les réseaux de capteurs sans fil (RSF) constituent l'une des technologies clés pour l'Internet des objets. Les WSN ont de nombreuses applications allant de la surveillance de la qualité de l'air à la domotique, surveillance de sécurité, surveillance de la santé, etc. Dans cette thèse, nous nous concentrons principalement sur la conception de protocoles WSN pour les utiliser dans le cadre des applications domicile pour la sante.

Actuellement, la population âgées ne cesse d'augmenter, de sorte que le suivi du comportement des personnes âgées et des personnes handicapées est devenu un enjeu majeur de santé publique. Il est primordial de maintenir une bonne qualité de vie pour leur permettre de vivre et de conserver une bonne autonomie. Néanmoins, cette autonomie peut rapidement se transformer en dépendance en cas d'accident tel qu'une chute ou un évanouissement. La solution pour résoudre ce dilemme entre l'autonomie et la surveillance consiste à instrumentaliser l'environnement de la personne. En effet, en surveillant automatiquement les principales caractéristiques environnementales de leur espace de vie, il est possible d'obtenir un mode de vie de la personne. Par exemple, mesurer la température, l'humidité, la luminosité, le niveau sonore, la présence, dans de nombreux domaines stratégiques à la maison peut fournir des données utiles pour interpréter une activité physique dans l'espace et le temps sans intervention humaine directe.

Un tel type d'environnement peut être configuré en utilisant un WSN. WSNs sont composés d'un grand nombre de nœuds de capteurs alimentés par batterie qui ont la capacité de détecter l'environnement physique, de calculer les informations obtenues et de communiquer en utilisant les interfaces radio. Un nœud spécifique, appelé sink, est en charge de la collecte et du traitement de l'information.

Les nœuds capteurs ont de nombreux avantages tels que flexibilité, absence d'infrastructure fixe et faible coût d'implémentation, mais aussi plusieurs limitations, telles que la source d'alimentation, la capacité de traitement, la bande passante, l'incertitude des données détectées et la vulnérabilité des capteurs. . D'autre part, il existe une demande de qualité de service (QoS) élevée pour les données hautement sensibles telles que les applications de soins de santé, qui nécessitent un faible délai de bout en bout, une grande fiabilité de communication et une faible consommation d'énergie. L'énergie est consommée au niveau du nœud et au niveau du routage / MAC. Les émetteurs-récepteurs consomment une grande partie de l'énergie nécessaire à la sélection de l'itinéraire, mais les connexions du réseau et du maillage entre les capteurs et l'évier sont des facteurs importants de la consommation d'énergie. Ainsi, le routage est l'une des principales sources de consommation d'énergie dans un réseau WSN où une donnée de capteur doit être transmise à l'évier en plusieurs bonds en utilisant d'autres nœuds de capteurs. Par conséquent, la routage dans un WSN est un problème de routage multi-contraintes (MCRP), satisfaisant toutes ces contraintes (délai, fiabilité, consommation d'énergie). Ce problème de décision associé au routage WSN est NP-Complete cite wang. Plusieurs solutions proposent d'aborder le MCRP en considérant différentes métriques, mais ce qui manque encore est un protocole de routage multi-contraint qui prend en compte des métriques de toute nature (mais pas seulement additives ou multiplicatives).

Un autre problème important est la durée de vie du réseau qui dépend de la durée de vie du capteur. La crise énergétique représente la partie la plus importante lors de la conception d'un protocole de routage WSN en raison de la capacité limitée des batteries de capteurs. Les opérations de rechargement ou de remplacement ne sont généralement pas faciles en raison d'environnements peu pratiques ou de coûts. L'un des principaux défis est de savoir comment

prolonger la durée de vie du réseau, en tenant compte des limites de la source d'énergie. Plusieurs approches écoénergétiques ont été suggérées pour minimiser la consommation d'énergie. Dans cette thèse, nous nous concentrons sur les protocoles de routage basés sur les flux des réseaux. L'objectif principal est de proposer une stratégie simple mais efficace pour acheminer le trafic en prolongeant la durée de vie du réseau et en tenant également compte de la couche MAC (c'est-à-dire en combinant les couches réseau et MAC).

