Network Topology and Protocol Design for Efficient Consensus in Sensor Networks

César Asensio Marco

Network Topology and Protocol Design for Efficient Consensus in Sensor Networks

Doctoral Dissertation for the Degree *Philosophiae Doctor (PhD)* at
the Faculty of Engineering and Science, Specialisation in
Information and Communication Technology

University of Agder
Faculty of Engineering and Science
2017

Doctoral Dissertation at the University of Agder 151
ISSN: 1504-9272
ISBN: 978-82-7117-845-1

Printed by the Printing Office, University of Agder
Kristiansand

To my wife, Paloma Juárez,
to my daughter Martina Asensio,
to my parents, Antonio and Alicia.

This thesis is dedicated to the memory of Eugenio Celada.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Baltasar Beferull-Lozano for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I thank Daniel Alonso for being my unofficial co-advisor, for all the discussion and constructive comments. For all the time we spent doing research together. You are one of the best researchers and persons I have ever met.

I thank all the lab-mates and researchers that have participated at some point in the research involved in this Thesis, for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we had travelling together due to project meetings and conferences around the globe. A brief list of you guys: Eduardo Portu, Gustavo Hernández, Ibrahim Kaliffe, Huang Xioage, Santosh Sah and Fernando Camaró.

I would also like to thank Tonje Sti and Julia Cervera for her help during our first months in Norway and Emma Elisabeth Horneman for fighting on our side against the evil monster of bureaucracy. You three form the lighthouse that guides any PhD Student to the coast of Grimstad.

I thank my parents for supporting me spiritually throughout writing this thesis and my life in general.

Last, but never least, I must thank my unbelievably supportive wife Paloma. She has demonstrated amazing patience throughout my lengthy working sessions over the last years. Thank you for your support and for giving birth to our beautiful daughter, Martina, I am honoured to have the opportunity to know you and humbled to raise you. I look forward to accompanying you along your own quest to comprehend and find your place in the world. Thank you, pretty girl, for motivating me to keep reaching for excellence. Thank you for everything that you are, and everything you will become.

Summary

In the new era of Internet of Things, complex sensor networks are becoming crucial to link the physical world to the Internet. These sensor networks, composed by hundreds or thousands of nodes, provide many important services to promote a heightened level of awareness about the area of interest such as in predictive maintenance, intelligent buildings, enhanced security systems, etc. In order to make these services possible, several signal processing tasks are needed to support their operation, some widely used examples of these tasks are parameter estimation, signal detection and target tracking. These tasks allow to improve the services by inferring missing data, reducing samples noise, etc., at the cost of some collaboration of the network nodes that implies their repeated communication over time. Most of these solutions are consensus-based strategies, which have recently attracted a great deal of research work because of its simplicity. These are in-network algorithms, where each node only exchanges information with its immediate neighbors and these are able to obtain global information as a function of some sensed data. A relevant example is the average consensus algorithm, which its goal is to obtain, in a distributed way, the average of the initial data. These algorithms avoid the need of performing all the computations at one or more sink nodes, thus, reducing congestion around them and incrementing the robustness of the network.

In this dissertation, we focus on improving consensus algorithms in terms of different parameters and under different types of communications and network configurations. Each setting considered requires its own assumptions and methodologies, since the convergence conditions for each of them are related but different in general. In particular, in this work, all of the methodologies proposed are based on designing how the underlying communications are performed.

In static networks, where the asymptotic convergence to the average value is easily ensured, a topology optimization can be a priori performed in terms of several relevant parameters. In particular, we optimize the network topology to make consensus algorithms fast and energy efficient. In this setting and for continuous systems, we derive a general framework to minimize several energy related functions under different network and nodes constraints.

To solve it, we propose a fractional convex-concave optimization problem with different constraints that leads to obtain the optimal topology in terms of the energy function considered. As a significant variation of the previous results, we also optimize the network topology in discrete systems. The discretization of the system introduces a weight matrix and certain step-size (related to the discrete increments of time) in the process. We show that if this step-size is small enough, the energy related problems stated before can be still casted as convex-concave fractional problems with the weight matrix as a unique optimization variable. As the step-size of the process increases in size, a discrete system requires a different approach. To solve it, we aim to find another formulation based on adding a constraint on the connectivity and solving the problem several times (for different values of the step-size). In addition, two low-complex methodologies with different computational requirements are proposed to a posteriori redesign an existing topology by the collaboration of the network nodes.

On the contrary, in time-varying (random) networks, it is needed to guarantee a minimum accuracy of the algorithm, while maximizing the number of simultaneous exchanges of data to ensure fast convergence. Regarding random and asymmetric communications, we propose a novel gossip algorithm, which is based on the residual information that is generated when an asymmetric communication is performed. We exploit this information to preserve the summation of the process and accelerate it. Moreover, our proposal is useful in the case of having both unicast and broadcast communications, presenting faster convergence in both schemes than existing approaches in the related literature. When the problem of wireless interferences constraining the communications is additionally taken into account, we propose a novel and computationally efficient link scheduling protocol that correctly operates in the presence of secondary interference. Our protocol is easily implementable and does not require global knowledge of the network. The main objective of this new protocol is to be suitable for a cross-layer scheme in which the execution of the average consensus algorithm is favoured, ensuring necessary conditions for its convergence with certain accuracy. Additionally, the number of simultaneous links is additionally considered in order to make the convergence of the consensus process as fast as possible.

Contents

List of Figures

List of Tables

Symbols and Abbreviations

Sets

$\mathscr{X}$	a finite nonempty set of elements
$x \in \mathscr{X}$	x belongs to the set $\mathscr{X}$
$\|\mathscr{X}\|$	cardinality of the set $\mathscr{X}$
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^+$	the set of positive real numbers
S	the set of symmetric matrices
S^+	the set of positive semidefinite matrices

Vectors and matrices

$x, \mathbf{x}, \mathbf{X}$	a scalar, a vector and a matrix
$[\mathbf{x}]_i$	the entry of the i-th row of vector $\mathbf{x}$
$[\mathbf{X}]_{ij}$	the entry of the i-th row and j-th column of matrix $\mathbf{X}$
$\mathbf{x}^T, \mathbf{X}^T$	the transpose of a vector $\mathbf{x}$, the transpose of a matrix $\mathbf{X}$
$\mathbf{X} \succeq \mathbf{Y}$	means that $\mathbf{X} - \mathbf{Y} \in S$
$\mathbf{X} \succ 0$	$\mathbf{X} \in S$, that is, X is positive definite
$\mathbf{X} \succeq 0$	$\mathbf{X} \in S$, that is, X is positive semidefinite

Operators and relations

$\approx$	approximately
$\equiv$	is equivalent
Pr	probability
lim	limit
max	maximum
min	minimum
$\|x\|$	the absolute value of x
$\|\|\mathbf{x}\|\|_2$	the 2-norm of a vector $\mathbf{x}$
$\mathbb{E}[x]$	the expected value of x

Graph theory terms

$\mathscr{G}$, $\mathscr{G}(k)$	a time-invariant graph, a time-varying graph
$\mathscr{V}$	the time-invariant set of nodes of $\mathscr{G}$
$\mathscr{E}$, $\mathscr{E}(k)$	the time-invariant and time-varying set of links of $\mathscr{G}$
e_{ji}	edge (link) from node j to node i
$\mathscr{N}_i$, $\mathscr{N}_i(k)$	the time-invariant and the the time-variant neighborhood of node i
$\mathbf{A}$, $\mathbf{A}(k)$	the time-invariant and the time-variant Adjacency matrix
$\mathbf{D}$, $\mathbf{D}(k)$	the time-invariant and the time-variant Degree matrix
$\mathbf{L}$, $\mathbf{L}(k)$	the time-invariant and the time-variant Laplacian matrix

Common notations associated with consensus

$x_i(k)$	state of node i at time k
$x_i(0)$	initial state of node i
$\mathbf{x}(0)$	initial state vector
$\mathbf{x}_{\text{avg}}$	Weight matrix
$\mathbf{0}$	the zero vector
$\mathbf{1}$	the all-ones vector
$\mathbf{I}$	the identity matrix
$\mathbf{P}$	the connection probability matrix
$\rho(\mathbf{X})$	the spectral radius of $\mathbf{X}$
$\lambda_i(\mathbf{X})$	i-th eigenvalue of $\mathbf{X}$
$\lambda_1(\mathbf{X})$	the first eigenvalue of $\mathbf{X}$
$\lambda_N(\mathbf{X})$	the last eigenvalue of $\mathbf{X}$
α	step size (link weight)

Common notations associated with communications

β	SINR threshold
γ	path loss exponent
p_i	power transmission of node i
R_{max}	maximum transmission range
R_{inh}	inhibition radius
N_0	background noise

Acronyms

SINR	signal to interference noise ratio
MAC	medium access control
WSN	wireless sensor network
SWN	small world network
SFN	scale free network
RGN	random geometric network
MSE	mean square error
r.v.	random variable
i.i.d.	independent identically distributed

Part I

Introduction and Background

Chapter 1

Introduction

Due to recent advances in microelectronics, the development of tiny devices with computing, wireless communication and sensing capabilities has drawn great attention both in the academia and the industry. These are low cost, low energy consumption and small footprint devices, which have made possible to create complex sensor networks for very diverse applications. The sensor nodes gather data from the environment and eventually make simple processing of it. The data can be transferred to a central node in a centralized network, or instead it can be locally processed in a decentralized network [52][6][34], being this second approach more scalable and robust in complex sensor networks. In many real scenarios, such sensor networks have been revealed as excellent solutions for advanced monitoring [61] and control problems [68][1], which usually rely on fast-decision tasks, such as event detection [29][80], dynamic estimation [70][78][94] or tracking [79][36], see Figure 1.1.1. Current sensor networks, composed by hundreds or thousands of nodes, are able to distributively solve those tasks by the repeated communication of their nodes. Most of these solutions are consensus-based strategies [65] [88], which have recently attracted a great deal of research work because of its simplicity. These are in-network algorithms, where nodes only exchanges information with their neighbors and are able to obtain global information as a function of some sensed data. A relevant example is the average consensus algorithm, which its goal is to obtain, in a distributed way, the average of the initial data. These algorithms avoid the need of performing all the computations at one or more sink nodes, thus, reducing congestion around them and incrementing the robustness of the network.

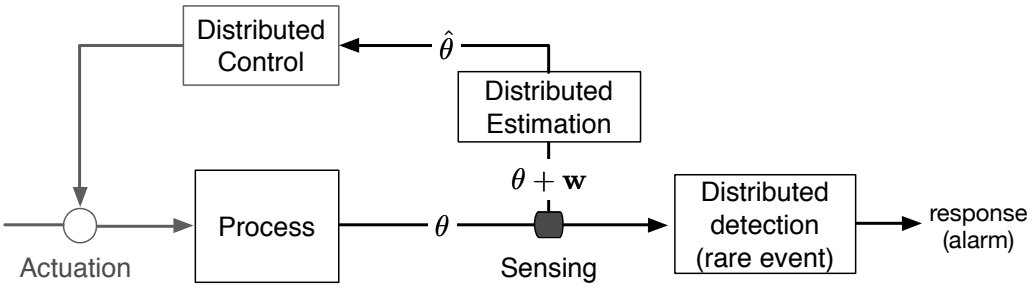


Figure 1.1.1: Sensor networks have been revealed as an excellent tool for performing advanced monitoring and control in a totally distributed manner. In these tasks, consensus algorithms play a crucial role by refining measurements $\theta + \mathbf{w}$, performing distributed estimations $\hat{\theta}$ of the monitored parameters and distributively controlling the phenomena of interest θ.

1.1 Motivation of this work

Although consensus algorithms are widely used due to their simplicity and decentralized philosophy, these are iterative processes that incur in a repeated exchange of data with an associated power consumption. Then, if the consensus process converges slow, taking lot of time steps, the total energy consumption of this process may become excessive. This is specially dramatic when sensor nodes are not plugged in to any form of power supply, then energy becomes a scarce resource and must be properly administrated. In situations where reaching the devices is practically impossible or when the cost of mobilizing personnel is high, used-up batteries may not be easily replaced. Although a good option is to provide the sensor nodes with self-rechargeable batteries using for instance solar energy, this is not always possible and, in any case, a low-power consumption is always essential to guarantee a longer lifetime for the entire network.

Many solutions have been proposed for energy efficiency at various levels of the system architecture, covering from the hardware design [43], coverage [35], MAC [92], nodes placement [30], etc. In regards to consensus algorithms, the convergence speed and the power consumption have been identified as the most important performance metrics [76], which in general, are both determined by the topology of the network. The combination of the convergence time and the power consumption (per time step and per node) leads to the crucial concepts of total energy consumption and network lifetime,

which determine the energy efficiency of the consensus process.

Thus, it is crucial to devise new topology optimization methods capable of both accelerating the consensus process and reducing the associated power consumption (per time step and per node), in order to improve both the energy required to reach consensus and the global network lifetime.

An additional limitation to being iterative is that traditional consensus algorithms are synchronous [65][89], which in practice means that every node is required to communicate with all of their neighbors before a new information exchange is started, see Table 1.1.1. This problem has been solved by allowing asynchronous communications between nodes, leading to the appearance of the so-called gossip algorithms [31]. However, both consensus schemes [89] and [31] still require all the communications to be performed in a symmetric manner, that is, information exchanges between nodes are bidirectional at every time step, see Table 1.1.1. Unfortunately, due to the time constraints imposed by consensus-based tasks, which require frequent wireless communications, it is not easy to satisfy bidirectional communications. Note that, in this setting, ensuring symmetric communications implies that any node t, after a transmission of data to any other node r, should wait until the acknowledgement, including the data from r, is received. In addition, this latter node r should also wait for the acknowledgement from t. In this context, every node waiting for an acknowledgment cannot mix information with any other node until the current data exchange is finished. Thus, while a node is waiting, the data received from other nodes, must be stored and processed after the current one. This procedure requires a control mechanism that introduces both a communication overhead and an uncontrolled delay in gossip algorithms, which is opposed to their simplicity and decentralized nature.

Thus, it is also necessary to derive new asymmetric gossip schemes capable to obtain similar results to traditional gossip schemes, but only using asymmetric communications.

In practice, all these wireless communications are performed under interference, making it necessary to include further processing to obtain collision avoidance during the consensus process, see Table 1.1.1. A collision implies a waste of energy due to an unsuccessful communication. Moreover less information is exchanged per time step, making consensus algorithms to converge slower. Finally, random packet losses due to collisions make more difficult to

ensure instantaneous symmetric communications. In the context of WSNs, different MAC protocols have been widely utilized to control the channel access and deal with the interference in order to improve the throughput, the packet reception rate or both at the same time. However, none of them take into account explicitly the performance parameters of the application in a cross-layer scheme.

Thus, it is also necessary to include further processing at MAC level in order to favour, in a cross-layer manner, the convergence of consensus algorithms, reducing both the associated consensus error and the energy consumption, while satisfying the time constraints imposed by the consensus-based tasks.

Type communications		**Implementation details**
Synchronous communications	Symmetric	Every node *bidirectionally* communicates with all their neighbors in every iteration: uncontrolled delay, ACKs needed
	Asymmetric	Every node communicates with all their neighbors in every iteration: uncontrolled delay
Asynchronous communications	Symmetric	One or several nodes (randomly chosen) *bidirectionally* communicate with one or several neighbors in every iteration: ACKs needed
	Asymmetric	One or several nodes (randomly chosen) communicate with one or several neighbors
Random communications		One or several nodes (randomly chosen) communicate with one or several neighbors. Packet losses may occur due to several factors.

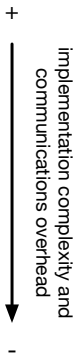


Table 1.1.1: Synchronization and symmetry in the communications can only be ensured by introducing control mechanisms that imply certain overhead. These extra mechanisms allow to exactly characterize the convergence time of these algorithms and ensure asymptotic convergence to the average value. As soon as the communication exchanges are random, the consensus process becomes random and only results on probabilistic terms can be derived. Thus, different approaches and solutions are needed to improve these algorithms, depending on the type of communications assumed.

We have identified three different problems to be solved within the consensus schemes found in the related literature. This Thesis is motivated by these unsolved problems and focus on deriving new mechanisms at different layers of the protocol stack in order to improve consensus algorithms in terms of one or several of the following consensus-related parameters: error, convergence time, energy consumption and network lifetime.

1.2 State of the art

Consensus and gossip algorithms have been the subject of intensive research in recent years. These algorithms form the core of multiple tasks, such us distributed estimation, tracking, detection and control. All these tasks may be found separately in practice as standalone applications or forming part of more complex system, as the control loop illustrated in Figure 1.1.1. In some of these tasks, the convergence time is crucial to ensure their correct operation, since inaccurate data due to a slow convergence may significantly affect the whole system. This is why most of the early research in this field have focused on the convergence time of consensus algorithms.

In the presence of a *synchronous* scheme of communications, where the topology is generally undirected, it is well known that the convergence rate can be measured in terms of the so-called algebraic connectivity [46]. For that reason, several early works have focused on maximizing this parameter based on redesigning the underlying network graph. In most cases [65][64], these topology perturbations involve generating and adding some unrealistic long links to the network. The main idea behind adding these long links is to introduce the small-world effect [86] into the original network. However, creating links between distant nodes might not be always possible due to the power constraints that are frequently present in sensor nodes. Alternatively, it has been also investigated how to obtain suitable weights that nodes can use to mix the exchanged information in order to improve the convergence time. One important work in this direction is the one carried out in [88], where the authors showed that it is possible to formulate the problem of finding the fastest convergence as a convex semidefinite optimization problem. This approach gives the optimal weights to combine the information based on the solution of a convex optimization problem. An important drawback of this method is that it requires knowing, at every node, the connectivity of the deployed network. To solve this drawback, a sub-gradient algorithm was also proposed in [88]. However this method is relatively slow, and has no simple stopping criterion that guarantees a certain level of sub-optimality. Other approaches [67] and [77] focus on other advanced techniques, such us extrapolation methods and diffusion equations of particles, in order to accelerate the distributed linear iterations performed by the consensus algorithm.

Most of the aforementioned consensus solutions are able to solve the average consensus problem in a reasonable number of communication steps. However, these do not take into account the power constraints that sensor nodes usually have, which implies that these are not easy to implement in real sensor networks. Moreover, fast convergence does not mean an energy efficient process. Thus, for obtaining a distributed approach useful in a sensor network scenario, it is also necessary to take into account the power consumption required by each node at each communication step involved during the consensus process.

Only few works take into account both the convergence time and the power consumption simultaneously. First, in [24] it was shown that there exists an optimum power consumption per iteration step that minimizes the total energy required to achieve consensus. In this first work, the results are over random geometric graphs assuming no a priori known locations of the nodes. This work was extended in [76], where it is shown how to jointly optimize both parameters, namely, the convergence time and the power consumption, by using a convex-concave fractional problem. However, in this method, the power consumption of each individual node is not taken into account, which leads to significantly different nodes lifetime. Finally, due to the decentralized nature of the consensus algorithm itself, it might be also desirable to find distributed methods that lead to better designed topologies with an associated faster average consensus algorithms with a reduced power consumption.

In practice, when having a synchronous scheme of communications, it is rare to have asymmetric exchanges of data. The main reason is that the synchronization mechanism, utilized to ensure that all the nodes have finished their exchanges of data with their neighbors before the next exchange takes place, can be also used to ensure the symmetry of the communications. As a consequence, this is a scenario that only few works [65] have explored and it is generally investigated at theoretical level only.

In the case of being in the presence of *asynchronous* communications, the simplest algorithm solving the average consensus problem is the pairwise gossip [31], where each node randomly picks up a neighbor computes a symmetric pairwise average with it. This approach introduces an interesting asynchronous mechanism to the original deterministic protocol for consensus

[89] and the nodes are still able to converge to the average with certain degree of accuracy. However, this approach suffers from the locality of the updates, taking lot of iterations to reach consensus. Since each iteration involves an associated communication cost, geographic routes [41] and broadcast communications [22] has been proposed to reduce the total number of performed iterations. Finally, the work in [41] was refined by adding path averaging in [26], which makes this algorithm even faster.

In order to reach convergence almost surely to the average value, previous schemes need symmetric communications, which would imply the use of several control mechanism with an additional communication cost. This is exacerbated when it is required to ensure symmetric long routes or efficient broadcast communications, which both require complex control mechanisms in practice. In order to partially solve the problem of having asymmetric communications, several algorithms have been proposed in the related literature. Most of them are based in the so called push-sum algorithm. The push-sum algorithm was initially presented in [58], it was latter generalized in [25] to different weights, and was finally applied to a broadcast scenario in [53].

Besides asynchronous and asymmetric communications, other factors have been considered in the related literature, such as quantized data [59] [33], delayed communications [66][87], packet losses [95], etc. All these problems may appear when considering real communications, where interference and other environmental factors constraint the amount of information exchanged per time step. In order to simplify consensus algorithms and avoid the need to include further processing to deal with the aforementioned constraints, some results have been derived in [82]. In this work, necessary and sufficient conditions for almost sure asymptotic consensus are derived for such framework. The first condition is that the network must be connected on average, that is, a multi-hop path between every pair of nodes must be possible along the time the consensus process lasts. This first condition can be easily introduced into existing carrier sense-based protocols [7] [37][72][74][83] by ensuring that the transmission range is large enough to create a multi-hop path between any pair of nodes over time. The second condition is that the instantaneous weight matrices that result from the random activation of links are balanced. This second condition is not easily satisfied for real communication and we have to search for an alternative that suits a real scenario, while ensuring cer-

tain degree of accuracy. An alternative condition easier to satisfy is that the connection probabilities between any pair of nodes are symmetric, that is, the links are symmetric only on probabilistic terms. However, even this last condition requires some additional mechanism that can hardly be obtained by using standard carrier sense-based protocols without being extremely conservative. In addition, the benefits of this condition need to be clarified.

Regardless of the condition to be satisfied, these require to introduce certain criteria about how the links are activated during the iterations of the consensus algorithm. Deciding how the links are activated is a common characteristic of time division multiple access (TDMA) protocols, which are a family of protocols that are present in different ways in the context of wireless networks. For example, in [23], flows and demands are given and the problem consists of obtaining a set of routes, a bandwidth assignment, and a link schedule over a sufficiently long scheduling interval that maximizes the total throughput. In [28], the problem is explicitly referred as link scheduling, where link demands are given and the problem is to construct a schedule of minimum length that satisfies all demands. However, in all cases of existing link scheduling protocols the targeted metric is the communications throughput (related to convergence time) not the consensus energy.

Type of communications		**State of the art** solid work in this topic I some relevant work in this topic I marginal work in this topic
Synchronous communications	**Symmetric**	convergence time: completely characterized and well optimized consensus energy: some centralized methods available to optimize it network lifetime: not considered consensus error: it is asymptotically zero, no work needed here
	Asymmetric	convergence time: characterized for certain particular cases consensus energy: not considered network lifetime: not considered consensus error: characterized for certain particular cases
Asynchronous communications	**Symmetric**	convergence time: probabilistic properties totally characterized consensus energy: only number of exchanged packets considered network lifetime: not considered consensus error: characterized for certain particular cases
	Asymmetric	convergence time: certain characterization, but not optimized consensus energy: not considered network lifetime: not considered consensus error: new schemes proposed to alleviate it
Random communications and losses		convergence time: characterization for some particular cases consensus energy: average energy expression becomes too complex network lifetime: average lifetime expression becomes too complex consensus error: some bounds derived

Table 1.2.1: Summary of the state of the art in consensus algorithms under different schemes of communications. The main parameters considered in this study are the convergence time, the energy consumption, the network lifetime and the deviation from the average value.

1.3 Beyond State of the Art: Scope and Objectives

Although a lot of research work has already been done in the area of consensus algorithms, there is still much more to do, see Table 1.2.1. In this table we have identified four main parameters that ideally should be optimized in all existing consensus schemes. However, as introduced in the previous section, real communications and environmental factors make this optimization extremely difficult in general.

There are two different ways of dealing with real communications and the constraints these impose. The first methodology focuses on masking all the associated complexity by introducing several control mechanisms at different layers of the communications stack. Then, consensus algorithms operate as if these communications were synchronous and symmetric with an uncontrolled delay between time-steps. The second methodology focuses on dealing with the asynchronous and asymmetric communications directly at consensus level by introducing new mechanisms to ensure certain degree of accuracy. Intuitively, an additional methodology can be envisioned, which would consist of a combination of the two previous schemes, that is, dealing only with synchronization or symmetry at consensus level and treating at alternative layers the one not considered at consensus level.

In this dissertation, we focus on improving consensus algorithms in terms of one or several parameters (see Table 1.2.1) under the two first schemes described before. A summary of our objectives at different layers, including some cross-layer[1] interactions can be seen in Table 1.4.1.

When synchronous and symmetric communications are satisfied by certain control mechanisms at lower layers, our main objective is to optimize the network topology in order to make consensus algorithms fast and energy efficient. In this setting and for continuous systems, we aim to derive a general framework to solve the problem of minimizing the total energy consumption and maximizing the network lifetime under general network and nodes constraints (***objective 1***). Then, in order to show the efficiency of this approach, we focus on particularizing it to three different node settings appeared in a real industrial environment [1].

[1] A cross-layer approach sends feedback dynamically via the layer boundaries to improve the results of one or the two layers involved.

The first setting is an heterogeneous network composed by sensor nodes with different communication and energy supply capabilities. The location of the nodes is known and these only perform point to point communications. Besides of improving the energy efficiency of this network setting, we aim to show that the resulting optimal topology varies from a Random Geometric Graph to a Small World Graph structure and this is related to certain optimization parameters (***objective 1.1***). We also consider a variation of the previous setting, where an existing network infrastructure already exists and the objective is to provide extra sensing capabilities by adding new sensor nodes. These sensor nodes, with much lower energy capabilities than the existing ones, are added to the network over time. In this case, we aim to solve the problem iteratively in order to obtain the best connectivity for the new incoming node. Moreover, we focus on showing that depending on the affinity that these new nodes have for the existing ones, the resulting graph tends to follow the growth rule of a Scale Free Graph (***objective 1.2***). Finally, we also consider the setting where the communications are broadcast, and the transmission range of every node is limited to a certain amount. In this case, besides improving the energy efficiency of the network, we focus on showing that the resulting graph follows a Random Geometric Graph model where the nodes may present different transmission ranges (***objective 1.3***).

Besides the topology optimization framework for different nodes settings, we also focus on introducing several relevant functions related to the energy consumption of nodes (***objective 2***). The topology optimization framework

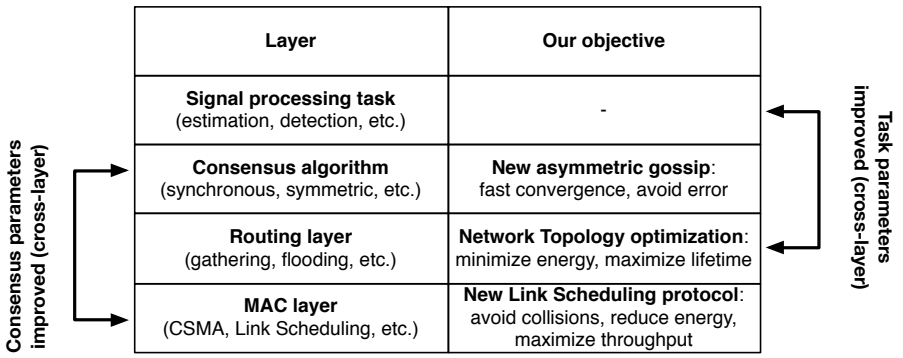


Table 1.3.1: A summary of our objectives at different layers, including some cross-layer techniques.

is then applied to two real applications appeared in practice, such as the distributed detection of a known signal (***objective 2.1***) and the distributed estimation of an unknown parameter (***objective 2.2***). In the first case, an energy efficiency is required to ensure the actuation of every node, while in the second, certain minimum accuracy of the estimator is required, which in turn is related to the nodes lifetime. Both tasks illustrate the importance of consensus algorithms and how our methodology works in practice.

As a significant variation of the results for continuous systems, we also aim at optimizing the network topology in discrete systems (***objective 3***). The discretization of the system introduces a weight matrix and certain step-size (related to the discrete increments of time) in the process. In this context, we focus on showing the influence of the step-size in the optimization solution. When the step-size is small enough, our objective is to show the solution found for discrete systems is equivalent to the one for continuous systems (***objective 3.1***). Oppositely, if the step-size is not small enough, since both systems are no longer equivalent, we focus on deriving a related optimization scheme (***objective 3.2***) based on applying the previous methodology for different maximum values of the step size. Finally, we are also interested on finding alternative methodologies to be applied in already formed topologies that can be performed in a distributed manner (***objective 3.3***).

When control mechanisms are no longer used at lower layers, we have to deal with asynchronous and asymmetric communications at the consensus level. In this scenario, our main objective is to design a new asymmetric gossip algorithm that exploits the residual information involved in each asymmetric exchange to reach average consensus almost surely (***objective 4***). Under this new gossip scheme we aim at achieving faster convergence (***objective 4.1***) and with less energy consumption (***objective 4.2***) than existing studies in the related literature. Moreover, we focus on making this new scheme suitable for both unicast and broadcast communications (***objective 4.3***).

Alternatively, when interference is also considered, we are interested on a cross-layer scheme in which the MAC-layer decisions are influenced by the consensus layer (***objective 5***). We first aim to show why symmetric probabilities of connection ensure average consensus with certain degree of accuracy. Then, we should derive the conditions that a MAC protocol should satisfy in

order to ensure these symmetric probabilities (***objective 5.1***). For this purpose, a new link scheduling protocol suitable for this cross-layer approach is needed (***objective 5.2***). Finally, we aim at finding a close expression for the number of simultaneous links, which is a crucial parameter for our link scheduling protocol and influences the convergence rate of the average consensus protocol (***objective 5.3***).

1.4 Thesis Outline and Contributions

The work described in this PhD thesis is the result of the study of different consensus schemes that appeared under different communication settings. On the one hand, when all the complexity introduced by real communications is masked at lower layers, we focus on studying the network topology and how it affects the operation of the average consensus algorithm. What is sought is to optimize this topology in order to improve the convergence time, the energy consumption that a consensus process requires to reach asymptotically average consensus and also the network lifetime that results from it. On the other hand, when the complexity must be treated at consensus level, we propose new schemes of communication to reduce the probabilistic error and to improve, whenever possible, the convergence time and the energy consumption of this process.

The organization of this Thesis and the main contributions from each chapter are described below, along with the publications resulted from the involved research. The use of graph theory terms and concepts is widely used throughout the dissertation, and for that reason, the next chapter is devoted to introduce them. The third chapter is devoted to consensus algorithms while the subsequent four chapters present the main results of this Thesis, concerning consensus under different schemes of communications and network settings. A last chapter, including the conclusions and the related open problems is finally presented.

Chapter 2

This chapter presents a review of fundamental concepts of algebraic graph theory and the notation used in subsequent chapters. Notions of graph con-

nectivity and an overview of common graph topologies are presented. The matrices associated with a graph, with special focus on the Laplacian matrix along with its spectral properties are described. Definitions for deterministic time-invariant and for random time-varying graphs are finally presented.

Chapter 3

This chapter presents an overview on consensus algorithms, giving an overview of the different schemes of communications where these algorithms are executed and how these affect their performance.

Chapter 4

When synchronous and symmetric communications are satisfied by certain control mechanisms at lower layers, we optimize the network topology to make consensus algorithms fast and energy efficient. In this setting and for continuous systems, we derive a general framework to minimize several energy related functions under different network and nodes constraints. This is based on a fractional convex-concave optimization with different constraints that leads to obtain the optimal topology in terms of the energy functions considered. In addition, we relate the use of several parameters to the appearance of certain graph models. Finally, we show how two signal processing tasks

Type of communications		**Our contribution**
Synchronous communications	**Symmetric**	We provide a general framework for the optimization of the total energy consumption of consensus and the network lifetime. Different type of complex networks and functions are considered in the optimization. Continuous and discrete systems are both studied.
	Asymmetric	-
Asynchronous communications	**Symmetric**	-
	Asymmetric	We propose a new asymmetric gossip scheme that ensures average consensus with certain degree of accuracy, outperforming in terms of convergence time and power consumption the existing approaches in the related literature.
Random communications and losses		We propose a crosslayer approach where the decisions at the MAC layer are made to favour the performance of the consensus algorithm. Ensuring certain accuracy of the process.

Table 1.4.1: A summary of our contributions to the different schemes of communications considered in this Thesis.

can be benefited from the use of our methodology.

The work in this chapter has led to the following publications:

[11] **Asensio-Marco C.**; Alonso-Román, D.; Camaró-Nogués, F.; Beferull-Lozano, B.; "Topology optimization for a trade off between energy cost and network lifetime in average consensus." *The 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013.
[19] **Asensio-Marco, C.** and Beferull-Lozano, B.;" Optimal topology design for energy efficient consensus in broadcast Wireless Sensor Networks." *2014 IEEE International Conference on Communications (ICC)*, June 2014.
[8] **Asensio-Marco, C.**; Alonso-Román, D.; Beferull-Lozano, B.; "Achieving energy-efficient distributed consensus in wireless Scale Free Networks." *2014 IEEE International Conference on Communications (ICC)*, June 2014.
[9] **Asensio-Marco, C.**; Alonso-Román, D.; Beferull-Lozano, B., "Ensuring High Performance of Consensus-Based Estimation by Lifetime Maximization in WSNs," *2015 International Conference on Distributed Computing in Sensor Systems (DCOSS)*, vol., no., pp.108,114, 10-12, June 2015.
[20] **Asensio-Marco, C.** and Beferull-Lozano, B. "Energy Efficient Consensus Over Complex Networks", *IEEE Journal of Selected Topics in Signal Processing*, On page(s): 292 - 303 Volume: 9, Issue: 2, March 2015.

Chapter 5
As a significant variation of the previous results, we also optimize the network topology in discrete systems. The discretization of the system introduces a weight matrix and certain step-size (related to the discrete increments of time) in the process. We show that if this step-size is small enough, the energy related problems stated before can be still casted as convex-concave fractional problem with the weight matrix as a unique optimization variable. As the step-size of the process increases in size, a discrete system requires a different approach. To solve it, we propose a formulation based on adding a constraint on the connectivity and solving the problem several times (for different values of the step-size). Finally, we propose two distributed methodologies with different computational cost that, by perturbing the topology, are able to improve the same parameters than the centralized one.

The work of this chapter has led to the following publications:

[12] **Asensio-Marco, C.** and Beferull-Lozano, B., "Accelerating Consensus Gossip Algorithms: Sparsifying Networks Can Be Good for You," *2010 IEEE International Conference on Communications (ICC)*, May 2010.
[13] **Asensio-Marco, C.** and Beferull-Lozano, B., "Finding sparse connectivity patterns in power-constrained ad-hoc networks for accelerating consensus algorithms." *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, August 2010.
[14] **Asensio-Marco, C.** and Beferull-Lozano, B., "A greedy perturbation approach to accelerating consensus algorithms and reducing its power consumption", *IEEE Statistical Signal Processing Workshop (SSP)*, June 2011.
[17] **Asensio-Marco, C.** and Beferull-Lozano, B., "Network Topology Optimization for Accelerating Consensus Algorithms under Power Constraints", *IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, May 2012.

An important publication that relates Chapters 4 and 5 is the following:

[21] **Asensio-Marco, C.** and Beferull-Lozano, B., "Optimal topology design for energy efficient consensus in Broadcast Wireless Sensor Networks", submitted to IEEE Transactions on Wireless Communications. October 2016.

Chapter 6
Regarding random and asymmetric communications, we propose a novel gossip algorithm, which is based on the residual information that is generated when an asymmetric communication is performed. We exploit this information to preserve the summation of the process and accelerate it. Moreover, our proposal is useful in the case of having both unicast and broadcast communications, presenting faster convergence and less energy consumption in both schemes as compared to existing approaches.

The work of this chapter has resulted in the following publications:

[18] **Asensio-Marco, C.** and Beferull-Lozano, B.; "Fast Average Gossiping Under Asymmetric Links in WSNs." *2014 Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*, August 2014.

Chapter 7

Regarding random and asymmetric communications in the presence of wireless interference, we propose a computationally and efficient link scheduling protocol that correctly operates in the presence of secondary interference. Our protocol is easily implementable and does not require global knowledge of the network. The main objective of this new protocol is to be suitable for a cross-layer scheme in which the execution of the average consensus algorithm is favoured, satisfying certain conditions that are required to ensure convergence with certain accuracy.

The work of this chapter has led to the following publications:

[16] **Asensio-Marco, C.** and Beferull-Lozano, B., "Link scheduling in sensor networks for asymmetric average consensus," *2012 IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2012.

[15] **Asensio-Marco, C.** and Beferull-Lozano, B., "Asymmetric average consensus under SINR-based interference," *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, August 2012.

[10] **Asensio-Marco, C.**; Alonso-Román, D. and Beferull-Lozano, B., "Cross-layer MAC Protocol for Unbiased Average Consensus under Random Interference," submitted to IEEE/ACM Transactions on Networking, August 2016.

Chapter 8

This chapter summarizes the results and impact of this Thesis and discusses open problems.

Other relevant publications related to this Thesis[2] *are the following:*

[52] Hernández-Peñaloza, G.; **Asensio-Marco, C.**; Beferull-Lozano, B., "Distributed estimation of statistical correlation measures for spatial inference in WSNs," *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, August 2012.
[7] Alonso-Román, D.; Celada-Funes, E.; **Asensio-Marco, C.**; Beferull-Lozano, B.; "Improving reliability and efficiency of communications in WSNs under high traffic demand." *IEEE Wireless Communications and Networking Conference" (WCNC)*, April 2013.
[34] Camaró-Nogués F.; Alonso-Román, D.; **Asensio-Marco, C.**; Beferull-Lozano, B.; "Reducing the observation error in a WSN through a consensus-based subspace projection." *IEEE Wireless Communications and Networking Conference" (WCNC)*, April 2013.
[6] Alonso-Román, D.; Camaró-Nogués, F.; **Asensio-Marco, C.**; Beferull-Lozano, B.; "Distributed subspace projection over wireless sensor networks with unreliable links". *IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2013.
[37] Celada-Funes, E.; Alonso-Román, D.; **Asensio-Marco, C.**; Beferull-Lozano, B.; "A reliable CSMA protocol for high performance broadcast communications in a WSN." *IEEE Global Communications Conference (GLOBECOM)*, December 2014.
[3] Alonso-Román, D.; **Asensio-Marco, C.**; Celada-Funes, E. and Beferull-Lozano, B. "Consensus based distributed estimation of biomass concentration in reverse osmosis membranes." *In Proceedings of the 1st ACM International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater'15)*, April 2015.

[2]We have decided not to include these results in the body of this Thesis for not extending its length excessively. These works are only referenced along the text to illustrate possible applications of consensus algorithms.

Chapter 2

Graph Theory Concepts

Most of the network structures in the real world that present an information flow between their components can be modeled as a graph. In general, a graph is a mathematical abstraction used to represent the relation between the elements of a given set. When talking about graphs, the elements of the set are usually called vertices and then the edges are the entities that describe the relation between pairs of these vertices. In the context of sensor networks, the abstractions of vertices and edges correspond to nodes and links respectively. The term node in a real network may refer to any existing sensor device, while the term link may correspond to any form of communication between two or several devices. Finally, the suitable communication of multiple nodes leads to the formation of different complex networks, which can be generally represented by well-known random graph models. Some relevant models are: the random geometric graph (RGG) model, the small world graph (SWG) model and the scale free graph (SFG) model, among others.

This chapter gives an overview over the main concepts in algebraic graph theory, which are useful in the analysis of consensus algorithms. We start by defining, in Section 2.1, fundamental graph concepts together with their associated matrices. Then, in Section 2.2, well-known random graph models are presented. We introduce the Laplacian matrix in Section 2.3, a powerful tool for the analysis of the performance of consensus algorithms. Finally, the impact of the distance between the nodes is considered in Section 2.4.

2.1 Fundamental Concepts of Graph Theory

In general, a graph is defined as $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ is the set of vertices and $\mathscr{E}$ is an unordered set of pairs of vertices from $\mathscr{V}$, which are called edges and represent a connection between any two vertices. In general, a graph can be classified according to the characteristics of the set of edges. In particular, if the set of edges change over time, we are in the presence of a time time-varying graph. Oppositely, if this set remains static over time, the graph can be classified as a time-invariant graph. Similarly, if the edges are directed or undirected, we say that the resulting graph is directed or undirected respectively. Finally, if there exist edges that connect vertices with themselves, the graph is said that contains *self-loops*. Independently of these general graph properties, there are several concepts that are common to every graph:

A path is a sequence of vertices such that from each of them there is an edge to the next vertex in the sequence. A path may be infinite, but a finite path always has a first vertex, called its start vertex, and a last vertex, called its end vertex. Both of them are called terminal vertices of the path. The other vertices in the path are internal vertices.

A *cycle* is a path such that the start vertex and end vertex are the same. The choice of the start vertex in a cycle is arbitrary.

A *shortest path* is a path connecting two vertices with the minimum number of edges. The *distance* between two vertices in a graph is the number of edges in a shortest path connecting them.

The *diameter* of a graph is the greatest distance between any pair of vertices. To find the diameter of a graph, first find the shortest path between each pair of vertices. The greatest length of any of these paths is the diameter of the graph.

The *clustering coefficient* of a graph is a measure of the degree to which nodes in a graph tend to cluster together.

A graph is a *strongly connected graph* if there is a multi-hop path between every pair of vertices in the graph.

Finally, a graph is said to be *fully connected* if every pair of distinct vertices is connected by a unique edge.

2.1.1 Time-invariant and Time-varying Graphs

This section introduces the concepts of time-varying and time-invariant graphs, which are both used throughout this work.

Time-varying graph: When the set of edges of a graph change over time[1], it can be modeled as a *time-varying* graph $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$, consisting of a set $\mathcal{V}$ of N nodes and a set $\mathcal{E}(k) \subset \mathcal{E}$ of edges. The set $\mathcal{E}$ denotes the maximum number of links that can be simultaneously established, which may include every of the $N \times (N-1)$ possible links or just a subset of them. Note that we are assuming graphs without self-loops.

A time-varying graph can be generated from a finite number of edge sets or can be the result of some random process. In this work, every time-varying graph considered is random, where the set of vertices is assumed constant throughout time and the set of edges varies randomly over time.

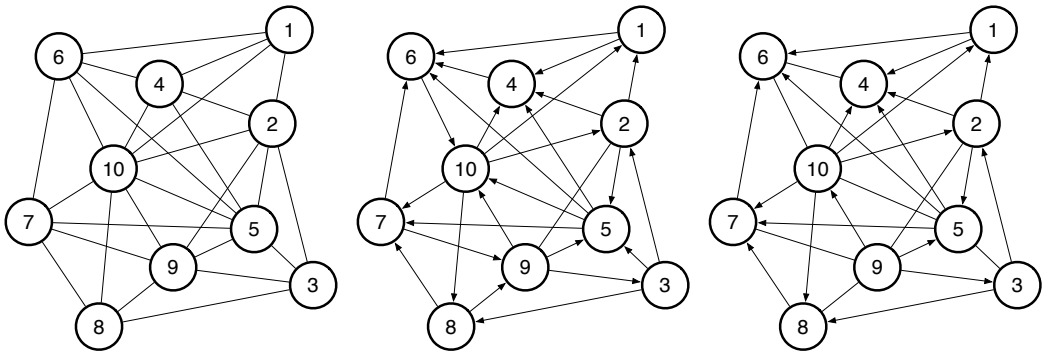
Time-invariant graph: In the particular case that the sequence of connectivity patterns (subsets of links) remains constant over time[2], that is, $\mathcal{E}(1) = \mathcal{E}(2) = \ldots = \mathcal{E}(k-1) = \mathcal{E}(k)$, the graph can be modeled as a *time-invariant* graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

In this work, when a time-invariant graph is considered, this must be strongly connected. Alternatively, if the considered graph varies over time, this only should be connected on average terms.