A.3 Contributions Principales

Il y a deux contributions majeures dans cette thèse:

1. La première contribution se concentre sur la maximisation de la durée de vie du réseau. Nous utilisons des modèles de programmation basés sur les flux en utilisant des contraintes d'énergie, ce qui permet d'équilibrer la consommation d'énergie et donc de prolonger la durée de vie du réseau. Nous proposons un protocole de routage probabiliste optimisé pour l'énergie (OPEAR) cite opear qui utilise les flux extraits par le modèle d'optimisation. Nous comparons OPEAR avec EAR cite energyAware, un protocole de routage probabiliste existant, prouvant que notre protocole surpasse le protocole existant.

2. En plus, nous proposons le protocole OCRP (Operator Calculus Routing Protocol) cite ocrp. OCRP est un protocole de routage multi-contraintes qui prend en compte les métriques de toute nature. Il est basé sur l'algèbre de calcul d'opérateur (OC) cite OC-Graphs qui est proposé comme algorithme hors ligne. Nous l'avons ensuite étendu à une version en ligne. La version offline d'OC est comparée à SAMCRA, montrant la faible complexité de l'algorithme cite jamilla. OCRP est basé sur la version en ligne de OC. Il est comparé à la diffusion, au routage des arborescences et à la RPL prouvant les avantages de l'approche proposée.

Nous avons validé la faisabilité et la performance de tous les projets proposés à travers des émulations et des évaluations détaillées. Nous avons décidé d'utiliser un émulateur et non un simulateur, comme dans beaucoup de travaux précédents. Cela permet de valider nos protocoles dans un environnement réaliste prenant en compte les effets du logiciel et du hardware d'un WSN. Nous avons utilisé le système d'exploitation Contiki avec le cycle d'utilisation contikiMAC, conçu pour les moteurs WSN. Nous avons lancé des émulations à Cooja, un logiciel qui émule à la fois le logiciel et le matériel des nœuds de capteurs.

A.4 Résumé des Résultats

En général, il existe de nombreux protocoles de routage dans la littérature qui tentent de résoudre le problème du routage dans les réseaux WSN, améliorant ainsi les performances du réseau en termes de fiabilité, de délai, de consommation d'énergie, etc. protocoles qui aideraient à améliorer encore plus la performance de ces protocoles. D'autre part, certains algorithmes de routage théoriques récents sur un graphique présentent des potentiels pratiques élevés (par exemple, SAMCRA, OC), leur utilisation dans la conception de protocole de routage WSN serait très bénéfique.

Nous avons propose le protocole EAR, un protocole de routage probabiliste basé sur l'énergie qui propose des probabilités heuristiques de la sélection du chemin, visant à équilibrer la consommation d'énergie sur tous les chemins possibles à chaque saut. Cependant, l'optimalité de ces

probabilités n'a pas été abordée, alors qu'il existe des algorithmes de routage basés sur les flux de réseau qui s'avèrent optimaux. Pour combler cette lacune, nous avons étudié formellement le problème de l'existence des probabilités optimales qui conduisent à maximiser la durée de vie du réseau. Cela nous a conduit à proposer OPEAR qui utilise les probabilités basées sur un modèle d'optimisation, en comparant les deux; OPEAR fait mieux en termes de durée de vie du réseau.

L'optimisation de l'énergie a été l'objectif principal de la conception du protocole WSN. Cependant, avec la diversification des applications et des protocoles eux-mêmes dans l'IoT, de plus en plus d'exigences de QoS doivent également être prises en charge par les protocoles WSN et IoT. Par exemple, l'IETF a spécifié plusieurs métriques de sélection de chemin pour RPL que les utilisateurs peuvent choisir. Cependant, un réseau devrait souvent fournir un soutien pour plusieurs ou même tous. En ce qui concerne la conception de protocole de routage, nous nous sommes intéressés à fournir des solutions pour prendre en charge à la fois des mesures d'économie d'énergie et de qualité de service multiple (par exemple, retard, fiabilité).