A *graph connected on average* is a time-varying graph in which any of the instantaneous subgraphs $\mathcal{G}(\mathcal{V}, \mathcal{E}(k))$ may not be a strongly connected graph, but with the passage of sufficient time, a route between any two nodes is established.

[1] Due to several factors affecting the wireless channel, a pair of nodes that is communicating in a particular time instant might not be able to communicate in future communication steps. This variation over time may occur due to several reasons, such as nodes failures, packet losses, delays, etc.

[2] In a real wireless environment, ensuring every node communicate with each other in every iteration requires several control mechanisms and a communications overhead.



(a) undirected graph (b) directed graph (c) mixed graph

Figure 2.1.1: An example of graphs with different type of edges.

2.1.2 Directed and Undirected Graphs

Besides that the edges may vary over time, these may be directed or undirected. We denote an edge from node j to node i as e_{ji}, where the presence of an edge e_{ji} between two nodes indicates that there exists an information flow from node j to node i.

Directed graph: If a direction is assigned to the edges, the relations are asymmetric and the graph is called a directed graph, or a digraph. For a directed edge e_{ji}, j is called the transmitter and i is called the receiver of the information flow. In practice, this implies that a node sends a packet without controlling whether is or is not correctly received, that is, without requiring that the corresponding receiving node sends a response with its own information.

Undirected graph: If no direction is assigned to the edges, the relations are symmetric and the graph is called an undirected graph. In practice, these bidirectional communications requires a control mechanism such as acknowledgements (ACKs) to ensure that the packets in both directions are correctly received.

Figure 2.1.2 shows an example of graphs where the links do or do not present direction. In this work, we only consider directed and undirected graphs which are the ones that are found in practice.

2.1.3 Matrices and Structures associated with a Graph

Given any instantaneous graph[3] $\mathscr{G}(k) = (\mathscr{V}, \mathscr{E}(k))$ at certain time instant k, the neighborhood of a node i can be defined as:

$$\mathscr{N}_i(k) = \{j \in \mathscr{V} : e_{ij} \in \mathscr{E}(k)\} \quad i \in \mathscr{V},\ i \neq j$$

and we can assign an $N \times N$ adjacency matrix[4] $\mathbf{A}(k)$, given by:

$$[\mathbf{A}(k)]_{ij} = \begin{cases} 1 & \text{if } e_{ij} \in \mathscr{E}(k) \\ 0 & \text{otherwise} \end{cases} \tag{2.1.1}$$

where an entry $[\mathbf{A}(k)]_{ij}$ is equal to 1 if there exists a link between node i and j and 0 otherwise.

As illustrative example, lets take the undirected graph presented in Figure 2.1.1 (a). The corresponding adjacency matrix is the following:

$$\mathbf{A}(k) = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Then, at time instant k, a graph realization $\mathscr{G}(k)$ can be seen as a spanning subgraph of a graph with fixed edges set.

[3] Note that for a time-invariant graph, the formulation can be obtained by removing the time index, since for a time constant graph $\mathscr{E}(1) = \mathscr{E}(2) = \ldots = \mathscr{E}(k-1) = \mathscr{E}(k)$.

[4] The matrix $\mathbf{A}(k)$ represents an instantaneous connectivity pattern that is determined by the communications between the network nodes at time instant k.

Given any instantaneous adjacency matrix, we can define the corresponding in-degree and out-degree of a vertex i as the sum of the rows and columns of this matrix, that is:

$$d_i^{in}(k) = \sum_{j=1}^{N} [\mathbf{A}(k)]_{ji} \text{ and } d_i^{out}(k) = \sum_{j=1}^{N} [\mathbf{A}(k)]_{ij}$$

A vertex i is said to be balanced at time instant k if its in-degree and out-degree are equal, that is, $d_i^{in}(k) = d_i^{out}(k)$. An instantaneous digraph is said to be balanced if all its vertices are balanced. Therefore, every undirected graph is a balanced graph.

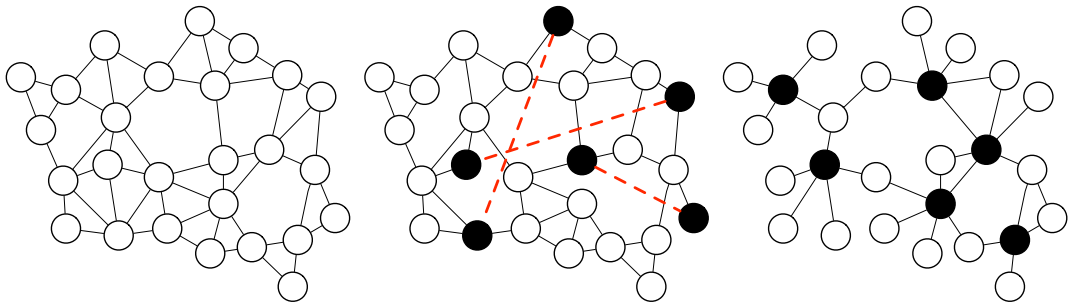
The instantaneous degree matrix $\mathbf{D}(k)$ is the $N \times N$ matrix with entries given by:

$$[\mathbf{D}(k)]_{ij} = \begin{cases} d_i^{out}(k) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{2.1.2}$$

where $d_i^{out}(k)$ is the out degree of node i at time instant k. Thus, the entries of the degree matrix are equal to the row sums of the adjacency matrix.

Again, if we consider the example of Figure 2.1.1 (a), the corresponding degree matrix is:

$$\mathbf{D}(k) = \begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{pmatrix}$$



(a) RGG (b) SWG (c) SFG

Figure 2.1.2: Examples of three different graph models: (a) Random geometric graph (RGG), (b) Small world graph (SWG) and (c) Scale free graph (SFG).

2.2 Examples of Complex Graphs

Complex graphs can be classified according to their diameter, clustering coefficient and degree distribution. The combination of these parameters characterize the graph model:

Small world graph: These graphs present a degree distribution similar to a regular graph. However, in a small-world graph, most nodes are not neighbors of one another, but most nodes can be reached from every other by a small number of hops [86]. In other words, these graphs present a high cluster coefficient and a small diameter.

Scale free graph: The distinctive characteristic of scale-free graphs is that their degree distribution follows a power law relationship defined by $p(q) \sim cq^{-h}$, where c is a normalization constant and $2 \leq h \leq 3$. The highest-degree nodes are often called *hubs*, and are thought to serve specific purposes in their graphs [2]. A high clustering coefficient and a medium to small diameter characterize this model.

Random geometric graph: This is a graph where some random coordinates are assigned to each of the N nodes in a square area and only those who are separated by a distance shorter than a given radius R are connected by a link. The probability p of any node to have q neighbors, in a large scale RGG, is given by $p(q) = \frac{e^{-d_{\text{avg}}} d_{\text{avg}}^{q}}{q!}$, where d_{avg} is the average number of neighbors [69]. These graphs present a medium to high clustering coefficient and a medium to large diameter.

2.3 Laplacian Matrix of a Graph

This section is devoted to the Laplacian matrix, which is a useful mathematical tool to describe in a compact form the connectivity particularities of a graph. We denote by $\mathbf{L}(k)$ the instantaneous $N \times N$ Laplacian matrix of the graph $\mathscr{G}(k)$ given by:

$$\mathbf{L}(k) = \mathbf{D}(k) - \mathbf{A}(k) \tag{2.3.1}$$

where $\mathbf{D}(k)$ and $\mathbf{A}(k)$ are the degree matrix and the adjacency matrix defined in (2.1.2) and (2.1.1) respectively.

By construction, the Laplacian matrix of an undirected graph is always symmetric and the opposite occurs with a directed graph. Then, for the case of the instantaneous undirected graph presented in Figure 2.1.1, we have the following Laplacian matrix:

$$\mathbf{L}(k) = \begin{pmatrix} 4 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & -1 \\ -1 & 6 & -1 & -1 & -1 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 4 & 0 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & -1 & 0 & 6 & -1 & -1 & -1 & 0 & 0 & -1 \\ 0 & -1 & -1 & -1 & 7 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & -1 & 5 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & -1 & -1 & 5 & -1 & -1 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & -1 \\ 0 & -1 & -1 & 0 & -1 & 0 & -1 & -1 & 6 & -1 \\ -1 & -1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & 8 \end{pmatrix}$$

Assuming that the eigenvalues of the Laplacian matrix are arranged in increasing order $0 = \lambda_1(\mathbf{L}(k)) \leq \lambda_2(\mathbf{L}(k)) \leq ... \leq \lambda_N(\mathbf{L}(k))$, all of them are positive except the first one, which is equal to zero, that is, $\lambda_i(\mathbf{L}(k)) \geq 0 \, \forall i > 1$ and $\lambda_1(\mathbf{L}(k)) = 0$, which means that $\mathbf{L}(k)$ is always positive-semidefinite. The number of times 0 appears as an eigenvalue in the Laplacian is the number of connected components in the graph. $\lambda_1(\mathbf{L}(k))$ is always 0 because every Laplacian matrix has an eigenvector that, for each row, adds the corresponding nodes degree to a -1 for each neighbor, thereby producing zero by definition. The second smallest eigenvalue of $\mathbf{L}(k)$ is the algebraic connectivity (or Fiedler value) of $\mathscr{G}(k)$.

2.4 Graphs in the Euclidean Space

When considering real networks, maintaining the underlying topology implies a cost on communications, which usually dependes on the physical distance between the nodes. In the particular case of wireless networks, the information must be transmitted over the air, which requires certain radio power consumption. Thus, to maintain a topology at time instant k, each node of the network needs to invest some power consumption for this. Such powers can be stored in a vector as follows:

$$\mathbf{p}(k) = (\mathrm{p}_1(k), \mathrm{p}_2(k), \ldots, \mathrm{p}_N(k)) \tag{2.4.1}$$

where $[\mathbf{p}(k)]_i = \mathrm{p}_i(k)$ denotes the power consumption at time instant k of node i. Let us assume a generic path loss model with path loss exponent γ, then the power that node i requires to successfully communicate with node j is given by $p_{ij} = p_{\min} r_{ij}^{\gamma}$, where $p_{\min}$ is the minimum power required at the receiver to successfully decode the incoming information[5] and r_{ij} is the distance between nodes i and j. Let us consider broadcast communications where a unique transmission is sufficient to reach the nodes within the transmission range. Then, the expression for $\mathrm{p}_i(k)$ is given by[6]:

$$\mathrm{p}_i(k) = \max_{j \in \mathscr{N}_i(k)} \{p_{ij}\} = \max_{j \in \mathscr{V}} \{p_{ij} \cdot [\mathbf{A}(k)]_{ij}\} \tag{2.4.2}$$

Notice that there is a one-to-one mapping between $\mathbf{p}(k)$ and matrix $\mathbf{A}(k)$.

Besides having to ensure a minimum power $p_{\min}$ at the receiver, there exists another inherent limitation on radio communications, consisting of a maximum power transmission $p_{\max}$ at the transmitter. Both constraints determine the maximum distance $R_{\max}$ to which a receiver can correctly decode a message in absence of interference from a transmitter, which is given by:

$$R_{\max} = \left(\frac{p_{\max}}{p_{\min}} \right)^{1/\gamma} \tag{2.4.3}$$

[5] For simplicity, we are assuming a simple SNR model in this work.

[6] If unicast communications are considered instead, the equivalent expression would be $\mathrm{p}_i(k) = \sum_{j \in \mathscr{N}_i(k)} p_{ij} = \sum_{j \in \mathscr{V}} p_{ij} \cdot [\mathbf{A}(k)]_{ij}$.

Chapter 3

Distributed Consensus Algorithms

A consensus algorithm is an iterative process in which the network nodes compute in a totally distributed way a common value of interest. In each time step of the consensus process, one or several nodes communicate with one or several neighboring nodes. The data obtained throughout these communications is mixed with the current data of the node and this determines the value of convergence. In particular, if the value to be computed is the average of the initial data, the algorithm is called the average consensus algorithm. This distributed philosophy avoids the necessity of forwarding any information to one or several sink nodes, avoiding congestion around them and increasing then the robustness of the network.

Consider a network composed of N nodes indexed with $i = 1, \ldots, N$. Each of these N nodes contains an associated value x_i, which is generally initialized with the data sensed from the environment. This value is generally known as the state of the node i. Therefore, consensus is achieved if this state is the same for all of the network nodes, that is, $x_i = x_j \; \forall \; i, j \in \mathscr{V}$. Consensus algorithms can be classified according to the value computed. When the application requires only a global agreement and the value reached is not important, the algorithm is denoted unconstrained. On the other hand, when the application requires the computation of a specific function of the initial measurements, the algorithm is denoted constrained. In this Thesis we focus on a constrained consensus algorithm, where the function to be obtained is the average of the initial values.

The distributed implementation of inference tasks in WSNs entails that

nodes have only access to local information, and communicate with just one-hop neighbors. The most common solution is to split the global problem in simple local tasks, each one consisting of a first step that involves local computation followed by a refinement through cooperation between nodes. In this way, the decomposition of the main task in separable functions that can be computed locally by nodes and executed in parallel has traditionally been a popular topic in the computer science community [27]. In general, given both a network of N nodes, where each i has access to generic local information given by the vector ς_i, and a global objective function $f_0(\varsigma_1, \cdots, \varsigma_N)$ to be computed in a distributed fashion, the goal is to express the function f_0 as follows:

$$f_0(\varsigma_1, \cdots, \varsigma_N) = \sum_{i=1}^{N} f_i(\varsigma_i)$$

Therefore, if each node i takes $x_i(0) = N f_i(\varsigma_i)$ as its initial value, the objective function can be computed as the average of these values. Alternatively, when the objective function can be decomposed in a product of the form:

$$f_0(\varsigma_1, \cdots, \varsigma_N) = \prod_{i=1}^{N} \gamma_i(\varsigma_i)$$

the average of the initial values $x_i(0) = N \log(\gamma_i(\varsigma_i))$ results in $\log(f_0)$. It turns out that many statistical problems (Maximum Likelihood Estimator, Neyman Pearson detector, optimization by ADMM, etc.) can be partially cast or decomposed in some of the proposed methods. Hence, as long as all the nodes are able to compute the average of their initial values, the whole network can accomplish the global task. As explained before, a common approach to achieve this in a distributed fashion is to apply algorithms of consensus, where the N nodes of the network aim at computing the average of their initial values by successive exchanges of information with one hop neighbors in an iterative scheme.

As we explained in the previous chapter, the graphs and their associated spectral properties are determined by how the communications are performed between the network nodes. The same occurs with the convergence properties of consensus algorithms.

3.1 Consensus in Time-invariant Graphs

A time-invariant graph is the mathematical abstraction of a network of nodes where the communications, at each time instant, are always performed between the same pairs of nodes, that is, the same set of links is activated at every communication step. In practice, this implies to have several control mechanisms to ensure that these communications are always correctly performed, so that, every link is activated before a new communication step is started.

3.1.1 Undirected Graphs

Let us assume that nodes have some initial data at time instant $t = 0$. We collect them in an initial vector $\mathbf{x}(0)$, whose average is $\mathbf{x}_{\text{avg}} = \frac{\mathbf{1}\mathbf{1}^T\mathbf{x}(0)}{N}$, where $\mathbf{1}$ denotes the all ones column vector. The most simple consensus approach is the general linear update of the state of all sensors at each time instant t, given by:

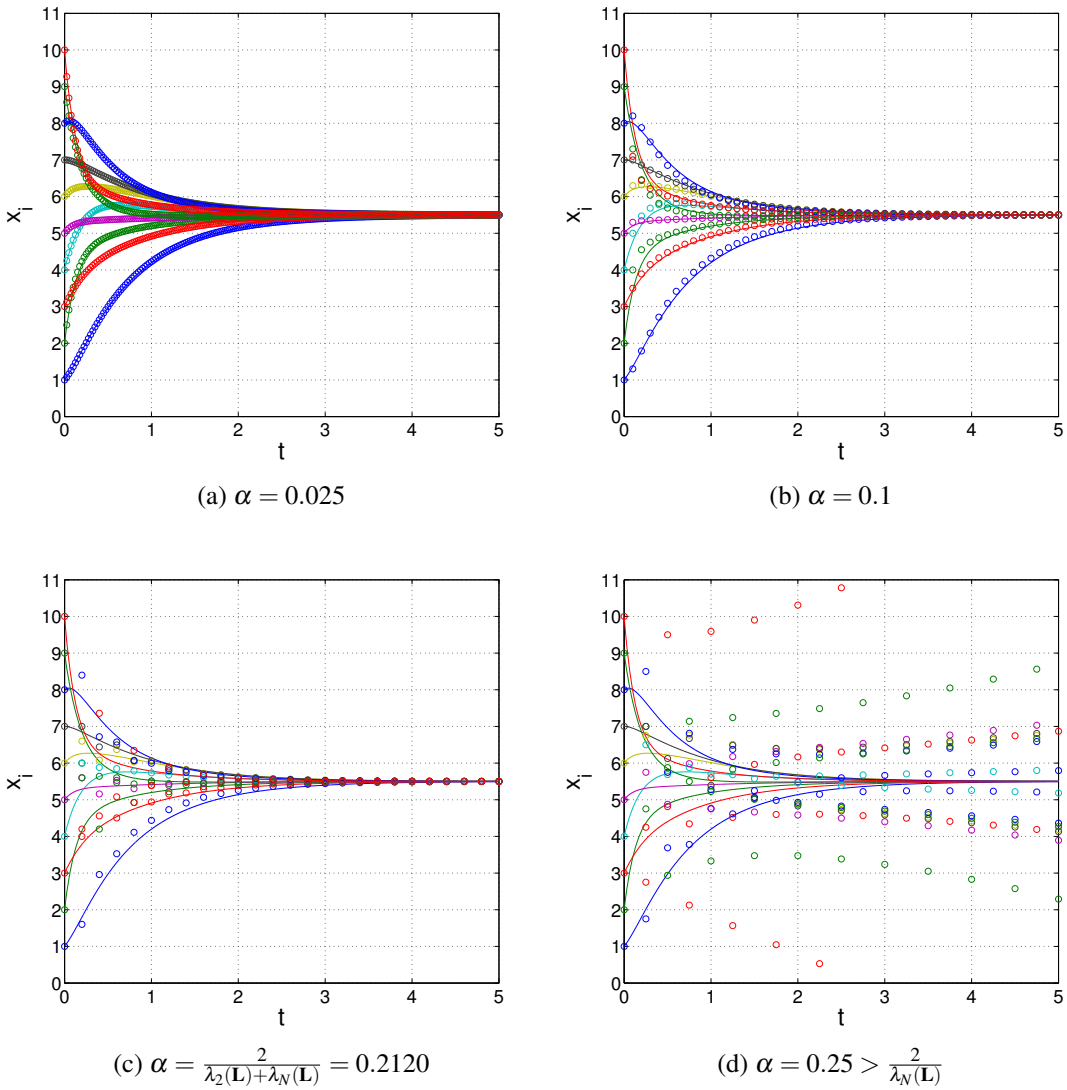
$$\dot{\mathbf{x}}(t) = -\mathbf{L}\mathbf{x}(t) \tag{3.1.1}$$

If the graph is undirected, the matrix $\mathbf{L}$ is positive semidefinite by construction, and the convergence to the average of the iterative process described in (3.1.1) is then ensured. The convergence time $t(\mathbf{L})$ of this linear update can be defined as the time required by the second smallest eigenvalue of the matrix $\mathbf{L}$ to be reduced by a certain factor $\rho < 1$. This is equivalent [65] to reduce the disagreement of $\mathbf{x}(t)$ with respect to $\mathbf{x}_{\text{avg}}$ by this same factor ρ. Accordingly, convergence time can be expressed as:

$$t(\mathbf{L}) = -t_s \frac{\log(\rho)}{\lambda_2(\mathbf{L})} \tag{3.1.2}$$

where t_s is the duration of a time slot unit.

Considering the continuous system defined in (3.1.1) and a small enough time step Δt implies that $\Delta\mathbf{x} = -\mathbf{L}\mathbf{x}(t)\Delta t$ and then, $\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - \mathbf{L}\mathbf{x}(t)\Delta t$ or equivalently $\mathbf{x}(t+\Delta t) = (\mathbf{I} - \Delta t\mathbf{L})\mathbf{x}(t)$. If we discretize this continuos system by discretizing the instants at which the state of the nodes can be



(a) $\alpha = 0.025$

(b) $\alpha = 0.1$

(c) $\alpha = \frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})} = 0.2120$

(d) $\alpha = 0.25 > \frac{2}{\lambda_N(\mathbf{L})}$

Figure 3.1.1: State $\mathbf{x}_i$ of $N = 10$ nodes as a function of time t. Solid lines represent the evolution of the continuous system in an average consensus process. Circles represent the instants (iterations) where the discrete system is updated. When α is kept small, the discrete system coincides with the continuous one (a). As α increases the discrete system is no longer able to follow the evolution of the continuous system, but still converges to the average (b) and (c). When $\alpha > \frac{2}{\lambda_N(\mathbf{L})}$, the discrete system diverges (d).

updated, we have the following evolution equation:

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k) \tag{3.1.3}$$

where the discrete concept of iteration k is introduced. This implies that we can consider the general linear update of the state of each sensor i at iteration

k, using only local data exchange, namely:

$$\mathbf{x}_i(k+1) = \sum_{j=1}^{N} [\mathbf{W}]_{ij}\mathbf{x}_j(k) \;\; \forall i = 1,2...N \tag{3.1.4}$$

where $x_i(k)$ denotes the value of node i at time instant k and $[\mathbf{W}]_{ij}$ is the entry of the matrix $\mathbf{W}$ used by node i to weight the data received from node j. Thus, matrix $\mathbf{W}$ denotes the mixing matrix:

$$\mathbf{W} = \mathbf{I} - \alpha\mathbf{L} \tag{3.1.5}$$

where $\alpha \approx \Delta t$ is the so-called step-size of the process and gives the relation between discrete iterations and continuous time. In particular, larger values of α imply that more continuous time separates two consecutive discrete iterations (see Figure 3.1.1).

The expression in (3.1.5) ensures that $\mathbf{W}$ is doubly stochastic[1], satisfying the convergence conditions $\mathbf{W1} = \mathbf{1}$ and $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T$ [88]. However, to ensure convergence, an extra condition is needed, namely, $\rho\left(\mathbf{W} - \frac{\mathbf{11}^T}{N}\right) < 1$, where $\rho(\mathbf{M})$ denotes the spectral radius of matrix $\mathbf{M}$, that is, $\rho(\mathbf{M}) = \max\limits_{\{1\leq i\leq N\}} |\lambda_i(\mathbf{M})|$. This last condition is satisfied if:

$$0 < \alpha < \frac{2}{\lambda_N(\mathbf{L})} \tag{3.1.6}$$

Since $\rho\left(\mathbf{W} - \frac{\mathbf{11}^T}{N}\right) = \max\{\lambda_2(\mathbf{W}), -\lambda_N(\mathbf{W})\}$ and $\lambda_i(\mathbf{W}) = 1 - \alpha\lambda_i(\mathbf{L})$, then it holds that $\rho\left(\mathbf{W} - \frac{\mathbf{11}^T}{N}\right) = \max\{1 - \alpha\lambda_2(\mathbf{L}), \alpha\lambda_N(\mathbf{L}) - 1\}$ and thus, the values for α satisfying (3.1.6) ensure that $\rho\left(\mathbf{W} - \frac{\mathbf{11}^T}{N}\right) < 1$.

By using again the relation between the eigenvalues of matrices $\mathbf{L}$ and $\mathbf{W}$, we can express the convergence time in (3.1.2) as follows:

$$t(\mathbf{A}, \mathbf{W}) = -t_s \frac{\alpha(\mathbf{A}) \log(\rho)}{1 - \lambda_2(\mathbf{W})} \tag{3.1.7}$$

where we have made explicitly the dependence of α on the matrix $\mathbf{A}$.

[1]Note that we are considering here undirected graphs, so that both matrices $\mathbf{A}$ and $\mathbf{W}$ are symmetric.

Differentiation between Continuous and Discrete Systems

As we increase the value of α and it separates from zero and approaches the upper limit $\frac{2}{\lambda_N(\mathbf{L})}$, systems in (3.1.1) and (3.1.4) become no longer equivalent (see Figure 3.1.1). Then, the convergence time of the system in (3.1.4) must be redefined. Let us first rewrite the system defined in (3.1.4) as follows:

$$\mathbf{x}(k) = \mathbf{W}^k \mathbf{x}(0) \tag{3.1.8}$$

The distributed linear iteration (3.1.8) converges to $\mathbf{x}_{\text{avg}}$ if and only if $\lim_{k\to\infty} \mathbf{W}^k = \frac{\mathbf{1}\mathbf{1}^T}{N}$. Accordingly, the asymptotic convergence factor can be defined by:

$$r(\mathbf{W}) = \sup_{\mathbf{x}(0)} \lim_{k\to\infty} \left(\frac{||\mathbf{x}(k) - \mathbf{x}_{\text{avg}}||_2}{||\mathbf{x}(0) - \mathbf{x}_{\text{avg}}||_2} \right)^{1/k}$$

which has been shown [88] to be equal to $\rho\left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right)$. Given the matrix $\mathbf{W}$ in (3.1.5), the optimal expression of the step-size α for minimizing $\rho\left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right)$ and satisfying (3.1.6) is given by:

$$\alpha = \frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})} \tag{3.1.9}$$

which makes $1 - \alpha\lambda_2(\mathbf{L}) = \alpha\lambda_N(\mathbf{L}) - 1$ and then $\rho\left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right) = \lambda_2(\mathbf{W})$.

Then, the convergence time $t(\mathbf{W})$ in a discrete system can be defined by:

$$t(\mathbf{W}) = \frac{-1}{\log(\lambda_2(\mathbf{W}))} \tag{3.1.10}$$

which gives the (asymptotic) number of steps for the error to decrease by a factor $\frac{1}{e}$.

3.1.2 Directed Graphs

Directed and undirected graphs are the abstraction of networks following different implementation philosophies. Since directed graphs do not require any control mechanism to maintain their structure, these are found in lightweight scenarios, where what is sought is to keep the operation of nodes simple. Oppositely, undirected graphs appears when it is more important to satisfy some application requirements than having a simple operation. The use of ACKs and other control mechanisms become reasonable in this later case.

Continuous system: Lets consider first the system defined in (3.1.1), whose convergence is totally characterized by the Laplacian matrix. This is a matrix with the sum of all of its rows equal to zero and with a zero eigenvalue $\lambda_1(\mathbf{L}) = 0$. This zero eigenvalue corresponds to the eigenvector $\mathbf{1}$ because it belongs to the null-space of $\mathbf{L}$, that is $\mathbf{L1} = \mathbf{0}$. Since $\mathscr{G}$ is strongly connected, $rank(\mathbf{L}) = n-1$ and $\mathbf{L}$ has a zero eigenvalue $\lambda_1(\mathbf{L}) = 0$ with algebraic multiplicity of one. Based on the Gersgorin disk theorem, the rest of the eigenvalues of $\mathbf{L}$ have positive real-parts and therefore the linear system in (3.1.1) is stable. On the other hand, any equilibrium of the system (3.1.1) is a right eigenvector of $\mathbf{L}$ associated to $\lambda_1(\mathbf{L}) = 0$. Since the eigenspace associated to the zero eigenvalue is one-dimensional, there exists a value $c \in \mathbb{R}$ such that the consensus has the form $c\mathbf{1}$. Thus, for a strongly connected graph, no mater if this is directed or undirected, the convergence is always globally exponentially stable [65]. However, this does not guarantee whether the convergence value $c\mathbf{1}$ of nodes is equal to $\mathbf{x}_{\text{avg}}$ or not. In fact, this value c may significantly differ from the average of the initial values. A sufficient condition to ensure $c\mathbf{1} = \mathbf{x}_{\text{avg}}$ is that the graph is balanced ($\mathbf{1}^T\mathbf{L} = \mathbf{0}$), which is always satisfied for undirected graphs, but not for general digraphs.

Discrete system: Lets consider now, a system that evolves according to (3.1.4) with a weight matrix given by $\mathbf{W} = \mathbf{I} - \alpha\mathbf{L}$. In a general case, according to the expression in (3.1.4), the convergence of $\mathbf{x}(k)$ to a vector of the form $\mathbf{x}_{\text{avg}}$ depends on the existence of $\lim_{k\to\infty}\mathbf{W}^k = \frac{\mathbf{1}\mathbf{1}^T}{N}$, which again is independent of the initial set of values $\mathbf{x}(0)$. Given that $\mathbf{W}$ is diagonalizable with eigenvalues $\lambda_i(\mathbf{W})$, $i = 1,\ldots,N$. The weight matrix can be eigen-decomposed as $\mathbf{W} = \mathbf{U}\Lambda\mathbf{V}^{-1}$, where $\mathbf{U}$ is a nonsingular matrix whose columns are the right eigenvectors $\mathbf{u}_1, \mathbf{u}_2 \ldots \mathbf{u}_N$. Λ is a diagonal matrix with the N eigenvalues of

$\mathbf{W}$ arranged in non-increasing order of magnitude, and the columns of $\mathbf{V}$ are the left eigenvectors $\mathbf{v}_1, \mathbf{v}_2 \ldots \mathbf{v}_N$ of $\mathbf{W}$. Substituting $\mathbf{W}$ for its factorization we can compute:

$$\mathbf{W}^k = \sum_{i=1}^{N} \lambda_i^k(\mathbf{W}) \mathbf{u}_i \mathbf{v}_i^T \tag{3.1.11}$$

where the equality $\mathbf{v}_i^T \mathbf{u}_i = 1$ is satisfied for all i.

Let us consider now the spectral radius of $\mathbf{W}$ defined as:

$$\rho(\mathbf{W}) = \max_i |\lambda_i(\mathbf{W})|, i = 1..N$$

If $\rho(\mathbf{W}) > 1$, then $\mathbf{W}^k$ does not converge as k increases because the sum in (3.1.11) grows unbounded, and therefore $\mathbf{x}(k)$ in (3.1.4) cannot converge to a vector of the form $c\mathbf{1}$. Alternatively, if $\rho(\mathbf{W}) < 1$, then $\mathbf{W}^k$ converges to an $N \times N$ zero matrix as k increases and $\mathbf{x}(k)$ converges to the zero vector. However, the algorithm converges for the case $\lambda_1(\mathbf{W}) = \rho(\mathbf{W}) = 1$, implying that $|\lambda_i(\mathbf{W})| < 1, i = 2 \ldots N$ must be satisfied for all i. Therefore, $\mathbf{x}(k)$ converges to $c\mathbf{1}$ for any set of initial values $\mathbf{x}(0)$ if and only if $\mathbf{W}$ satisfies $\mathbf{W1} = \mathbf{1}$ and $\rho\left(\mathbf{W} - \frac{\mathbf{1u}_1^T}{N}\right) < 1$. However, this does not ensure convergence to the average value. Note that, according to (3.1.11) the consensus value is given by $c = \mathbf{v}_1^T \mathbf{x}(0)$. In order to reach the average of the initial values, the left eigenvector associated to $\lambda_1(\mathbf{W})$ must be $\mathbf{u}_1 = \mathbf{1}$, such that, as k tends to infinity, we have $\mathbf{W}^k = \frac{\mathbf{11}^T}{N}$. In other words, the matrix $\mathbf{W}$ has largest eigenvalue 1 with algebraic multiplicity one, with associated left and right eigenvectors $\mathbf{1}$. The condition $\mathbf{1}^T \mathbf{W} = \mathbf{1}^T$ implies that the topology is balanced.

The three conditions to ensure convergence to the average are again:

$$\mathbf{W1} = \mathbf{1}, \quad \mathbf{1}^T \mathbf{W} = \mathbf{1}^T, \quad \rho\left(\mathbf{W} - \frac{\mathbf{11}^T}{N}\right) < 1 \tag{3.1.12}$$

The first and third conditions can be easily satisfied for directed and undirected topologies if the matrix structure $\mathbf{W} = \mathbf{I} - \alpha\mathbf{L}$ is chosen and a strongly connected graph is ensured. However, while the second condition is directly satisfied for undirected graphs, in directed graphs additional considerations must be taken into account to ensure that the topology is balanced.

Review of Existing Weight Matrix Expressions

Once continuous and discrete systems are decoupled, the weight matrix may present different expressions. Depending on the application, the communication links may be whether directed or undirected. Some expressions assume that nodes have global information available while other expressions can be implemented using local information gathered directly by the nodes through cooperation. The expression can be also chosen depending on the connectivity of the network, i.e., whether it is time-varying or fixed. When the topology is fixed, the expression is chosen to satisfy certain convergence conditions. Oppositely, when the topology is time-varying, every instantaneous weight matrix should satisfy that same conditions, which is not always possible.

Max-degree weights: A common expression for the matrix $\mathbf{W}$ in graphs with time-invariant topology consists in assigning a weight on each edge equal to the maximum out-degree of the network [88], that is:

$$[\mathbf{W}]_{ij} = \begin{cases} 1/d_{\max}^{\text{out}} & \text{if } e_{ij} \in \mathscr{E} \\ 1 - d_i^{\text{out}}/d_{\max}^{\text{out}} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{3.1.13}$$

where $d_{\max}^{\text{out}} = \max_{i \in \mathscr{V}} \{d_i^{\text{out}}\}$. Note that the value of $d_{\max}^{\text{out}}$ must be determined and broadcasted across the network before running the consensus algorithm. This is an example of a distributed scheme using global information.

Local-degree weights: Another expression for the matrix $\mathbf{W}$ consists of assigning a weight on each link equal to the largest out-degree of its two incident nodes as follows [88]:

$$[\mathbf{W}]_{ij} = \begin{cases} 1/\max\{d_i^{\text{out}}, d_j^{\text{out}}\} & \text{if } e_{ij} \in \mathscr{E} \\ 1 - \sum_{j \neq i} [\mathbf{W}]_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{3.1.14}$$

For this expression, each node requires knowledge of the out-degrees of all its neighboring nodes. Similarly to the previous example, this information must be known by the nodes before running the consensus algorithm and can be exchanged locally at the beginning of the algorithm. This is an example of a distributed scheme using local information.

3.2 Consensus in Time-varying (Random) Graphs

When the communication links of a network vary over time, the connectivity is characterized by a dynamic graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}(k))$, where the set of edges $\mathscr{E}(k)$ varies over time while the set of nodes $\mathscr{V}$ remains constant. The consensus mechanism on a network with a variable topology becomes a linear switching system:

$$\dot{\mathbf{x}}(t) = -\mathbf{L}(t)\mathbf{x}(t) \tag{3.2.1}$$

First, we assume at any time instance, the network topology is a balanced digraph (or undirected graph) that is strongly connected. Let us denote $\lambda_2((\mathbf{L}(t) + \mathbf{L}(t)^T)/2)$ by $\lambda_2(\mathbf{L}(t))$ for a topology dependent Laplacian $\mathbf{L}(t)$.

Suppose every graph considered is a balanced digraph that is strongly connected and let $\lambda_2^* = \min_k\{\lambda_2(\mathbf{L}(t))\}$. Then, the agents asymptotically reach an average-consensus [65] for all initial states with a speed faster or equal to λ_2^*.

To obtain more detailed results for time-varying graphs, we need to focus on the discrete consensus algorithm in (3.1.4), which can be rewritten as follows:

$$\mathbf{x}(k) = \prod_{l=0}^{k} \mathbf{W}(k-l)\mathbf{x}(0) \tag{3.2.2}$$

Clearly, the convergence of $\mathbf{x}(k)$ to a vector of the form $c\mathbf{1}$ is determined by the convergence of the product $\prod_{l=0}^{k} \mathbf{W}(k-l)$ to a rank one matrix:

$$\lim_{k\to\infty} \prod_{l=0}^{k} \mathbf{W}(k-l) = \mathbf{1}\mathbf{v}_1^T \tag{3.2.3}$$

The convergence above is ensured if all the matrices involved in the product satisfy the first and the third convergence conditions in (3.1.12). In addition, if all the weight matrices satisfy the second convergence condition in (3.1.12), $\mathbf{x}(k)$ converges to the average consensus vector $\mathbf{x}_{\text{avg}}$. It is important to notice that the state vector still converges to a consensus although the condition $\rho(\mathbf{W} - \mathbf{1}^T\mathbf{v}_1^T)$ is not satisfied for every iteration k, provided that the weight matrices have row-sums equal to one.

Types of Convergence

Given the system in (3.2.2) and depending on the scheme of communications being used, different forms of convergence for vector $\mathbf{x}(k)$ are utilized:

- Sure convergence - We say that $\mathbf{x}(k)$ converges to the vector $\mathbf{x}_{\text{avg}}$ if:

$$\lim_{k\to\infty} \mathbf{x}(k) = \mathbf{x}_{\text{avg}}$$

- Almost sure convergence - We say that $\mathbf{x}(k)$ converges almost surely -or with probability one- to $\mathbf{x}_{\text{avg}}$ if:

$$\Pr\left\{\lim_{k\to\infty} \mathbf{x}(k) = \mathbf{x}_{\text{avg}}\right\} = 1$$

  Almost sure convergence is also denoted as strong convergence.

- Convergence in expectation - We say that $\mathbf{x}(k)$ converges to $\mathbf{x}_{\text{avg}}$ in expectation if:

$$\lim_{k\to\infty} \mathbb{E}[\mathbf{x}(k)] = \mathbf{x}_{\text{avg}}$$

- Mean square convergence - We say that $\mathbf{x}(k)$ converges to $\mathbf{x}_{\text{avg}}$ in the mean square sense if:

$$\lim_{k\to\infty} \mathbb{E}[||\mathbf{x}(k) - \mathbf{x}_{\text{avg}}||_2^2] = 0$$

  where $||\cdot||_2$ denotes the 2-norm.

- Convergence in probability - We say that $\mathbf{x}(k)$ converges to $\mathbf{x}_{\text{avg}}$ in probability if

$$\Pr\left(\frac{||\mathbf{x}(k) - \mathbf{x}_{\text{avg}}||_2^2]}{||\mathbf{x}(0)||_2^2} > \delta\right) \leq \delta \tag{3.2.4}$$

  for all $k \geq t(\mathbf{W}, \delta)$, where $t(\mathbf{W}, \delta)$ is the first integer such that it is accomplished (3.2.4).

Almost sure convergence implies convergence in probability. Convergence in the mean square sense implies also convergence in probability.

3.2.1 Undirected Graphs: Gossip Algorithms

Gossip algorithms are an interesting variation of consensus algorithms where an asynchronous random mechanism is introduced. As opposite to consensus algorithms, a gossip scheme randomly activates at each iteration one or several nodes, which communicate with only one of its neighbors at a time so that an information exchange is performed. The random way in which nodes are activated and communicate in this scheme with each other involve a network topology that varies over time. Gossip algorithms were first introduced by Boyd et al. [31], which analyzes the properties of this distributed process, providing important analytical results about its performance in terms of convergence time and communication cost.

Given a network of N nodes and assuming that each node i has some initial measurement x_i, the average gossip algorithm allows every node to estimate the global average with certain accuracy. In this process, no central entity is available so that the average has to be obtained by only using local information and local communications. Most of the relevant work in gossip algorithms, in order to ensure convergence to the average value almost surely, requires the following: i) a strongly connected undirected graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ in which all the instantaneous subgraphs are included $\mathscr{G}(k) \subset \mathscr{G}$, ii) a symmetric $N \times N$ matrix $\mathbf{W}(k)$, at each iteration k, such that $[\mathbf{W}(k)]_{ij} = 0$ if $[\mathbf{A}(k)]_{ij} = 0$ and iii) $\mathbf{W}(k)\mathbf{1} = \mathbf{1}$, $1^T\mathbf{W}(k) = \mathbf{1}^T$.

In this random scheme, the non-zero entries of $\mathbf{W}(k)$ are generated by randomly activating, at each time instant k, an undirected edge, which determines the nodes that pairwise exchange and mix information at that iteration. The information is mixed according to a fix real number $\alpha \in (0,1)$, which is again the step size of the process. This scheme leads to the nodes i and j to produce a new state according to the equations:

$$x_i(k+1) = (1-\alpha)x_i(k) + \alpha x_j(k) \tag{3.2.5}$$

$$x_j(k+1) = \alpha x_i(k) + (1-\alpha)x_j(k) \tag{3.2.6}$$

and the state of the rest of nodes remains unaltered.

Thus, this simple random scheme leads all the nodes to reach the average consensus value $\mathbf{x}_{\text{avg}}$ almost surely.

By analyzing the convergence of this algorithm in probabilistic terms, interesting bounds on its convergence time can be obtained. In this context, convergence time refers to how many steps the value of $\mathbf{x}(k)$ is close to $\mathbf{x}_{\text{avg}}$ with high probability. For this purpose, lets define $\mathbf{e}_i$ as the all-zero vector with exception of its i-th entry which is set to one. Now, lets us assume that at iteration k, nodes i and j are randomly chosen to exchange data. Accordingly, the corresponding instantaneous weight matrix $\mathbf{W}(k)$ can be defined as follows:

$$\mathbf{W}(k) = \mathbf{I} - \frac{(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T}{2}$$

Thus, for any pair-wise exchange between any two nodes i and j, it is accomplished that $\mathbb{E}[\mathbf{W}^T\mathbf{W}] = \overline{\mathbf{W}}$, since:

$$\begin{aligned}
\mathbf{W}(k)^T\mathbf{W}(k) &= \left(\mathbf{I} - \frac{(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T}{2}\right)^2 \\
&= \left(\mathbf{I} - \frac{(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T}{2}\right) \\
&= \mathbf{W}(k)
\end{aligned}$$

Observe that this implies that $\overline{\mathbf{W}}$ is positive semidefinite, since it is also the expected value of positive semidefinite matrices $\mathbf{W}^T\mathbf{W}$. So the spectral radius of $\overline{\mathbf{W}} - \frac{\mathbf{1}\mathbf{1}^T}{N}$, which governs the rate of convergence of $\mathbb{E}[\mathbf{x}(k)]$, is simply the second largest eigenvalue of $\overline{\mathbf{W}}$.

In this particular case, the averaging time has been shown [31] to be bounded as follows:

$$\frac{0.5\log\delta^{-1}}{\log\lambda_2(\overline{\mathbf{W}})^{-1}} \leq t(\overline{\mathbf{W}},\delta) \leq \frac{3\log\delta^{-1}}{\log\lambda_2(\overline{\mathbf{W}})^{-1}} \tag{3.2.7}$$

This is the time for which on any sample path, the values at the nodes are all δ-close to the average value with probability greater than $1-\delta$.

The basic gossip algorithm in [31] has been improved in [41] by introducing geographic information and then refined in [26]. These two new approaches provide a convergence time and a communication cost of smaller order than the basic gossip approach.

3.2.2 Directed Graphs: Asymmetric Gossip

In an asymmetric scenario, all the instantaneous graphs $\mathscr{G}(k) \subset \mathscr{G}$ are directed. The weight matrix $\mathbf{W}(k)$, at each iteration k, is asymmetric and only the condition $\mathbf{W}(k)\mathbf{1} = \mathbf{1}$ is fulfilled. Finally, the data is mixed according to a fix real number $\alpha \in (0,1)$.

Let us consider first the simplest asymmetric gossip scheme:

$$x_j(k+1) = \alpha x_i(k) + (1-\alpha)x_j(k),\ \forall j \in \mathscr{N}_i(k) \tag{3.2.8}$$

The remaining nodes in the network, including the transmitter node i, do not update their state value, that is:

$$x_\ell(k+1) = x_\ell(k),\ \forall \ell \notin \mathscr{N}_i(k) \tag{3.2.9}$$

This is the simplest asymmetric scheme for gossiping, which is studied in [45]. As explained before, this scheme reaches consensus to a value that may significantly differ from the average.