Pour traiter ce problème, la plupart des approches de routage existantes reposent principalement sur la combinaison de certaines métriques dans une fonction de coût globale. Cependant, l'optimisation de ces fonctions de coût peut introduire une injustice parmi les paramètres en fonction du poids et de la manière de les combiner. Il est également intéressant de trouver des solutions réalisables, mais non optimisées, car en effet, les problèmes de routage multi-contraintes induisent des fronts de Pareto. SAMCRA est l'un de ces algorithmes en théorie des graphes capables de résoudre des problèmes de routage QoS multi-contraintes. Cependant, il ne considère que la combinaison de paramètres additifs, ce qui n'est pas suffisant pour inclure tous les paramètres utiles pratiques (par exemple, la probabilité de fiabilité des liaisons est multiplicative, l'énergie résiduelle pourrait être supérieure ou inférieure à un seuil donné, etc. C'est pourquoi nous nous intéressons aux algorithmes OC (calcul d'opérateur) qui peuvent simultanément considérer plusieurs paramètres QoS avec n'importe quelle combinaison d'additif, multiplicatif et sur seuil. De plus, il ne se limite pas au poids du trajet positif (bien que nous n'ayons pas trouvé dans la pratique le cas d'utilisation du poids du trajet négatif). Pour vérifier l'intérêt de OC, nous avons d'abord étudié sa complexité pour son utilisation en ligne dans un protocole de routage. Grâce à des simulations numériques intensives et à une comparaison avec SAMCRA (la meilleure avant OC), nous avons montré que OC a une faible complexité et peut traiter des grandes instances. Dans un second temps, nous avons étendu l'algorithme OC centralisé à une version distribuée, et conçu un protocole de routage basé sur une théorie, OCRP, qui comble le fossé entre la théorie (algorithme OC sur le graphique) et la pratique (OCRP pour WSN multi- routage QoS contraint). OCRP peut être appliqué efficacement non seulement dans les WSN mais également dans une classe plus large de réseaux où plusieurs contraintes doivent être prises en compte lors du choix du chemin.

A.5 Organisation du Manuscrit

Le manuscrit est organisé en cinq chapitres. Le premiere chapitre présente l'introduction général avec la motivation de cette thèse. Le deuxième chapitre présente une introduction à Réseaux de capteurs sans fil, et il donne également au lecteur les éléments nécessaires pour comprendre le reste du manuscrit. Le chapitre 3 présente l'état de l'art, y compris les travaux connexes sur les protocoles de routage dans les WSN, en les classant en fonction des métriques de routage qu'ils utilisent. Le chapitre 4 présente la première contribution OPEAR et le chapitre 5 la deuxième contribution OCRP. La thèse se termine avec la conclusion et quelques remarques finales qui présentent des motivations pour d'autres directions de recherche possibles qui pourraient découler

du travail.

Bibliography

[1] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks", *Proceedings of IEEE International Conference on Distributed Computing Systems*, pp. 46-55, 2005.

[2] Chipara, O.; He, Z.; Guoling Xing; Qin Chen; Xiaorui Wang; Chenyang Lu; Stankovic, J.; Abdelzaher, T., "Real-time Power-Aware Routing in Sensor Networks", *14th IEEE International Workshop on Quality of Service (IWQoS 2006)* , pp.83-92, June 2006.

[3] R. Bellman, "On a routing problem", *Quarterly of Applied Mathematics*, **16** (1958),87-90.

[4] H.W. Corley, I.D. Moon, "Shortest paths in networks with vector weights", *Journal of Optimization Theory and Applications*, **46** (1985), 79-86.