The Push-sum Algorithm

In order to ensure probabilistic average consensus under asymmetric communications, several algorithms have been proposed in the related literature. Most of them are based on the push-sum algorithm, which was initially presented in [58], it was latter generalized in [25] to different weights, and was finally applied to a broadcast scenario in [53]. In this algorithm, at each time k, every node i maintains a sum $s_i(k)$, initialized to $s_i(0) = x_i(0)$, and a weight $w_i(k)$, initialized to $w_i(k) = 1$. At time k, the estimate of node i is computed as $x_i(k) = s_i(k)/w_i(k)$. Assuming that at time k the node i wakes up, it broadcasts the pair: $\left(\frac{s_i(k)}{d_i(k)+1}, \frac{w_i(k)}{d_i(k)+1}\right)$. Each of the $d_i(k)$ nodes receiving the packet updates its state according to:

$$\begin{aligned} s_j(k+1) &= s_j(k) + \frac{s_i(k)}{d_i(k)+1} \\ w_j(k+1) &= w_j(k) + \frac{w_i(k)}{d_i(k)+1} \end{aligned} \tag{3.2.10}$$

and node i updates its own state according to:

$$\begin{aligned} s_i(k+1) &= \frac{s_i(k)}{d_i(k)+1} \\ w_j(k+1) &= \frac{w_i(k)}{d_i(k)+1} \end{aligned} \tag{3.2.11}$$

while the nodes not receiving the packet do not update their state. Using matrix notation, the algorithm can be rewritten as follows:

$$\begin{aligned} \mathbf{s}(k) &= \mathbf{M}(k-1)\mathbf{s}(k-1) = \mathbf{P}(k)\mathbf{x}(0) \\ \mathbf{w}(k) &= \mathbf{M}(k-1)\mathbf{w}(k-1) = \mathbf{P}(k)\mathbf{1} \end{aligned}$$

where $\mathbf{P}(k) = \mathbf{M}(k-1)\mathbf{M}(k-2)\ldots\mathbf{M}(0)$, and the individual sums and weights at time k are respectively collected in the vectors $\mathbf{s}(k)$ and $\mathbf{w}(k)$. Each matrix $\mathbf{M}(k)$ can be written as:

$$\mathbf{M}(k) = \mathbf{I} - \mathbf{L}(k)\left(\mathbf{I} + \mathbf{D}(k)\right)^{(-1)} \tag{3.2.12}$$

and provided that the total of the $d_i(k)$ nodes receive the packet, $\mathbf{M}(k)$ is a column stochastic matrix ($\mathbf{1}^T\mathbf{M}(k) = \mathbf{1}^T \ \forall k$). It means that for both vectors $\mathbf{s}(k)$ and $\mathbf{w}(k)$, the sum of the initial values is preserved at each iteration:

$$\begin{aligned} \sum_{i=1}^{N} s_i(k) &= \sum_{i=1}^{N} x_i(0) = N\mathbf{x}_{\text{avg}} \\ \sum_{i=1}^{N} w_i(k) &= N \end{aligned} \tag{3.2.13}$$

so there is no deviation from the average almost surely. Moreover, in [58] it is proved that the series $(\mathbf{M}(k)_{k\geq 1})$ is weak ergodic. It means that as k grows, $\mathbf{P}(k)$ tends to have identical columns (which may vary in k), and the nodes achieve a common value ($x_i(k) = x_j(k) \ \ \forall i, j$). All these features imply that, with no packet losses, the protocol provides probabilistic average consensus. The accuracy of the method relies on the fact that each node knows, a priori, how many nodes receive correctly its packet.

3.2.3 Directed Graphs and Lossy Links: MSE Convergence

The push-sum-based algorithms developed in [25] and [53] are able to reach average consensus even in the presence of random and asymmetric communications. However, if this asymmetry is caused by packet losses (collision due to interferences, fading, etc.), these algorithms tend to deviate from the target value. In the presence of packet losses, where the convergence is to a random value is interesting to analyze the deviation in terms of the MSE:

$$\text{MSE}(\mathbf{x}(k)) = \frac{1}{N}\mathbb{E}\left[||\mathbf{x}(k) - \mathbf{x}_{\text{avg}}||_2^2\right] \tag{3.2.14}$$

By following [71] we can express (3.2.14) as:

$$\text{MSE}(\mathbf{x}(k)) = \frac{\sigma^2}{N}\text{tr}\left(\mathbf{R}_{\mathbf{w}}(k)\right) \tag{3.2.15}$$

where $\mathbf{R}_{\mathbf{w}}(k) = \mathbb{E}[\mathbf{M}_{\mathbf{w}}^T(k)\mathbf{M}_{\mathbf{w}}(k)]$, and by expanding this term we have the following:

$$\text{MSE}(\mathbf{x}(k)) = \frac{\sigma^2}{N}\text{tr}\left(\mathbf{R}_{\mathbf{w}}(k-1)\mathbf{C}_{\mathbf{w}}\right)$$

with $\mathbf{C}_{\mathbf{w}} = \mathbb{E}[\mathbf{W}(k)\mathbf{W}^T(k)]$ which can be expressed as:

$$\mathbf{C}_{\mathbf{w}} = \mathbf{I} - 2\alpha\overline{\mathbf{L}} + \alpha^2\mathbb{E}[\mathbf{L}(k)\mathbf{L}^T(k)]$$

For a probability $[\mathbf{P}]_{ij} = p\ \forall i, j \in \mathscr{V}$ equal for all the links, a closed expression can be obtained:

$$\text{MSE}(\mathbf{x}(k)) = \sigma^2\left(\frac{b}{1-a+b} - \frac{(a-1}{1-a+b}(a-b)^k\right) \tag{3.2.16}$$

where we have that $a = 1-2(N-1)p\alpha + (2(N-1)p + (n-1)(N-2)p^2)\alpha^2$ and $b = 2p\alpha - Np^2\alpha^2$. Finally, for different link probabilities a bound for the MSE can be derived:

$$\text{MSE}(k) \leq \frac{\sigma_0^2}{N}\left(N + (\text{tr}(\mathbf{C}_{\mathbf{w}}) - N)\frac{1-\lambda_1^{k-1}(\mathbf{C}_{\mathbf{w}})}{1-\lambda_1(\mathbf{C}_{\mathbf{w}})}\right) \tag{3.2.17}$$

3.3 Conclusions of the Chapter

In this chapter, we have provided an overview on consensus algorithms, explaining the different schemes of communications where these algorithms can be executed and how the underlying graph affects its convergence time. Despite essential progress, some issues in this area remain open. This partly pertains to a gap between necessary and sufficient conditions for consensus in continuous-time systems when time-varying topologies are considered. For discrete-time networks, the above gap was filled in the case of bidirectional interactions, which means that the graph is undirected but the couplings may be non-symmetric. The concerned criterion relies on the connectivity of the graph whose links connect nodes that interact infinitely many times. However this result cannot be applied even to sampled-data continuous-time systems since even exact discretization may destroy key companion requirements, e.g., strict convex hull shrinking or strict positivity of coupling gains.

Part II

Consensus in Time-invariant Graphs

Chapter 4

Topology Optimization: Continuous Systems

4.1 Introduction

In consensus-based applications [36] [55] [94], the convergence time (3.1.2) and the power consumption (2.4.1) have been identified as the most important performance metrics, which are both determined by the network topology [24][75]. The convergence speed determines the number of time steps needed to reach consensus. In each of these steps, because of the inter-node communication, there is an associated power consumption, which increases in value with the path loss exponent and the transmission range of each node. This power consumption maintained over time leads to certain energy consumption associated to the consensus process.

Let us consider the factor ρ in (3.1.2) to be small enough, then the total energy consumption of a node i after a single consensus process is the product between the consumed power p_i at each time step[1] and the convergence time $t(\mathbf{L})$. Formally, we have the following:

$$\mathscr{E}_i(\mathbf{A}) = t(\mathbf{L})\mathrm{p}_i = K\frac{\mathrm{p}_i}{\lambda_2(\mathbf{L})} \tag{4.1.1}$$

where K is a parameter value that groups all the irrelevant constants, i.e., $K = -t_s \log(\rho)$.

[1] Note that for a time-invariant graph, it is accomplished that $\mathrm{p}_i = \mathrm{p}_i(1) = \ldots = \mathrm{p}_i(k)\ \forall i \in \mathscr{V}$.

Moreover, since the nodes in these networks usually operate on batteries, this energy consumption per consensus process has a decisive bearing on the lifetime of the network. Regardless of the application being considered, the concept of network lifetime is associated to the time until one or several nodes run out of battery [38][90][93]. Thus, it is not only important the energy consumed by the whole network, but also the energy consumed by individual nodes. In an average consensus process, every node must participate to obtain the global average, which is no longer possible as soon as the first node runs out of battery. In this case, a unique node, the one consuming more energy, would determine the network lifetime. However, once the first node i runs out of battery, the remaining nodes are still able to reach consensus to a certain value while instantaneous connectivity between them is guaranteed. Thus, the network lifetime may alternatively imply several nodes.

Given that every node presents certain energy capability C_i, the number of consensus processes that can be executed before node i runs out of battery can be expressed as:

$$\mathscr{L}_i(\mathbf{A}) = \left\lfloor \frac{\mathrm{C}_i}{\mathscr{E}_i} \right\rfloor = \left\lfloor \frac{\lambda_2(\mathbf{L})\mathrm{C}_i}{K\mathrm{p}_i} \right\rfloor \tag{4.1.2}$$

Therefore, in several applications, minimizing the energy consumption at network and node levels is more appropriate than minimizing only the convergence time. This minimization, including the number of time steps to reach consensus and their associated power consumption due to communications, should be tackled by optimizing the network topology. Although, the search for the optimal topology is very well-known to be a NP-hard combinatorial problem [76], efficient heuristic solutions can be still used in this context.

In this chapter, we propose the optimization of the network topology in order to minimize certain function of the energy consumption of the nodes. In particular, we show how the cost function considered and the capabilities of these nodes determine the type of network graph obtained when applying the proposed optimization. Moreover, we present two signal processing applications where our topology optimization methodology can be exploited to satisfy certain properties of the application. Several numerical results are shown to illustrate the efficiency of our methodology.

$f(\mathbf{p})$	Definition	Description
$[\mathbf{p}]_i$	p_i	Power consumption of node i
$[\mathscr{O}(\mathbf{p})]_i$	$\mathscr{O}_i$	i-th largest power consumption of nodes
$\mathrm{sum}(\mathbf{p})$	$\mathrm{p}_{\mathrm{total}} = \sum_{i=1}^{N} \mathrm{p}_i$	Total power consumption of the network
$\max(\mathbf{p})$	$\mathrm{p}_{i_{\max}} = \max_{i \in \mathscr{V}} \{\mathrm{p}_i\}$	Maximum power consumption per node
$S_m(\mathbf{p})$	$\mathrm{p}_{m_{\max}} = \sum_{i=1}^{m} \mathscr{O}_i$	Summation of the m-largest powers of nodes

Table 4.1.1: List of energy functions considered in this chapter.

4.2 Topology Optimization: General Formulation

In this chapter, we focus on the optimization of the network topology, given by the matrix $\mathbf{A}$, which is our main optimization variable. As a cost function, we consider any relevant combination of the entries of the vector in (2.4.1) and the convergence time in (3.1.2). In general, our problem (**P0**) can be cast as follows:

$$\begin{aligned} \min._{\{\mathbf{A}\}} \quad & \frac{f(\mathbf{p}(\mathbf{A}))}{\lambda_2(\mathbf{L}(\mathbf{A}))} \\ \text{s. t.} \quad & \xi \leq \lambda_2(\mathbf{L}(\mathbf{A})) \\ & [\mathbf{A}]_{ij} = [\mathbf{A}]_{ji} \ \ \forall i, j \in \mathscr{V} \\ & [\mathbf{A}]_{ij} \in \{0,1\} \ \text{ if } \ \textit{condition1} \\ & [\mathbf{A}]_{ij} = 0 \ \text{ if } \ \textit{condition2} \end{aligned}$$

where $f(\cdot)$ can be any convex function and ξ is an arbitrary small positive constant to ensure that the resulting value of $\lambda_2(\mathbf{L}(\mathbf{A})$ is greater than zero. The list of functions $f(\cdot)$ considered in this chapter can be seen in Table 4.1.1.

condition1: The entries of the topology matrix should be zero or one values. This values correspond to the existence (a value of one) or the no existence (a value of zero) of a link connecting two nodes. This is a non-convex constraint, which we should relax to obtain a convex problem.

condition2: Depending on the maximum transmission range of each node, some links may not be established, so that the corresponding entries of matrix $\mathbf{A}$ are set to zero, reducing considerably the number of variables.

Some Background on Convex Optimization

Convex optimization is a special class of mathematical optimization that can be solved numerically very efficiently. There are several concepts common to every optimization problem, which most of them have already appeared in **P0**. Lets identify and define some of them:

objective function: is either a cost function or energy function which is to be minimized, or a reward function or utility function, which is to be maximized. In our problem we have energy functions of the form $\frac{f(\mathbf{p}(\mathbf{A}))}{\lambda_2(\mathbf{L}(\mathbf{A}))}$. The expressions in (4.1.1) and (4.1.2) are two examples of functions with this structure.

optimization variable: is the variable of the problem on which the objective function depends. In our problem our main optimization variable is $\mathbf{A}$.

constraints: set of conditions for the optimization variables that are required to be satisfied. These are topological constraints in our problem, which are given by the nodes capabilities.

optimal solution: is a feasible solution that has the smallest objective value among all matrices $\mathbf{A}$ that satisfy the constraints.

solution method: is an algorithm that computes a solution of the problem (to some given accuracy), given a particular problem from the class, i.e., an instance of the problem.

In particular, the objective function $\frac{f(\mathbf{p}(\mathbf{A}))}{\lambda_2(\mathbf{L}(\mathbf{A}))}$ is a convex-concave fractional function, since we have a convex function divided by a concave one. The function $f(\cdot)$ is convex since we are restricting it to be any convex combination of the entries of vector $\mathbf{p}(\mathbf{A})$, which is an affine function on $\mathbf{A}$. The function $\lambda_2(\mathbf{L}(\mathbf{A}))$ is well known to be concave, since this is the minimum of a linear function (eigenvalue) of $\mathbf{L}(\mathbf{A}) + \mathbf{1}\mathbf{1}^T$, which is a matrix with the spectrum equivalent to $\mathbf{L}(\mathbf{A})$.

Model	Set of devices	Communications	Static/Dynamic	Attachment
SWG	heterogeneous	unicast	static	no
SFG	heterogeneous	unicast	dynamic	yes
RGG	homogeneous	broadcast	static	no

Table 4.2.1: Main structural properties that characterize three well-known graph models.

4.3 Topology Optimization in Complex Networks

The frequent appearance of large-scale networks in reality has motivated the research of complex networks, i.e., networks whose structure is irregular, complex and that even dynamically evolves in time, with the main focus on systems with thousands or millions of nodes. These large-scale networks in the real world include transportation networks, phone call networks, the Internet and the World Wide Web, as well as systems of interest in biology and medicine, as neural networks or genetic, metabolic and protein networks.

The comparative analysis of such real networks from different fields has produced several interesting results, being most of them structural. In particular, the research on complex networks has involved new concepts and measures to characterize the topology of real networks, such as the identification of a series of unifying principles and statistical properties common to most of them, see Table 4.2.1.

Alternatively, a further novel topic concerns spatial networks. While most of the early works on complex networks have focused on the characterization of the topological properties, the spatial aspect has received less attention, when not neglected at all. However, it is common that the topology could be constrained by geographical criteria. For instance, the long range connections in a spatial network are constrained by the physical distance, which has important consequences on the statistical properties of the network.

In this section, we focus on the optimization of the topology in terms of several energy related functions that are relevant for consensus algorithms when are executed over large-scale sensor networks. Then, we identify three relevant real scenarios where this solution can be applied by introducing small changes in the formulation, relating certain network and nodes capabilities to the appearance of well-known graph models.

In order to formulate our optimization problem, we first explain how com-

mon properties of sensor networks can be represented in matrix form. Firstly, we introduce a matrix $\mathscr{P}$ representing the cost to communicate any two nodes in the network, i.e., $[\mathscr{P}]_{ij}$ depends on the distance r_{ij}^{γ} between nodes i and j. If these communicate over a wireless medium, we have $2 \leq \gamma \leq 4$. Alternatively, if nodes i and j communicate through a wired connection, we have $\gamma = 1$. Second, we define a matrix $\mathscr{C}$ representing the energy capabilities of the nodes, where $[\mathscr{C}]_{ij} = 1/\mathrm{C}_i \; \forall j$. Finally, a third matrix $\mathscr{A}$ representing the possible affinity for the pre-existing nodes in the network can be defined. This matrix is only needed when not all the nodes are available at the very beginning, and new nodes are added along time. Later on, we explain how this can be tackled appropriately in a real setting. We combine all this information in two different matrices by applying the Hadamard product, so we can define $\mathscr{W}_1 = \mathscr{P} \odot \mathscr{C} \odot \mathscr{A}$ and $\mathscr{W}_2 = \mathscr{P} \odot \mathscr{A}$, which are the costs associated to any link to be activated, that is, any entry of matrix $\mathbf{A}$ to be equal to one. Then, we can define the following cost function for a unicast scenario:

$$\kappa \max_{i \in \mathscr{V}} \left\{ \sum_{j \in \mathscr{V}} [\mathscr{W}_1]_{ij} [\mathbf{A}]_{ij} \right\} + (1-\kappa) \sum_{i \in \mathscr{V}} \sum_{j \in \mathscr{V}} [\mathscr{W}_2]_{ij} [\mathbf{A}]_{ij} \tag{4.3.1}$$

and the cost function, for a broadcast scenario, can be defined as follows:

$$\kappa \max_{i,j \in \mathscr{V}} \{ [\mathscr{W}_1]_{ij} [\mathbf{A}]_{ij} \} + (1-\kappa) \sum_{i \in \mathscr{V}} \max_{j \in \mathscr{V}} \{ [\mathscr{W}_2]_{ij} [\mathbf{A}]_{ij} \} \tag{4.3.2}$$

In these two equations (4.3.1) and (4.3.2), the parameter κ is a constant between 0 and 1 that controls the trade off between the two terms in each equation. Let us denote these terms as $C_{\max}$ and C_{total} respectively. Then, the general optimization problem can be formulated as follows (**P1**):

$$\begin{array}{ll} \min._{\{\mathbf{A}\}} & \kappa \frac{C_{\max}(\mathbf{A})}{\lambda_2(\mathbf{L}(\mathbf{A}))} + (1-\kappa) \frac{C_{\text{total}}(\mathbf{A})}{\lambda_2(\mathbf{L}(\mathbf{A}))} \\ \text{s. t.} & \xi \leq \lambda_2(\mathbf{L}(\mathbf{A})) \\ & \mathbf{A} = \mathbf{A}^T \\ & [\mathbf{A}]_{ij} \in \{0,1\} \;\; \forall i,j \in \mathscr{V} \\ & [\mathbf{A}]_{ij} = 0 \;\text{ if }\; r_{ij} > R_i \;\; \forall i,j \in \mathscr{V} \end{array}$$

where ξ is an arbitrary small positive constant to ensure that the resulting value of $\lambda_2(\mathbf{L}(\mathbf{A})$ is greater than zero and R_i is the maximum distance to

which node i can correctly communicate. The variation of κ gives the Pareto-optimal points of **P1**. The objective of **P1** is to find the entries of the adjacency matrix $\mathbf{A}$ that minimize the multi-objective function:

$$\kappa \frac{C_{\max}(\mathbf{A})}{\lambda_2(\mathbf{L}(\mathbf{A}))} + (1-\kappa)\frac{C_{\text{total}}(\mathbf{A})}{\lambda_2(\mathbf{L}(\mathbf{A}))} \tag{4.3.3}$$

where $\frac{C_{\max}(\mathbf{A})}{\lambda_2(\mathbf{L}(\mathbf{A}))}$ represents the node with minimum lifetime scaled by its scale-free property. Similarly, $\frac{C_{\text{total}}(\mathbf{A})}{\lambda_2(\mathbf{L}(\mathbf{A}))}$ represents the total energy consumption altered by the scale-free property of every node. Notice that in (4.3.3) we have made explicit the dependence of $\mathbf{L}(\mathbf{A})$, $C_{\max}(\mathbf{A})$ and $C_{\text{total}}(\mathbf{A})$, on $\mathbf{A}$.

As stated before, we consider the possibility that nodes are added to the network along time, allowing to add $n = 1, 2, \cdots, N$ nodes at a time in a pre-existing network composed by $m_0 = 0, 1 \cdots (N-1)$. This is summarized in **Algorithm 1**, which has to be executed for any addition of nodes.

Algorithm 1 Node addition based on **P1**

Require: κ, m_0, n, $\mathscr{P}$, $\mathscr{C}$, $\mathscr{A}$, $\mathbf{A}_{ini}$ and $R_i \; \forall i \in \mathscr{V}$
Ensure: $\mathbf{A} \in \mathbb{R}^{m_0+n \times m_0+n}$
$\%$ $\mathscr{P}$, $\mathscr{C}$ and $\mathscr{A}$ are $(m_0+n) \times (m_0+n)$ matrices

$$\begin{aligned}
\min._{\{\mathbf{A}\}} \quad & \kappa \frac{C_{\max}(\mathbf{A})}{\lambda_2(\mathbf{L}(\mathbf{A}))} + (1-\kappa)\frac{C_{\text{total}}(\mathbf{A})}{\lambda_2(\mathbf{L}(\mathbf{A}))} \\
\text{s. t.} \quad & \xi \leq \lambda_2(\mathbf{L}(\mathbf{A})) \\
& \mathbf{A} = \mathbf{A}^T \\
& [\mathbf{A}]_{ij} \in \{0,1\} \mid i > m_0, \;\; i,j \in \mathscr{V} \\
& [\mathbf{A}]_{ij} = 0 \text{ if } r_{ij} > R_i \;\; \forall i,j \in \mathscr{V} \\
& \mathbf{A} = \begin{bmatrix} \mathbf{A}_{ini} & \mathbf{A}_1 \\ \mathbf{A}_2 & \mathbf{A}_3 \end{bmatrix}
\end{aligned}$$

$\%$ $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ are $m_0 \times n$, $n \times m_0$ and $n \times n$ matrices
$\mathbf{A}_{ini} = \mathbf{A}$

Problem **P1** is a combinatorial one because of the binary variables constraint. In order to obtain a polynomial time solvable problem, we introduce a relaxation consisting on assuming the entries of $\mathbf{A}$ to be real variables between 0 and 1, which results in a fractional convex-concave problem. To solve it, we introduce the following function [42]:

$$h(\mu) = \min\{\kappa C_{\max}(\mathbf{A}) + (1-\kappa)C_{\text{total}}(\mathbf{A}) - \mu\lambda_2(\mathbf{L}(\mathbf{A}))\}$$

This function allows us to solve the original problem **P1** by applying **Algorithm 2**, which is based on repeatedly solving the following problem (**P2**):

$$\begin{array}{ll} \min._{\{\mathbf{A}\}} & \kappa C_{\max}(\mathbf{A}) + (1-\kappa) C_{\text{total}}(\mathbf{A}) - \mu \lambda_2(\mathbf{L}(\mathbf{A})) \\ \text{s. t.} & \xi \leq \lambda_2(\mathbf{L}(\mathbf{A})) \\ & \mathbf{A} = \mathbf{A}^T \\ & 0 \leq [\mathbf{A}]_{ij} \leq 1 \ \ \forall i, j \in \mathscr{V} \\ & [\mathbf{A}]_{ij} = 0 \ \text{ if } \ r_{ij} > R_i \ \ \forall i, j \in \mathscr{V} \end{array}$$

which can be easily obtained from **P1** by applying standard optimization tools [32]. Moreover, since $\lambda_2(\mathbf{L}(\mathbf{A}))$ is a concave function of matrix $\mathbf{A}$, the problem **P2** can be shown to be convex. Thus, applying the Dinkelbach's algorithm [42], we are able to obtain the optimal value of the parameter μ, which is equivalent to the optimal value of the relaxation of the original problem **P1**, as described by **Algorithm 2**.

The value of the parameter ε determines the precision of this algorithm, that is, how close its solution is from the optimal one of the relaxation of the problem **P1**. The result obtained from it are the entries $[\mathbf{A}]_{ij}$ of the adjacency matrix, which define the network topology. However, due to the relaxation procedure, these matrix coefficients are real variables belonging to the interval $[0,1]$, instead of binary values that determine the presence or absence of a specific link. Thus, in order to obtain a real network topology, entries with smaller values than a predefined threshold a_{th} are removed. For a given solution, the choice of a_{th} determines the final topology. As a_{th} approaches 0, high connectivity is obtained. On the other hand, for values of a_{th} above certain value, the network becomes disconnected.

The solution matrix $\hat{\mathbf{A}}$ that results from the previous projection procedure is feasible for the original problem, which it turns to be a lower bound in the cost function considered. The difference between the upper and the lower bounds is called the gap. The gap is always nonnegative and if it is zero, then $\hat{\mathbf{A}}$ is actually optimal for the original problem. Not much can be said about the gap, in general; for example, there are no generic useful bounds on how large it can be. The gap is, however, very useful when evaluated for a given problem instance.

In the following subsections we present three network scenarios that are

Algorithm 2 Solves relaxed version of **P1**

Require: ε
Ensure: $\kappa C_{\max}(\mathbf{A}) + (1-\kappa)C_{\text{total}}(\mathbf{A}) - \mu\lambda_2(\mathbf{L}(\mathbf{A})) \leq \varepsilon$
  Set matrix $\mathbf{A}$ as a feasible solution
  while $\kappa C_{\max}(\mathbf{A}) + (1-\kappa)C_{\text{total}}(\mathbf{A}) - \mu\lambda_2(\mathbf{L}(\mathbf{A})) > \varepsilon$ **do**
    set μ as $\frac{\kappa C_{\max}(\mathbf{A})+(1-\kappa)C_{\text{total}}(\mathbf{A})}{\lambda_2(\mathbf{L}(\mathbf{A}))}$
    Solve **P2** with the current μ
  end while

motivated by the need of providing new sensing capabilities in an industrial environment, e.g. desalination plant, water treatment plant, etc. [1][3]. In order to meet the specifications, multiple devices are required to sense the industrial environment magnitudes so that the diverse plants processes are improved by the early detection of important events, management of data uncertainty and proper actuation. In these application scenarios, three general network settings can be identified. These include nodes with different communication and energy supply capabilities that in some cases must be integrated with existing infrastructure of the industrial plant. Thus, to ensure energy efficient in-network processes under these network settings, the topology optimization must be adapted appropriately to each of them. Moreover, depending on the environment and nodes characteristics, the appearance of certain graph models: small world graphs (SWG) [86], scale free graphs (SFG) [2] and random geometric graphs (RGG) [69] can be identified, which present different performance in practice.

4.3.1 Heterogeneous Sensor Networks

The first network scenario arises from the necessity of acquiring accurate situational awareness (SA) knowledge in an heterogeneous sensor network. For this purpose, different sensing tasks to collect enough data are needed. Since this large amount of raw data lacks of significance, several inference tasks to add important information are additionally necessary. In particular, to carry out these tasks of collecting and pre-processing sensed data, it is common to use multiple sensor nodes to collect the raw data, plus several micro-servers with higher capabilities that are in charge of performing the inference tasks.

The former nodes are powered by batteries, while the latter are connected to the electricity grid. Although all nodes can communicate only wirelessly, the transmission range of the micro servers is significantly longer than the rest of nodes, allowing the appearance of shortcuts[2] in the network. We are implicitly assuming that R_i is not the same for all nodes.

The setting described above is an example of an heterogeneous network consisting of N nodes with different communication and energy supply capabilities. The former determines the variables in our optimization problem. These variables are the entries of the topology matrix $\mathbf{A}$ and define which pair of nodes can bidirectionally communicate and which cannot. Moreover, some nodes in the network present extra energy supply capabilities, so that, $[\mathscr{C}]_{ij} = 1/\mathrm{C}_i,\ \forall j \in \mathscr{V}$.

In this first setting, unicast communications are assumed, where the power consumption required to communicate nodes i and j depends on the distance r_{ij} that separates them. For simplicity, we assume a simplified path loss model, so that, $p_{ij} = p_{\min} r_{ij}^{\gamma}$, where γ is the path loss exponent and $p_{\min}$ is the minimum power required to decode successfully a message.

The corresponding optimization problem, taking into account both total energy consumption and network lifetime, is:

$$
\begin{array}{ll}
\min._{\{s,\mathbf{A}\}} & s \\
\text{s. t.} & \kappa \frac{\mathrm{p}_1(\mathbf{A})}{\mathrm{C}_1} + (1-\kappa)\mathrm{p}_{\text{total}}(\mathbf{A}) - \mu\lambda_2(\mathbf{L}(\mathbf{A})) \leq s \\
& \vdots \\
& \kappa \frac{\mathrm{p}_N(\mathbf{A})}{\mathrm{C}_N} + (1-\kappa)\mathrm{p}_{\text{total}}(\mathbf{A}) - \mu\lambda_2(\mathbf{L}(\mathbf{A})) \leq s \\
& \xi \leq \lambda_2(\mathbf{L}(\mathbf{A})) \\
& \mathbf{A} = \mathbf{A}^T \\
& 0 \leq [\mathbf{A}]_{ij} \leq 1 \ \ \forall i,j \in \mathscr{V} \\
& [\mathbf{A}]_{ij} = 0 \ \text{ if } \ r_{ij} > R_i \ \ \forall i,j \in \mathscr{V}
\end{array}
$$

which is the epigraph form of **P2** with an all-ones affinity matrix $\mathscr{A}$. Although neither growing nor preferential attachment behavior are present in this first setting, we have considered the different transmission and energy capabilities of the nodes.

[2]The appearance of shortcuts (long distance links) tend to reduce the number of hops between multiple pairs of nodes.

4.3.2 Scaling Heterogeneous Sensor Networks

The second configuration arises from the necessity of extending the sensing functionality of an existing network, whose nodes sense and control certain specific points of an industrial plant. These pre-existing nodes are generally connected to the power grid, and can communicate with each other through a wired network. The purpose, in this setting, is to add low cost sensor nodes over time according to the requirements of the final user. The existing nodes with higher capabilities act as hubs and the new nodes may have some affinity for them. For example, the preexisting nodes may be in charge of reprogramming the network, changing the sampling rate, etc. Growing in time and preferential attachment are characteristics of the SFG model.

In a SFG, the probability of a node having degree k decays following a power law, $P(k) \propto k^{-h}$, $h > 1$. As a consequence, SFGs are characterized by being the outcome of a random growth process with a hub-like core structure [60]. This means that the few nodes having a huge number of links are highly connected between them (by means of both one hop and multiple hop routes). In [60], an indicator to measure the extent to which high-degree nodes are connected to other high-degree nodes is proposed, which is given by:

$$s(\mathscr{G}) = \sum_{i \in \mathscr{V}} \sum_{j \in \mathscr{N}_i} d_i d_j \tag{4.3.4}$$

The Barabási-Albert (BA) model, proposed in [2], reproduces the formation of such type of networks. This procedure is based on the formation of the World Wide Web, and incorporates both the growing character of the network and the preferential attachment. An undirected graph $\mathbf{G}^{BA}$ with N nodes is constructed as follows: starting with m_0 isolated nodes, at each step $s = 1, 2, 3, \ldots, N - m_0$ a new node j with $m \leq m_0$ links is added to the network. The probability that a link connects node j to an existing node i is linearly proportional to the current degree of i:

$$\prod_{j \longrightarrow i} = \frac{d_i}{\sum_{l \in \mathscr{V}} d_l} \tag{4.3.5}$$

At step s, the network will have $N = m_0 + s$ nodes and $K = m \cdot s$ links, with

an average degree $d_{\text{avg}} = 2m$.

Due to the signal attenuation, distances between nodes in a wireless network become essential when establishing the links, since longer distances imply higher powers. More precisely, the required average power for node i to reach a destination node j can be expressed by the simplified path loss model. According to it, the work in [91] updates the BA model by considering the length of the connections, and the probability in (4.3.5) becomes:

$$\prod_{j \longrightarrow i} = d_i {r_{ij}}^{-\alpha} \tag{4.3.6}$$

divided by the appropriate factor to normalize the probabilities. It can be noticed that neither of these two procedures take into account the algebraic connectivity. Consequently, topology related features of the resulting network, such as robustness to random failures of nodes and execution time of iterative processes, e.g. consensus, are far from optimal.

We propose a new growth model where the algebraic connectivity is taken into account when performing the attachment, while the scale-free property is maintained. Our problem, at this point, is to obtain for each incoming node the set of links that lead to the minimum associated energy of consensus and maximum network lifetime, including preferential attachment, which means that a new node has an affinity for the existing nodes in the network proportional to their degree.

Therefore, we first set up the problem **P1** globally and then we restrict it to each incoming node, leaving the rest fixed. The complete procedure is summarized in **Algorithm 3**. First, given N nodes randomly deployed in a square area, we divide this area into m_0 squares of the same size, assuming m_0 to be the square of an integer number and also that there exist at least one node in each of these squares. We randomly pick one node from each of the m_0 squares, in order to form the set of initial existing nodes of the BA model. We connect these nodes to form a fully connected graph. By doing so, we ensure that the network is always connected (and $\lambda_2(\mathbf{L}) > 0$). This initial topology gives us $\mathbf{A}_{ini}$, which is a feasible $m_0 \times m_0$ matrix. Then, for each incoming node $n = 1, 2, \cdots, N - m_0$, we solve the relaxed version of **P1** with a $(m_0 + n) \times (m_0 + n)$ matrix $\mathbf{A}$. In this new matrix, the only variables are the entries of the $(m_0 + n)$-th row and the $(m_0 + n)$-th column. The rest of the entries are

initialized with the values of the previous matrix $\mathbf{A}_{ini}$, and become constants for the current optimization problem, whose complexity is also reduced as a consequence. Then, the resulting (m_0+n)-th row and (m_0+n)-th column determine the importance of every possible link that can be established by the incoming node n. Finally, we project the solution choosing the best a_{th} and $\mathbf{A}_{ini}$ is updated with the new solution matrix $\mathbf{A}$.

Algorithm 3

Require: κ, m_0, m, p, $\mathscr{C}$, $\mathscr{A}$, $\mathbf{A}_{ini}$ and $R_i \; \forall i \in \mathscr{V}$
Ensure: $\mathbf{A} \in \mathbb{R}^{N \times N}$
$n = 1$
while $n \leq N - m_0$ **do**
$\mathbf{A}$ and $\mathbf{D}$ are $(m_0+n) \times (m_0+n)$ matrices

$$\begin{aligned} \text{min.}_{\{\mathbf{A}\}} \quad & \kappa C_{\max}(\mathbf{A}) + (1-\kappa) C_{\text{total}}(\mathbf{A}) - \mu \lambda_2(\mathbf{L}(\mathbf{A})) \\ \text{s. t.} \quad & \xi \leq \lambda_2(\mathbf{L}(\mathbf{A})) \\ & \mathbf{A} = \mathbf{A}^T \\ & \mathbf{a} \cdot \mathbf{1}^T \leq m \text{ or } \mathbf{a} \cdot \mathbf{1}^T = m \\ & 0 \leq [\mathbf{A}]_{ij} \leq 1 \;\mid\; i = m_0 + n, \;\; j \in \mathscr{V} \\ & [\mathbf{A}]_{ij} = 0 \;\text{ if }\; r_{ij} > R_i \;\; \forall i, j \in \mathscr{V} \\ & \mathbf{A} = \begin{bmatrix} \mathbf{A}_{ini} & \mathbf{a}_{[1:(m_0+n-1)]} \\ \mathbf{a}^T_{[1:(m_0+n-1)]} & a_{m_0+n,m_0+n} \end{bmatrix} \end{aligned}$$

% $\mathbf{a}$ is the (m_0+n)-th column vector of $\mathbf{A}$
FIND a_{th} that minimizes the cost function
$\mathbf{A}_{ini} = \mathbf{A} \geq a_{th}$
end while

4.3.3 Homogeneous Sensor Networks

In industrial environments [1], there are several processes which require some automation in order to reduce associated costs. Examples of such processes are the heating system, ventilation system, etc., that depending on environmental parameters such as temperature, the presence of human operators, etc. can vary its operating mode to reduce energy consumption. To tackle this problem, the deployment of a wireless sensor network (WSN) is proposed. Thus, our last problem consists of efficiently connecting a wireless network composed by nodes powered by batteries that only perform wireless communications. Thus, we analyze how these inherently broadcast communications affect the optimization methodology and the resulting graphs. The

optimization problem that provides such pareto-optimization is:

$$\begin{array}{ll} \min._{\{s,\mathbf{A}\}} & \kappa \mathrm{p}_{i_{\max}}(\mathbf{A}) + (1-\kappa)\mathrm{p}_{\mathrm{total}}(\mathbf{A}) - \mu\lambda_2(\mathbf{L}(\mathbf{A})) \\ \text{s. t.} & \xi \leq \lambda_2(\mathbf{L}(\mathbf{A})) \\ & \mathbf{A} = \mathbf{A}^T \\ & 0 \leq [\mathbf{A}]_{ij} \leq 1 \ \ \forall i,j \in \mathscr{V} \\ & [\mathbf{A}]_{ij} = 0 \ \text{ if } \ r_{ij} > R \ \ \forall i,j \in \mathscr{V} \\ & [\mathbf{A}]_{ij} \geq [\mathbf{A}]_{il} \ \text{ if } \ r_{ij} \leq r_{il} \ \ \forall i \in \mathscr{V}, \forall j,l \in \mathscr{N}_i \end{array}$$

where R denotes the maximum transmission range common to all nodes in the network. Since we are considering homogeneous networks where neither growing nor preferential attachment behavior are considered, then $C_{\mathrm{total}} = \mathrm{p}_{\mathrm{total}}$ and $C_{\max} = \mathrm{p}_{i_{\max}}$.

Thus, in order to obtain a valid network topology, we apply the same thresholding technique as in the previous approaches. However, in a broadcast scenario, this may lead to non-optimal solutions because the value of $\lambda_2(\mathbf{L}(\mathbf{A}))$ always increases or, at least, remains unaltered when one or several links are added to the existing topology. Then, given a set of transmission ranges with an associated power consumption, the energy is always minimized when all the links under these transmission ranges are established. To ensure all those links within the communication range are active in the final projected binary solution, we have additionally introduced the constraint $[\mathbf{A}]_{ij} \geq [\mathbf{A}]_{il}$ if $r_{ij} \leq r_{il} \ \ \forall i \in \mathscr{V}, \forall j,l \in \mathscr{N}_i$, which ensures to give in every node i a greater value to $[\mathbf{A}]_{ij}$ as shorter the link with node j is.

4.3.4 Numerical Results

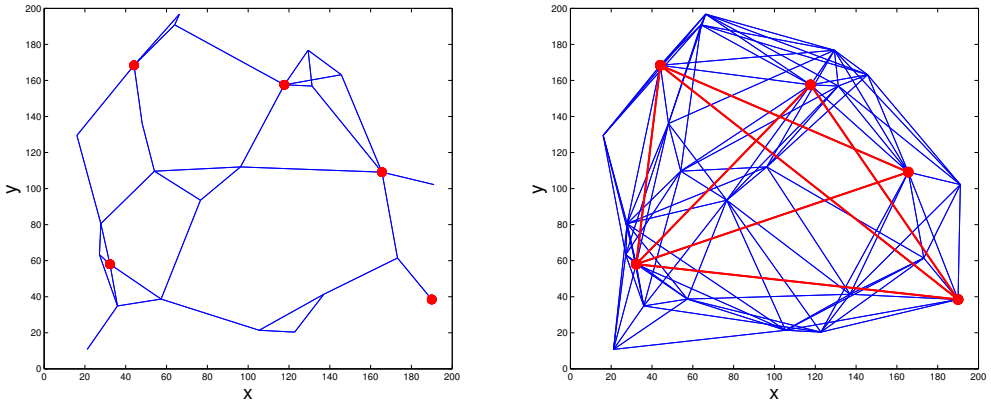
In this section, we present the numerical results associated to the different real settings found in [1]. In each setting, we apply the parameters that best fit the industrial environment. We assume $T_s = 1$ ms and the value of a_{th} that minimizes the cost function considered. Since the location of the nodes depend on the deployment area, the coordinates of the nodes are assumed to be generated randomly. Then, the results are averaged from 100 different random deployments. The power that each sensor node i requires to wirelessly send its message and reach a given neighbor j is computed by applying a simplified path loss model.

Heterogeneous sensor networks

Our first setting includes 20 sensor nodes and 5 micro-servers. Each micro-server is formed by a sensor node connected to a PC and acts as a cluster head of the closest nodes by helping on multiple tasks, for a list of them check [1]. Here, we focus on the collaboration of all these nodes (sensor nodes and micro-servers) to achieve average consensus, which serves to illustrate the performance of our topology optimization methodology. The area to cover has been defined to be approximately $L = 200$ meters side, corresponding to different phases of the filtrate in the water processing plant, where the accumulation of bacteria must be measured (among other parameters). Finally, a minimum power $p_{\min} = 1.5 \times 10^{-9}$ mW. required at the nodes to decode the signal is needed, such that, $p_{ij} = p_{\min} r_{ij}^{\gamma}$, expressed in mW.

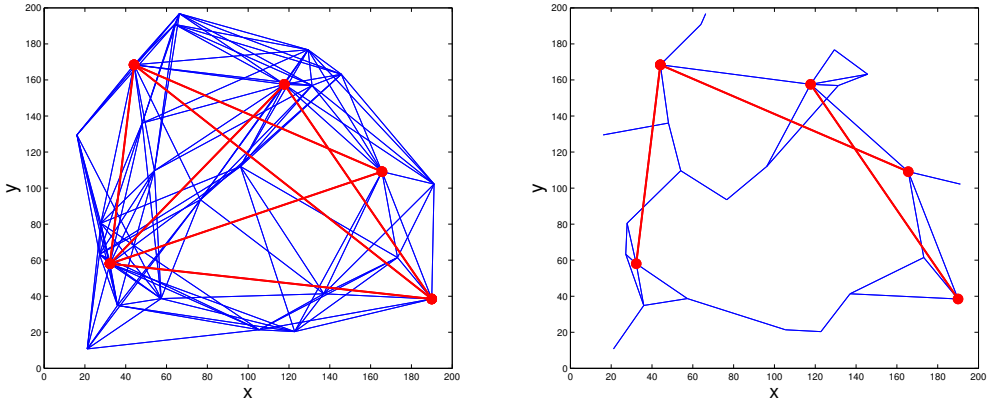
Figure 4.3.1 illustrates how the parameters κ and γ affect the graph resulted from applying the proposed optimization methodology. In this setting, a graph presents the characteristic shortcuts of a small world network whenever gamma is small enough regardless of the value of κ. This is because these shortcuts have a low energy cost as compared to the increasing effect on $\lambda_2(\mathbf{L})$, leading to reduce the total energy consumption and the maximum energy consumption per node. On the other hand, if γ presents large values, these links are created only when the network lifetime takes more importance than the total energy consumption. This is because the micro-servers are connected to the power grid and then are able to spend more energy on communications to connect between them without reducing their lifetime and making the convergence faster.

Figure 4.3.2 shows the averaged results in terms of energy consumption resulted from applying our optimization methodology to the settings proposed at the beginning of this subsection. The x-axis represents the applied threshold and the y-axis represents the associated energy magnitude. Since a value of the path loss exponent $\gamma = 3$ represents accurately the attenuation of the environment in which the final deployment is performed, it is the one chosen to generate Figure 4.3.1. Moreover, $\gamma = 3$ presents the most significant results in terms of topological variability. For values of $\gamma = 2$, the resulting graphs are very dense, while the value of $\gamma = 4$ results in sparse graphs, which both mask, in part, the resulting graph properties.