[5] H. Cruz-Sánchez, G.S. Staples, R. Schott, Y-Q. Song, "Operator calculus approach to minimal paths: Precomputed routing in a store-and-forward satellite constellation", *Proceedings of IEEE Globecom 2012*, Anaheim, USA, December 3-7. pp. 3438-3443.

[6] An Operator Calculus Approach for Multi-constrained Routing in Wireless Sensor Networks, B. Nefzi, R. Schott, S. Staples, Y. Q. Song and E. Tsiontsiou, MobiHoc 2015, Shanghai, China.

[7] Operator Calculus Algorithms for Multi-Constrained Paths, S. Jamilla, R. Schott, S. Staples, Y. Q. Song and E. Tsiontsiou, *International Journal of Mathematics and Computer Science*

[8] E.W. Dijkstra, "A note on two problems in connexion with graphs", *Numerische Mathematik*, **1** (1959), 269271.

[9] L.R. Ford Jr., D.R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.

[10] Optimal Energy-Aware Probabilistic Routing for Duty-cycled WSNs, E. Tsiontsiou, B. Addis, Y.Q. Song and A. Ceselli, NTMS 2016.

[11] R. Schott, G.S. Staples. *Operator Calculus on Graphs (Theory and Applications in Computer Science)*, Imperial College Press, London, 2012.

[12] R. Schott, G.S. Staples, "Operator calculus on generalized zeon algebras: theory and application to multi-constrained path problems", *Prépublications de l'Institut Elie Cartan*, 2011/18, (2011). Online at `http://hal.archives-ouvertes.fr/hal-00603748/PDF/staceyQoSjune11.pdf`.

[13] Z. Wang, J. Crowcroft, “QoS routing for supporting resource reservation", *IEEE Journal on Selected Areas in Communications*, **14** (1996), 1228–1234.

[14] P. Van Mieghem, F. A. Kuipers, “On the complexity of QoS routing", *Computer Communications*, **26** (2003), 376–387.

[15] G. Xue, W. Zhang, J. Tang, K. Thulasiraman, “Polynomial time approximation algorithms for multi-constrained QoS routing", *IEEE/ACM Transactions on Networking*, **16** (2008), 656–669.

[16] G.Liu and K. G. Ramakrushnam, “A* Prune: an algorithm for finding K shortest paths subject to multiple constraints" IEEE INFOCOM 2001 2(April 2001) 743-749.

[17] H. Alwan and A. Agarwal,“MQoSR: A Multiobjective QoS Routing Protocol for Wireless Sensor Networks,” ISRN Sensor Networks, vol. 2013, Article ID 495803, 12 pages, 2013.

[18] X. Huang, Y. Fang, Multiconstrained QoS multipath routing in wireless sensor networks, Journal of Wireless Networks 465-478

[19] RPL: IPv6 Routing Protocol for Low power and Lossy Networks - `http://tools.ietf.org/html/draft-ietf-roll-rpl-19`

[20] J. Polastre, J. Hill, and D. Culler. “Versatile low power media access for wireless sensor networks", *International Conference on Embedded Networked Sensor Systems (ACM SenSys)*, Baltimore, MD, USA, 2004.

[21] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: a short preamble MAC protocol for duty- cycled wireless sensor networks", *International Conference on Embedded Networked Sensor Systems (ACM SenSys)* , Boulder, Colorado, USA, 2006.

[22] K. Rana, M. Zaveri, A-star algorithm for energy efficient routing in wireless sensor network, Trends in Network and Communications, Springer (2011), pp. 232-241

[23] A. Dunkels, *The ContikiMAC Radio Duty Cycling Protocol* , SICS Technical Report T2011:13, ISSN 1100-3154, December 2011.

[24] Contiki OS. `http://www.contiki-os.org/`

[25] Timesynch module. `http://contiki.sourceforge.net/docs/2.6/a01741.html`

[26] TelosB data sheet. `http://www.willow.co.uk/TelosB_Datasheet.pdf`.