(a) $\gamma = 3$, $\kappa = 0$

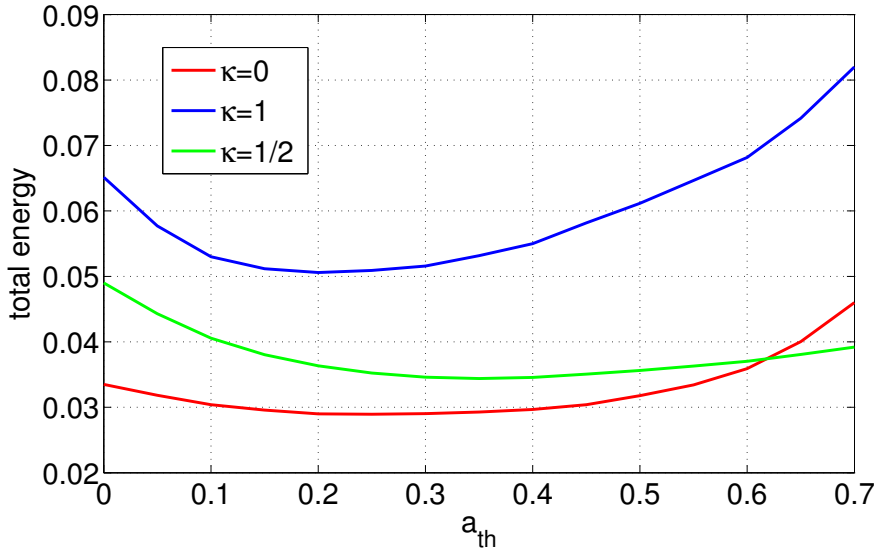
(b) $\gamma = 3$, $\kappa = 1$



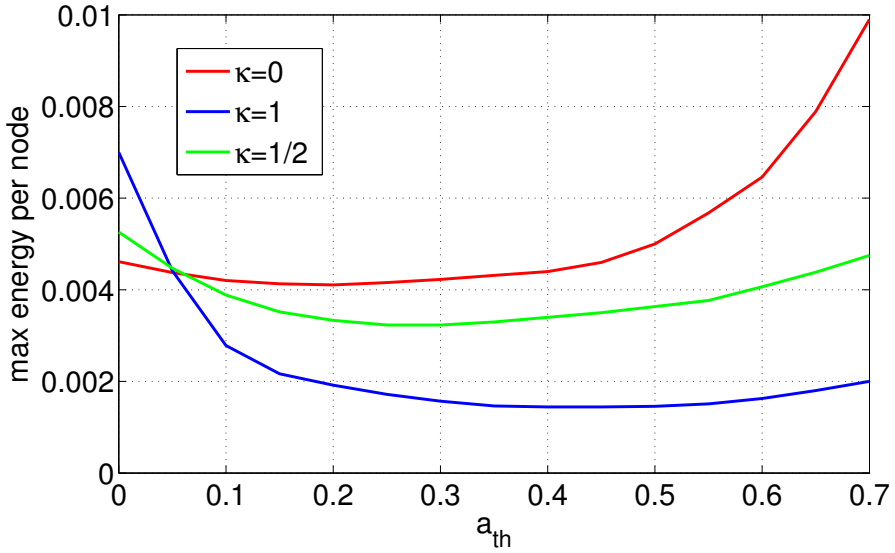
(c) $\gamma = 2$, $\kappa = \frac{1}{2}$

(d) $\gamma = 4$, $\kappa = \frac{1}{2}$

Figure 4.3.1: Illustrative example showing how the parameters γ and κ affect the optimization problem solution. Four different graphs composed by $N = 25$ nodes have been generated by applying **Algorithm 2** and choosing the optimal value of a_{th}. Red solid lines and circles represent shortcuts and nodes with extra energy supply capabilities respectively.



(a) Total energy consumption



(b) Maximum energy consumption per node

Figure 4.3.2: Influence of a_{th} on the total energy consumption and the maximum energy per node is shown for different values of κ and $\gamma = 3$.

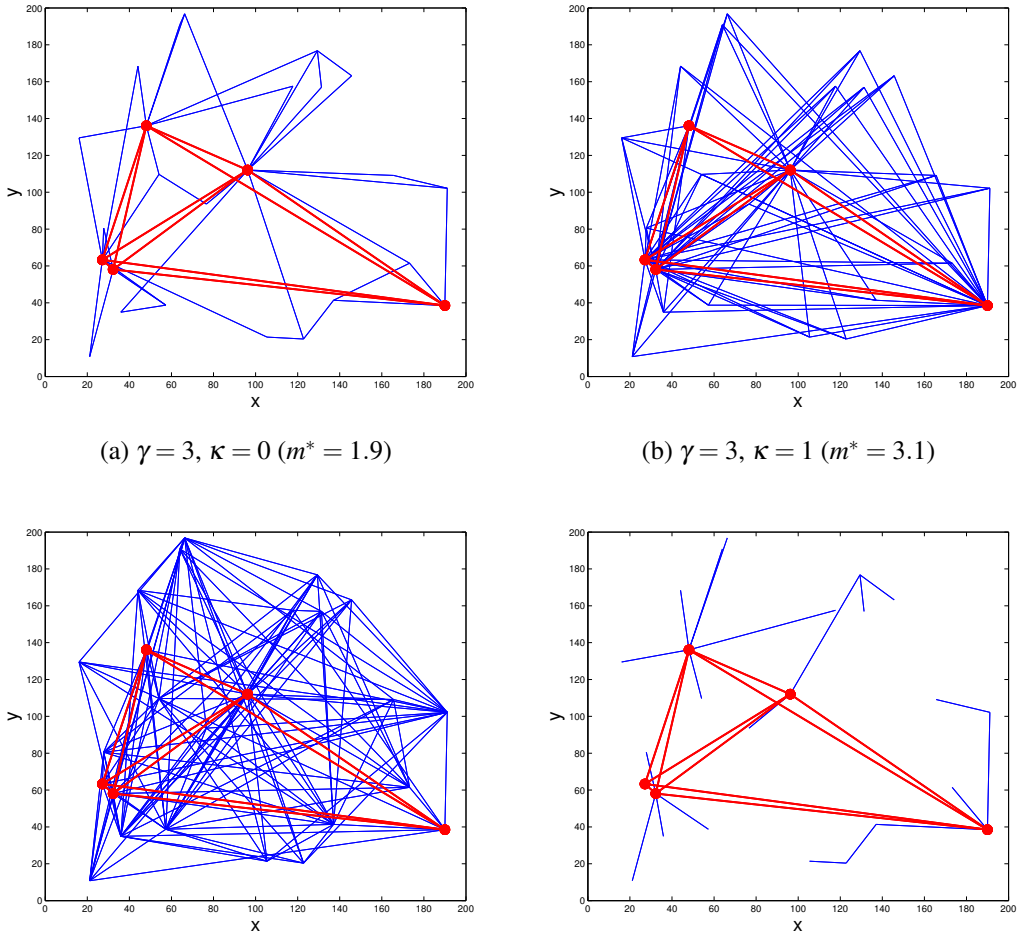
Scaling heterogeneous sensor networks

In this second case, we assume that a network infrastructure of $m_0 = 5$ nodes exists in the deployment area. These pre-existing nodes are characterized for being connected to both a power grid and a wired network and are in charge of communicating the sensor system with the central management point. To illustrate how our optimization methodology works in a network growing in time, we add, one by one, 20 extra sensor nodes.

The number m of links established by each new sensor node can be fixed ($m = 1, \cdots m_0$), as in most of the existing techniques [2][91], or it can be the result of our algorithm, which is the value that minimizes the cost function considered. This lead to certain average optimal value m^* at the end of the process. In this subsection, we first show several topologies obtained with the general constraint of $m \leq m_0$ and then we solve the problem for fixed values of the parameter m, evaluating the impact of this parameter on both the total energy consumption and the maximum energy consumption per node.

Figure 4.3.3 illustrates the influence of the parameters κ and γ in the structure of the resulting graph. On the one hand, the value of γ determines the length and the number of established links. If the value of γ is small enough, many long links are created favoring the formation of the SFG structure. Conversely, when γ takes large values, the high cost of the links implies that the network structure depends on the energy parameters to be optimized (consensus energy and network lifetime). For example, when what is sought is to maximize the network lifetime, the links connecting any node to a hub have less associated metric cost than the rest, favoring the formation of hubs. Oppositely, if what is sought is to minimize the total energy consumption, all the links contribute according to their lengths and less hubs are formed.

Figure 4.3.4 also shows the influence of the number of established links m in the final solution. It can be easily seen that the total energy consumption and the maximum energy per node are highly influenced by m. In this case, in order to minimize the former parameter, it is better to keep a moderately connected network, by constraining the number of links that each node establishes. On the contrary, in order to minimize the maximum energy consumption per node, it is better to let the nodes to establish the maximum number of available connections ($m = m_0$).



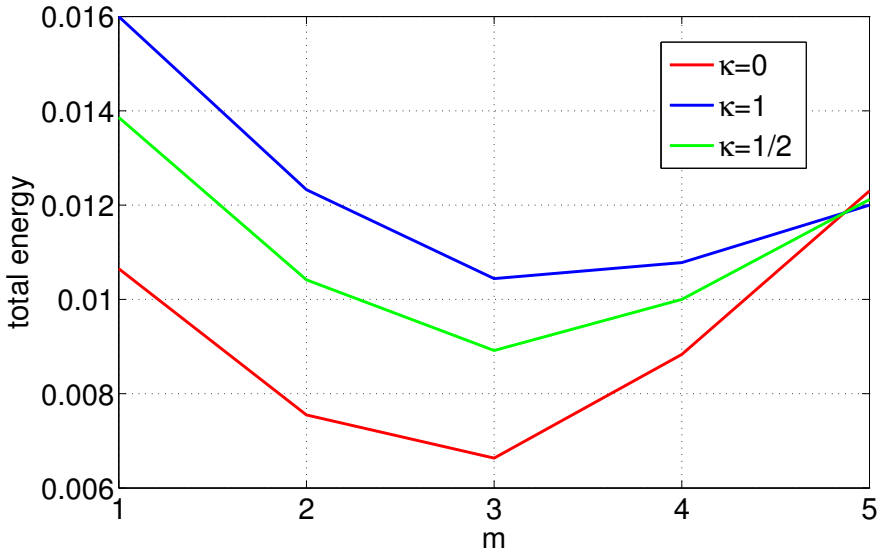
(a) $\gamma = 3$, $\kappa = 0$ ($m^* = 1.9$)

(b) $\gamma = 3$, $\kappa = 1$ ($m^* = 3.1$)

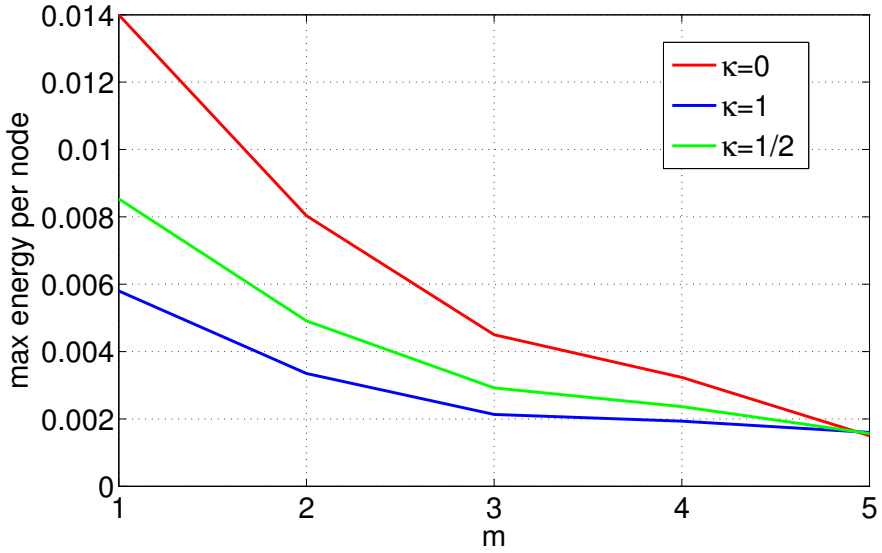
(c) $\gamma = 2$, $\kappa = \frac{1}{2}$ ($m^* = 4.2$)

(d) $\gamma = 4$, $\kappa = \frac{1}{2}$ ($m^* = 1$)

Figure 4.3.3: Illustrative example showing how the parameters γ and κ affect the optimization problem solution. Four different graphs composed by $N = 25$ nodes have been generated by applying **Algorithm 2**.



(a) Total energy consumption



(b) Maximum energy consumption per node

Figure 4.3.4: The influence of m on the total energy consumption and the maximum energy per node is shown. The parameter m^* represents the average value of m used by our optimization methodology.

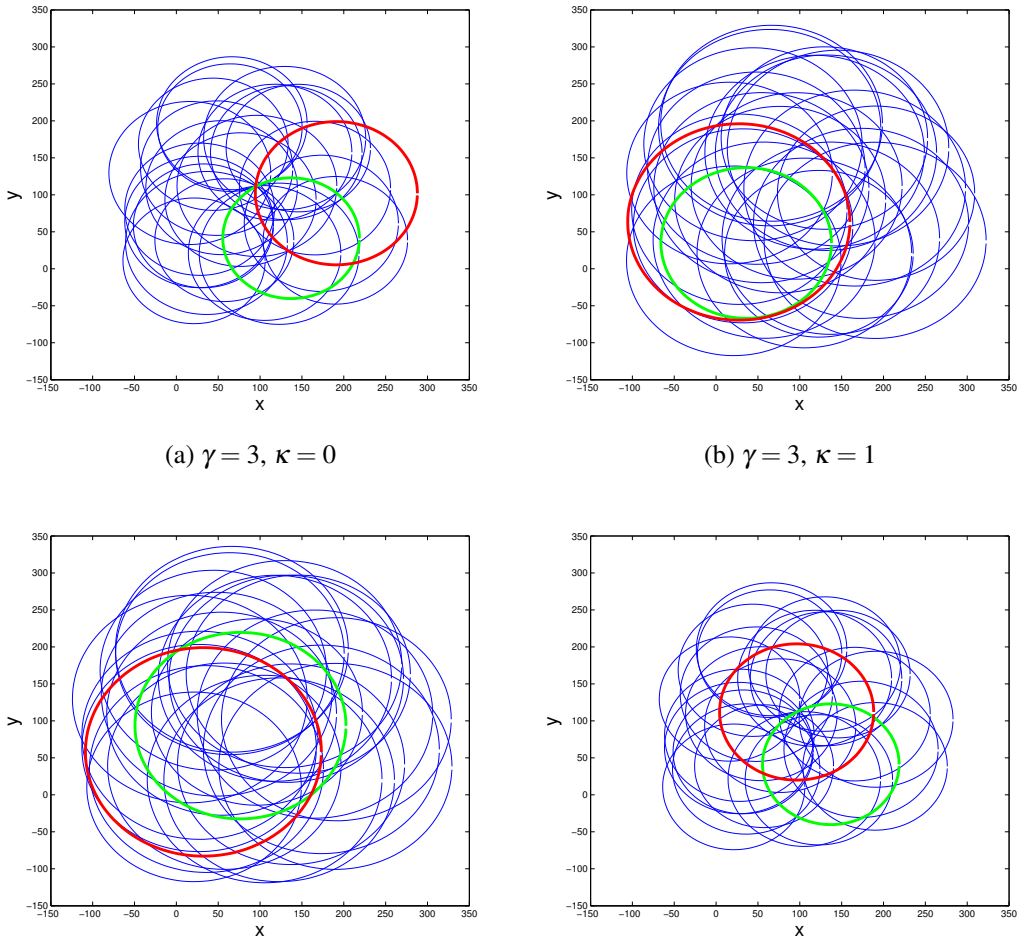
Broadcast wireless sensor networks

In our last setting, different magnitudes at multiple points of an industrial plant must be sensed to provide an automatic control of certain industrial equipment. Since the connection to the electricity grid is not possible, all sensor nodes must run on batteries. Moreover, all devices present the same energy and transmission capabilities, that is, $\mathrm{C}_1 = \mathrm{C}_2 = \cdots = \mathrm{C}_N$ and $R_1 = R_2 = \cdots = R_N = R$. Moreover, the communications between these nodes are broadcast, which implies that $\mathrm{p}_{\text{total}} = \sum_{i \in \mathcal{V}} \max_{j \in \mathcal{N}_i}(p_{ij})$.

Figure 4.3.5 shows an example of the optimization criterium that maximizes the lifetime of the network ($\kappa = 1$), that is, the time until the first node runs out of batteries. In this case, the best solution consists of making most of the nodes to transmit with almost the same transmission power, so that all of them run out of batteries almost simultaneously. Oppositely, if what is sought is to find the optimal graph in terms of total energy ($\kappa = 0$). The power is distributed among the nodes in a much more heterogeneous way than in the case of maximizing the network lifetime. The main reason is that any node may contribute to improve the total energy consumption by reducing its transmission range, as long as the algebraic connectivity is not significantly affected.

Finally, the resulting graphs, when broadcast communications are considered, present a much larger number of links in general. Once a link is established with a remote node, the nodes at shorter distance can also receive the data with no extra energy cost. Moreover, since these extra links always reduce the convergence time due to an increase of the algebraic connectivity, the total energy consumption and the network lifetime are both improved in case of adding them. The evolution of these two parameters is shown in Figure 4.3.6.

The experiments presented earlier in this chapter are useful to illustrate how our methodology, given a common deployment of nodes, leads to different graph formations depending on the constraints imposed by the nodes (R, p_{ij}, E_i), the environmental parameters (γ) and the chosen cost function (κ). However, small networks do not present a complex structure and important properties such scaling law, degree distribution, etc. are not revealed. For this



(a) $\gamma = 3$, $\kappa = 0$

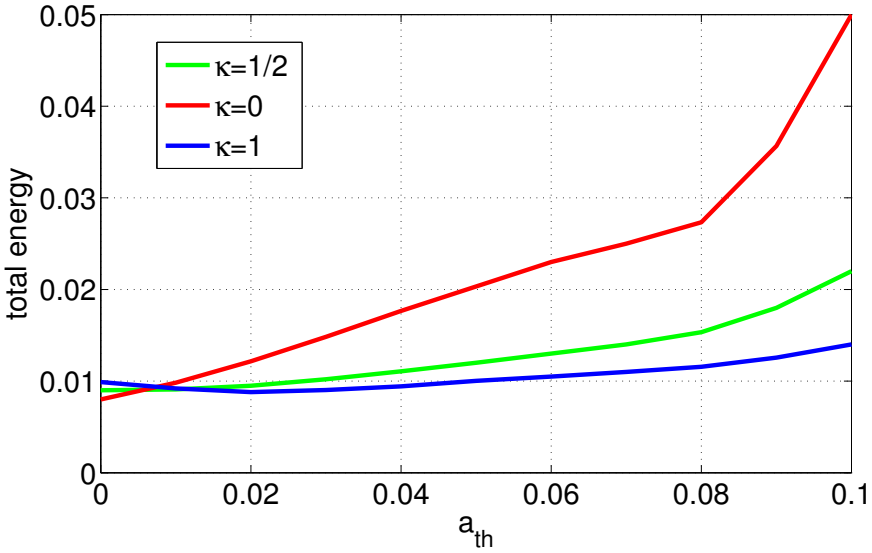
(b) $\gamma = 3$, $\kappa = 1$

(c) $\gamma = 2$, $\kappa = \frac{1}{2}$

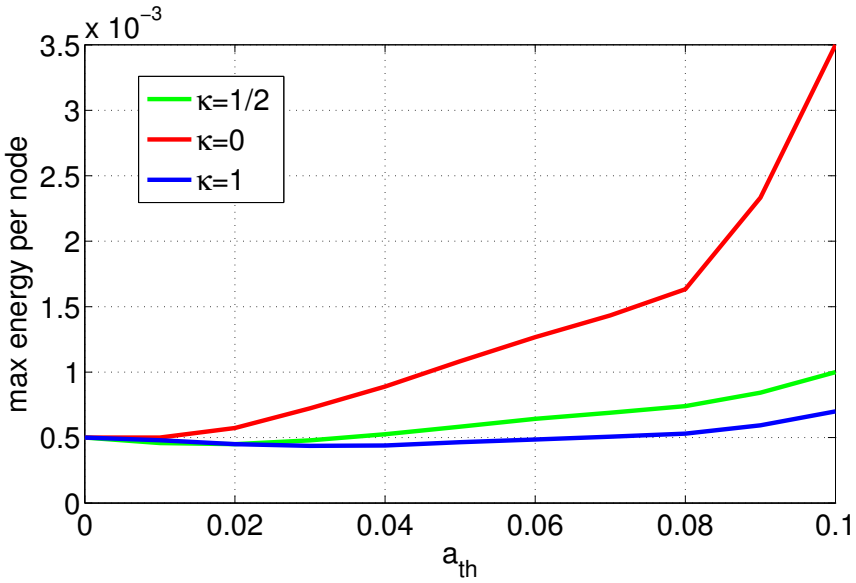
(d) $\gamma = 4$, $\kappa = \frac{1}{2}$

Figure 4.3.5: Illustrative example showing how the parameters γ and κ affect the optimization problem solution. Four different graphs composed by $N = 25$ nodes have been generated by applying **Algorithm 2** and choosing the optimal value of a_{th}. Red and green circles represents nodes having maximum and minimum transmission ranges.

purpose, from now on, we focus on complex networks of $N = 1000$ nodes, comparing the performance of existing models with our own methodology.



(a) Total energy consumption



(b) Maximum energy consumption per node

Figure 4.3.6: Influence of a_{th} on the total energy consumption and the maximum energy per node, which is evaluated for different values of the parameter κ and $\gamma = 3$.