[27] Patrick Olivier Kamgueu, Emmanuel Nataf, Thomas Djotio Ndie. On Design and Deployment of Fuzzy-Based Metric for Routing in Low-Power and Lossy Networks. IEEE SenseApp 2015, Oct 2015, Clearwater Beach, Floride, United States. 2015, SensAPP 2015.

[28] H. Cruz-Sanchez, L. Havet, M. Chehaider, and Y.-Q. Song, MPIGate: A Solution to use Heterogeneous Networks for Assisted Living Applications. In 9th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC 2012), pp104–111, Fukuoka, Japan, Sept. 2012.

[29] D. S. J. D. Couto, D. Aguayo, J. Bicket and R. Morris: A High-Throughput Path Metric for Multi-Hop Wireless Routing, MobiCom '03 Proceedings of the 9th annual international conference on Mobile computing and networking,2003.

[30] S. Biswas and R. Morris: ExOR: Opportunistic Multi-Hop Routing for Wireless Networks. SIGCOMM '05 Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications,2005.

[31] R.Vidhyapriya, P.T.Vanathi: Energy Aware Routing for Wireless Sensor Networks, ICSCN, 2007.

[32] V. Angelakis, N. Gazoni and D. Yuan. Probabilistic Routing in Opportunistic Ad Hoc Networks, Wireless Ad-Hoc Networks, Intech, 2012.

[33] R. C. Shah, J.M.Rabaey: Energy Aware Routing for Low Energy Ad Hoc Sensor Networks. Wireless Communications and Networking Conference Record, WCNC 2002.

[34] A. P. Patil: Design of an energy efficient routing protocol for MANETs based on AODV. International Journal of Computer Science Issues, vol. 8, no. 4, 2011.

[35] C. Perkins: Ad hoc on demand distance vector (AODV) routing. Internet-Draft,draft-ietf-manet-aodv-04.txt,pages312,October 1999.

[36] J. Broch, D. B. Johnson, and D. A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. INTERNET-DRAFT,draft-ietf-manet-dsr-03.txt, ,October 1999.

[37] P. Nand and S. C. Sharma: Probability based improved broadcasting for AODV routing protocol. In Proceedings of the International Conference on Computational Intelligence and Communication Systems (CICN '11), pp. 621-625, October 2011.

[38] Sang-Hyun Park, Seungryong Cho, and Jung-Ryun Lee: Energy-Efficient Probabilistic Routing Algorithm for Internet of Things. Journal of Applied Mathematics, vol. 2014, Article ID 213106, 7 pages, 2014.

[39] R. Madan, S. Cui, S. Lall, and A. Goldsmith, Cross-Layer Design for Lifetime Maximization in Interference-Limited Wireless Sensor Networks. In 24th IEEE Conference on Computer Communications (IEEE INFORCOM 2005). Vol 3. Miami, FL, 1964-1975.

[40] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney, Scheduling Sleeping Nodes in High Density Cluster-based Sensor Networks. Mobile Networks and Applications, 2005.

[41] K. Hellman, and M. Colagrosso, Investigating a wireless sensor network optimal lifetime solution for linear topologies, journal of Interconnection Networks, 2006.

[42] C. F. Chiasserini, I. Chlamtac, P. Monti, and A. Nucci, Energy Efficient Design of Wireless Ad Hoc Networks, 2nd IFIP Networking, 2002.

[43] J. M. Jaffe,Algorithms for finding paths with multiple constraints, Networks, vol. 14, pp. 95–116, 1984.

[44] D. Tian, and N. D. Georganas, A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks, 1st ACM International Workshop on Wireless Sensor Networks and Applications, 2002.

[45] B. Karp and H. Kung. Greedy Perimeter Stateless Routing. In Proceedings of ACM Conf. on Mobile Computing and Networking (MOBICOM), Boston, MA, 2000. ACM

[46] Y. Yu, D. Estrin, R. Govindan, Geographical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks, UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May 2001

[47] J. H. Chang, L. Tassiulas, Maximum Lifetime Routing in Wireless Sensor Networks, IEEE/ACM Transactions on Networking, 2014.

[48] K. Kalpakis, K. Dasgupta, P. Namjoshi, Maximum Lifetime Data Gathering and Aggregation in Proceedings of IEEE International Conference on Networking (NETWORKS 02), Atlanta, GA, August 2002.

[49] S.Gandham, M.Dawande, R.Prakash, Hop-COnstrained Energy-Aware Routing in Wireless Sensor Networks, IEEE Global Telecommunications Conference, GLOBECOM 2005.

[50] A. Ghaffari, An energy efficient routing protocol for wireless sensor networks using a-star algorithm, Journal of Applied Research and Technology, Volume 12, Issue 4, pages 815-822, 2014.

[51] J. Chen, R. Lin, Y. Li, and Y. Sun, LQER: a link quality estimation based routing for wireless sensor networks, Sensors, vol. 8, no. 2, pp. 1025–1038, 2008.

[52] E. Ghqdimi, O. Landsiedel, P. Soldati,S. Duquennoy and M. Johansson, Opportunistic Routing in Low Duty-Cycle Wireless Sensor Networks, ACM Transactions on Sensor Networks, June 2014.

[53] S. Duquennoy, O. Landsiedel and T. Voigt, Let the Tree Bloom: Scalable Opportunistic Routing with ORPL, In SenSys 13: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, November, 2013.

[54] M. Bhaedwaj, A. P. Chandrakasan : Bounding the Lifetime of Sensor Networks Via Optimal Role Assignments, INFOCOM 2002.

[55] H. U. Yildiz, M. Temiz and B. Tavli: Impact of Limiting Hop Count on the Lifetime of Wireless Sensor Networks, Signal Processing and Communications Applications Conference (SIU), 2013.

[56] C. Schurgers and M. Srivastara: Energy efficient routing in wireless sensor networks, MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, 2001.

[57] O. Vermesan, P. Friess, P. Guillemin et al., "Internet of things strategic research roadmap," in Internet of Things: Global Technological and Societal Trends, vol. 1, pp. 9–52, 2011.

[58] I. Peña-López, Itu Internet Report 2005: The Internet of Things, 2005.

[59] M. Uusitalo, "Global vision for the future wireless world from the WWRF", IEEE Veh. Technol. Mag., vol. 1, no. 2, pp. 4-8, Jan. 2006.

[60] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-Level Sensor Network Simulation with COOJA. In 31^{st} IEEE Conference on Local Computer Networks (LCN), pages 641-648, Tampa, FL, USA, November 2006.

[61] A. Brandt, J. Buron, and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", IETF, RFC 5826,2010

[62] V. Chvátal: Linear Programming, W. H. Freeman, 1983.

[63] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin: Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 1993.

[64] Adam Dunkels. The ContikiMAC Radio Duty Cycling Protocol. Technical Report T2011:13, Swedish Institute of Computer Science, December 2011.

[65] Multi Protocol Interface Gateway website, `http://mpigate.loria.fr/`, last access Oct. 23rd, 2015.

[66] W. Ye, J. Heidemann, D. Estin: Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactions on Networking (TON), Volume 12 Isuue 3, June 2004.

[67] Q. Wang, W. Yang: Energy Consumption Model for Power Management in Wireless Sensor Networks, SECON 2007: 142-151

[68] S. Prayati, D. Antonopoulos, T. Stoyanova, C. Koulamas, G. D. Papadopoulos: A modeling approach on the TelosB WSN platform power consumption. Journal of Systems and Software 83(8): 1355-1363 (2010)

[69] M. Zorzi and R. R. Rao. 2003. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance. IEEE Transactions on Mobile Computing 2, 4 (Oct.-Dec. 2003), 337–348.

[70] Ali kadhum Idress, Karine Deschinkel, Michel Salomon, Raphael Couturier. Distributed lifetime coverage optimization protocol in wireless sensor networks. The journal of supercomputing 71(12) November 2015.

[71] `http://www.ti.com/lit/ds/symlink/cc2420.pdf`

[72] Contiki project website, http://www.contiki-os.org, last access Oct. 23rd, 2015.

[73] `https://en.wikipedia.org/wiki/Sensor_node`

[74] M.I. Henig, The shortest path problem with two objective functions, European J. of Operational Research, 1985, vol. 25, pp. 281-291.

[75] F. K. Shaikh and S. Zeadally. Energy harvesting in wireless sensor networks: A comprehensive review, Renewable and Sustainable Energy Reviews, 55:1041–1054, 2016.

[76] Q. Wang, K. Xu, G. Takahara, H. Hassanein, Locally Optimal Relay Node Placement in heterogeneous Wireless Sensor Networks, proc. 48th IEEE Global Telecomm, Conf., St. Louis, Missouri, Nov. 2005.

[77] E. L. Lloyd and G. Xue, Relay Node Placement in Wireless Sensor Networks, IEEE Trans. on Computers, 56(1), pp. 134-138, Jan. 2007.

[78] Othman, M. F., Shazali, K. (2012). Wireless sensor network applications: A study in environment monitoring system. Procedia Engineering, 41, 1204–1210.

[79] H. Alemdar, C. Ersoy, Wireless sensor networks for healthcare: A survey, Computer Networks 54 (2010) 2688–2710.

[80] ZigBee Alliance, ZigBee Specifications, version 1.0, April 2005.

[81] Z-Wave, "Z-Wave Protocol Overview," v. 4, May 2007.

[82] Durisic MP, Tafa Z, Dimic G, Milutinovic V (2012) A survey of military applications of wireless sensor networks. In: 2012 Mediterranean conference on embedded computing (MECO). IEEE, New York, pp 196–199.

[83] IEC 62591 Ed. 1.0 b:2010, "Industrial communication networks - Wirelesscommunication network and communication profiles - WirelessHART", 2010.

[84] 802.15.4e-2012: IEEE Standard for Local and Metropolitan Area Networks — Part 15.4: Low-Rate Wireless Personal Area Networks (LRW-PANs) Amendment 1: MAC Sublayer, IEEE Std., Apr. 16, 2012.

[85] Wireless Systems for Industrial Automation: Process Control and Related Applications, ISA-100.11a-2009 Standard, 2009.

[86] Santar Pal Singh and S.C.Sharma, "A Survey on Cluster Based Routing Protocols in Wireless Sensor Networks," Elsevier's Procedia Computer Science , 2015,45: 687-695.

[87] Koç, M.; Korpeoglu, I. Controlled Sink Mobility Algorithms for Wireless Sensor Networks. Int. J. Distrib. Sens. Netw. 2014.

[88] Ball, W. W. R. and Coxeter, H. S. M. Mathematical Recreations and Essays, 13th ed. New York: Dover, 1987.

[89] L. Lemia, and Violeta Felea. "Routing and TDMA Joint Cross-Layer Design for Wireless Sensor Networks, " International Conference on Ad-Hoc Networks and Wireless. Springer International Publishing, 2016.

[90] E. F. Flushing, G. A. D. Caro, A flow-based optimization model for throughtput-oriented relay node placement in Wireless Sensor Networks, In Proceedings of the 28th Annual ACM Symposium on Applied Computing, 2013.

[91] E. L. Lloyd and G. Xue, Relay Node Placement in Wireless Sensor Networks, IEEE Trans. on Computers, 56(1), pp. 134-138, Jan. 2007.

[92] Q. Wang, K. Xu, G. Takahara, H. Hassanein, Locally Optimal Relay Node Placement in heterogeneous Wireless Sensor Networks, proc. 48th IEEE Global Telecomm, Conf., St. Louis, Missouri, Nov. 2005.

[93] A. Kumar, S. Varma, Geographic node-disjoint path routing for wireless sensor networks, IEEE Sensors Journal, 10 (6) (2010), pp. 1138-1139.