Comparison with existing approaches

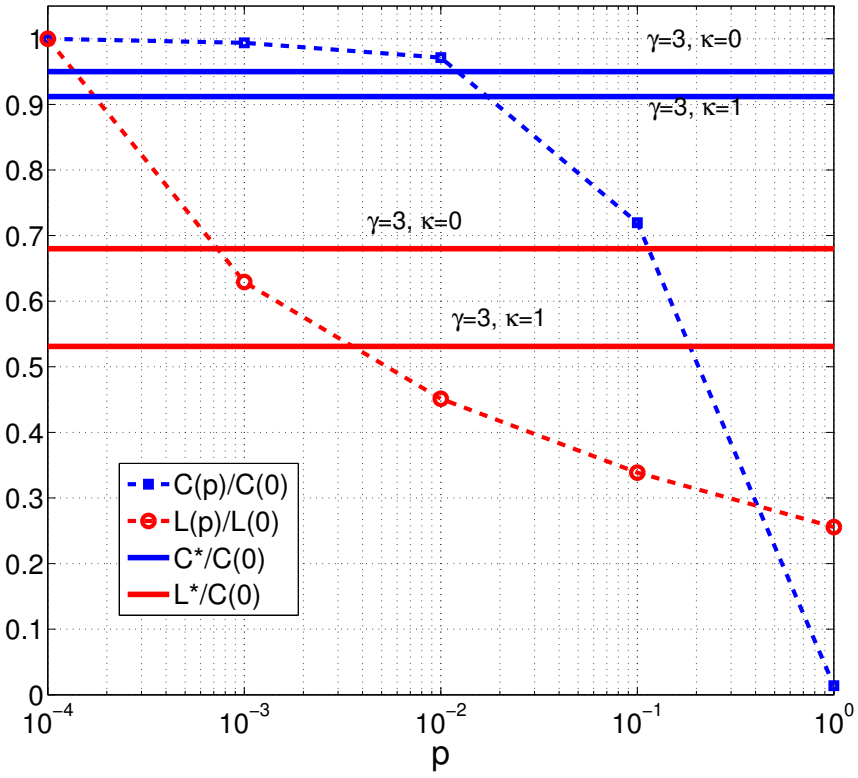
Formation of SWN

We first reproduce the methodology proposed by Watts and Strogatz [86] to create a small-world network by rewiring each existing link with a given probability p. The network from which the rewiring process is initiated, instead of a regular ring network, is a random deployment of $N = 1000$ nodes. These nodes establish a minimum number of links to ensure connectivity, that is, a multi-hop path exists between every pair of these nodes. Since a random network with a minimal connectivity has a initial diameter L greater than a regular ring, for values of $p = 1$ the result of the ratio $L(p)/L(0)$ is greater than the ratio resulted from the rewiring in the regular ring[3].

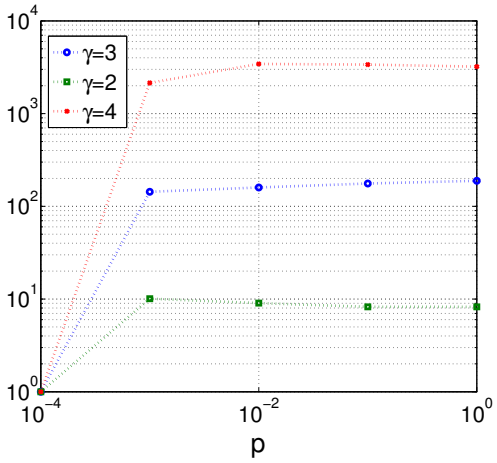
Figure 4.3.7 shows that a random rewiring methodology applied to the proposed network results in a clustering coefficient and a diameter similar to the one achieved in the work by Watts and Strogatz. It is easy to identify the region where the network begins to present small-world properties (values of the clustering coefficient similar to the ones of the initial topology combined with a considerable decrease of the diameter). However, this random rewiring does not take into account the spacial distribution of real networks. In real networks, it is common to have an associated cost to communications, so that, disregarding this can make the rewiring extremely inefficient from the point of view of the power consumption. Moreover, the convergence time may become prohibitive in applications where a dynamic process is executed. This time combined with the power consumption spent on communications lead to an extremely large energy consumption due to the rewiring process.

Figure 4.3.7 shows the total energy consumption and the maximum energy consumption per node associated to each rewiring probability. We have divided the energy results of the random rewiring by the values obtained with our methodology, where different values of γ are evaluated. For example, the energy consumption associated with the random rewiring process, when a value of $\gamma = 3$ is considered, can be 100 times larger than with our methodology.

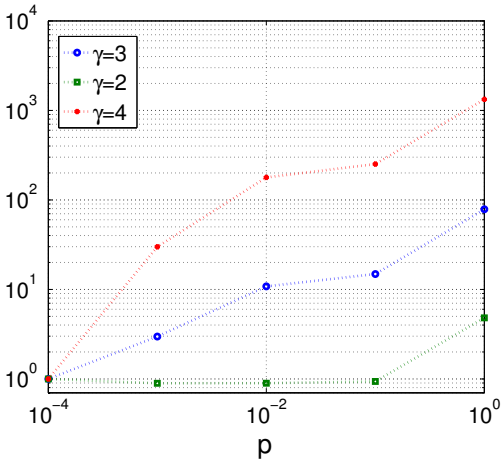
[3]The parameter $L(0)$ denotes the diameter of the initial graph, that is, when no rewiring have been performed. Oppositely the parameter L(p) represents such diameter when a rewiring with probability p has been performed.



(a) Clustering coefficient and diameter evolution



(b) $\frac{p_{i_{max}}(p)}{\lambda_2(\mathbf{L}(p))} \cdot \frac{\lambda_2(\mathbf{L}(\mathbf{A}))}{p_{i_{max}}(\mathbf{A})}$



(c) $\frac{p_{total}(p)}{\lambda_2(\mathbf{L}(p))} \cdot \frac{\lambda_2(\mathbf{L}(\mathbf{A}))}{p_{total}(\mathbf{A})}$

Figure 4.3.7: Comparison between the methodology proposed by Watts and Strogatz and the one presented throughhout this work. (a) Compares the resulting clustering coefficient and diameter, showing the transition region where the small-world effect is present. C* and L* represent the values resulted from applying our methodology of clustering coefficient and diameter respectively. (b) and (c) Shows that our methodology can improve the network topology as compared with the methodology proposed by Watts and Strogatz in terms of network lifetime and total energy consumption respectively.

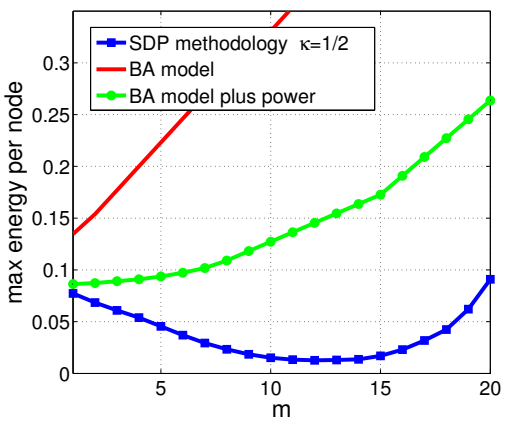
Formation of SFN

We additionally reproduce the methodology proposed in [2] to create a scale-free network using the preferential attachment. The network from which the process is initiated, is a fully connected network of 20 nodes which have been randomly deployed. The growing process is reproduced by adding 980 extra nodes, creating a large scale network of N=1000 nodes. This experiment is repeated 100 times and the results are averaged. The same tests are performed for the BA model plus power and our own methodology.

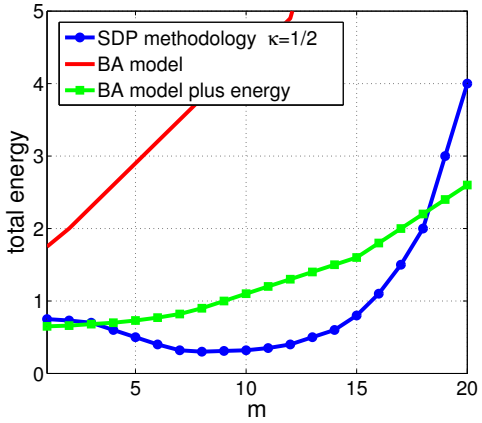
Figure 4.3.8 illustrates how the existing methods [2][91] for reproducing the formation of scale-free networks do not consider the energy that a dynamic process (e.g. consensus) consumes when is executed in such networks. The BA model, which does not take into account the distance between nodes nor the corresponding energy consumption, presents by far the worst energy results. This is because the preferential attachment is made with the nodes with highest degree regardless of the existing distance to them.

When spatial considerations are included, the total energy consumption is significantly improved. However, the individual energy consumption of each node is still not considered, which leads to some nodes consuming much more than others, and as a consequence the network lifetime is reduced.

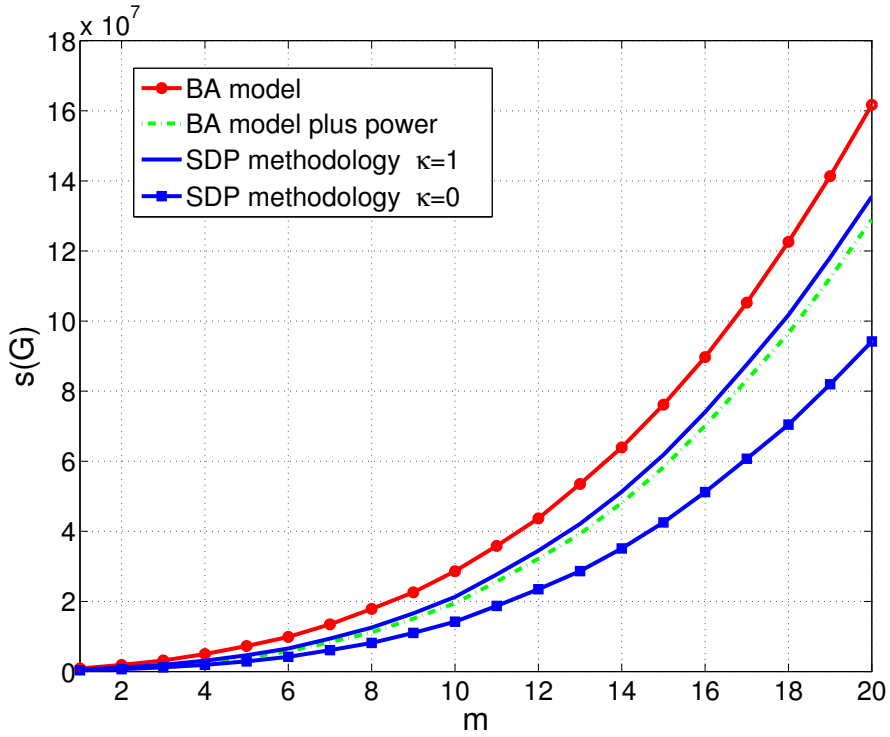
Our methodology is capable of solving such problems, choosing each time the connections to the nearest hubs that most improve the performance of the consensus process in terms of both total energy consumption and network lifetime. Besides this energy preservation, our methodology also allows to obtain very competitive values of the scale-free network indicator $s(\mathbf{G})$ as compared to the previous methodologies.



(a) maximum energy per node



(b) total energy consumption



(c) scale-free indicator

Figure 4.3.8: Comparison between existing methodologies for the creation of SFN and the one proposed in this work: a) maximum energy consumption per node, b) total energy consumption and c) scale-free indicator. Our methodology provides good results in terms of energy consumption and network lifetime while maintaining a high value of the SFG indicator.

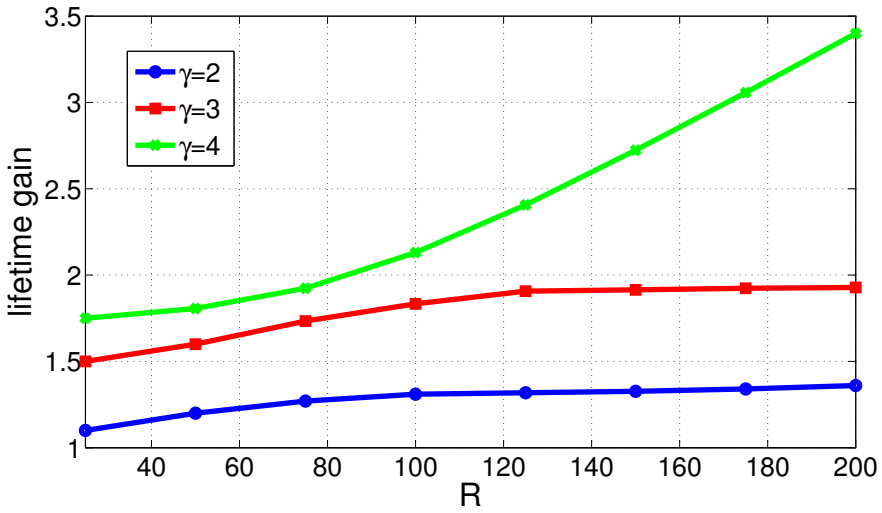
Formation of RGG

In the related literature, the formation of random geometric graphs follows a very simple process: Given a uniform random deployment of nodes, these connect with every other node at distance shorter than or equal to a given cut-off distance R.

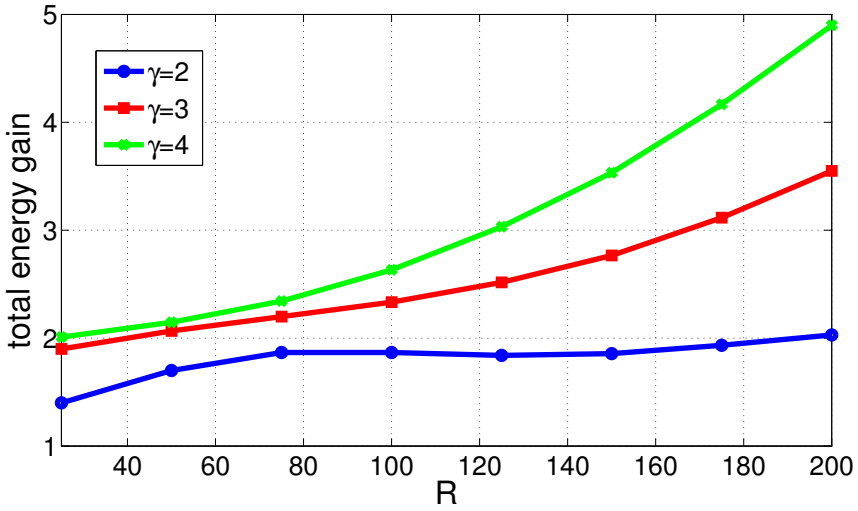
Finding the value of the parameter R that improves certain properties of the graph has been the subject of extensive study. However, most related work ignores the energy required by dynamic processes when these are executed over these graphs. The only work taking into account their total energy consumption is the work in [76]. However, even in this last related work, the solutions obtained are unsatisfactory because the transmission range must be the same for every node, which results in sub-optimal solutions. Furthermore, the network lifetime is not taken into account with nodes consuming much more energy than others.

To see the gain of our method with respect to methodologies that are based on a single cut-off distance, we simulate the formation of such networks from a minimum to a maximum value of the parameter R. This cut-off distance is used as a constraint in our problem, so the higher it is, the greater number of possible solutions are feasible.

Figure 4.3.9 shows the gain of our method with respect to the formation of an RGG graph using different cut-off distances and for different values of the path loss exponent γ. For high values of γ, the gain is increased, since establishing a link longer than necessary results in a much higher energy consumption. Similarly, from certain value of R, increasing more its value only worsens the energy consumption, since the links start to be too long (depending on γ). In our method, from certain value of R, the solution to the problem remains the same because the new possible solutions are worse and are discarded as a consequence. This means that, in general, relatively small values of power transmission are preferred, giving the minimum values of total energy consumption.



(a) lifetime gain



(b) total energy gain

Figure 4.3.9: Comparison in terms of total energy consumption and network lifetime between the existing methodology for the creation of a RGG and the one proposed in this work. Different values of γ are compared. Our methodology has been applied with $\kappa = 1/2$ in both graphs.

4.4 Applications of Topology Optimization

In this section, we show how to apply the topology optimization described in **P0** with different expressions of $f(\cdot)$, which are aimed at solving several real problems corresponding to different signal processing tasks. Then, rather than considering the consensus itself, we show how to use consensus as a tool to perform in a distributed manner two basic tasks in signal processing such as distributed signal detection and parameter estimation. In the first case, we focus on the energy efficiency of the process in order to extend the time that the network presents full detection capabilities, such that all the nodes are able to detect and perform their own actuation. In the second we focus on the distributed estimation problem where we need to ensure a minimum estimation quality, since the number of active nodes affect the estimation accuracy. Both problems, besides motivating the use of consensus algorithms, serve also to illustrate how the topology optimization can be utilized for different purposes by slightly varying the formulation of the problem. For simplicity, in this section, we assume homogeneous and broadcast WSNs.

4.4.1 Distributed and Energy Efficient Detection

Let us consider a network of N sensor nodes deployed over a certain area of interest with the objective of detecting a known[4] event $\mathbf{s}$. This problem can be formulated as [57]:

$$\begin{aligned} \mathscr{H}_0 &: \mathbf{y} = \mathbf{w} \\ \mathscr{H}_1 &: \mathbf{y} = \mathbf{s} + \mathbf{w} \end{aligned} \tag{4.4.1}$$

where $\mathbf{y}$ is the vector with the N node observations and $\mathbf{w}$ is the vector representing the white noise with zero mean and variance $\sigma_{\mathbf{w}}^2$. The problem defined by (4.4.1) can be solved by applying the well-known Neyman-Pearson detector $T(\mathbf{y})$, i.e., *matched filter*, where the filter impulse response is matched to the signal to be detected:

$$T(\mathbf{y}) = \frac{\mathbf{y}^T \mathbf{s}}{\sigma_{\mathbf{w}}^2} \geq \Psi \tag{4.4.2}$$

[4] If the signal is not known, a previous estimation step can be applied.

where Ψ is a predefined threshold that controls the detection probability [57].

It follows very easily that, in order to implement (4.4.2) in a distributed manner, it is enough to perform the consensus process in (3.1.1) with the following initial values:

$$x_i(0) = \frac{N y_i s_i}{\sigma_{\mathbf{w}}^2}, \qquad i = 1 \dots N$$

where $\frac{1}{\sigma_{\mathbf{w}}^2} \sum_{i=1}^{N} y_i s_i$ is the value that should be reached by all the nodes in the network.

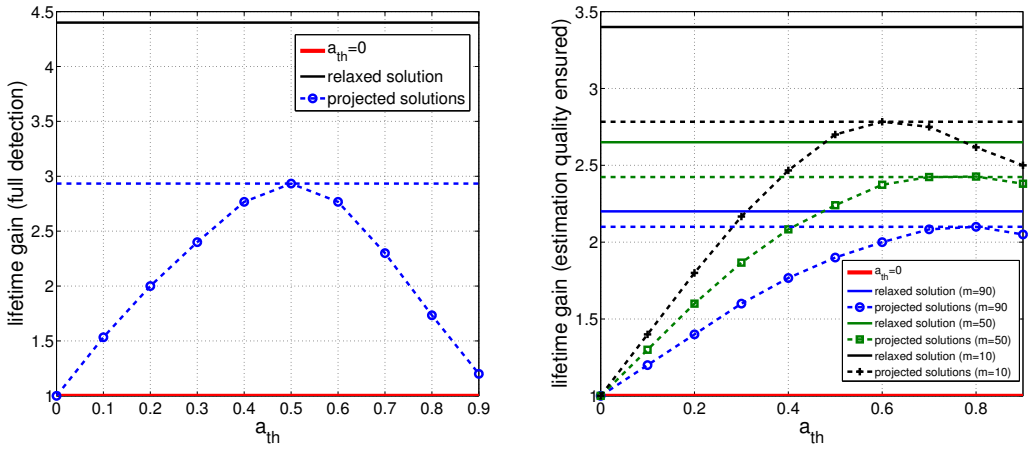
The result of the distributed detection, which is common to all nodes, can be exploited by each of them (e.g. to perform their own actuation). Thus, we are interested in maintaining the detection capability at every individual node. Then, since the heterogenous nodes performing the distributed detection process may have a different remaining energy budget C_i, it is crucial to consider the following parameter:

$$\mathscr{L}_{\min} = \min_{i \in \mathscr{V}} \left\{ \frac{\mathrm{C}_i \cdot \lambda_2(\mathbf{L})}{\sum_{j \in \mathscr{V}} p_{ij} \cdot [\mathbf{A}]_{ij}} \right\} \tag{4.4.3}$$

which defines the minimum number of consensus processes executed while full detection capabilities are maintained. In order to maximize this minimum lifetime $\mathscr{L}_{\min}$, we propose to introduce this parameter as the objective function of the problem **P1**. Thus, the relaxed version of **P1** particularized to the detection problem ($\kappa = 1$, $\mathscr{A} = \mathbf{1}\mathbf{1}^T$) can be formulated as follows (**P1a**):

$$\begin{aligned}
\min._{\{s,\mathbf{A}\}} \quad & s \\
\text{s. t.} \quad & \mathrm{p}_1(\mathbf{A}) - \mu \cdot \mathrm{C}_1 \cdot \lambda_2(\mathbf{L}(\mathbf{A})) \leq s \\
& \vdots \\
& \mathrm{p}_N(\mathbf{A}) - \mu \cdot \mathrm{C}_N \cdot \lambda_2(\mathbf{L}(\mathbf{A})) \leq s \\
& \xi \leq \lambda_2(\mathbf{L}(\mathbf{A})) \\
& [\mathbf{A}]_{ij} = [\mathbf{A}]_{ji} \ \ \forall i, j \in \mathscr{V} \\
& 0 \leq [\mathbf{A}]_{ij} \leq 1 \ \text{ if } \ r_{ij} \leq R_{\max} \ \ \forall i, j \in \mathscr{V} \\
& [\mathbf{A}]_{ij} = 0 \ \text{ if } \ r_{ij} > R_{\max} \ \ \forall i, j \in \mathscr{V} \\
& [\mathbf{A}]_{ij} \geq [\mathbf{A}]_{il} \ \text{ if } \ r_{ij} \leq r_{il} \ \ \forall i \in \mathscr{V}, \forall j, l \in \mathscr{N}_i
\end{aligned}$$

which is a convex parametric problem, whose solution is a function of the



(a) Lifetime gain (detection)

(b) Lifetime gain (estimation)

Figure 4.4.1: Lifetime gain to operate with (a) full detection capabilities and (b) ensuring a minimum quality of the estimation when our methodology is applied. This figure illustrates the gap (normalized by $a_{\text{th}} = 0$) between the relaxed solution $\mathbf{A}^*$ and the projected ones $\hat{\mathbf{A}}$. The optimal solution lies within this gap.

parameter μ.

Algorithm 2 finds the entries $[\mathbf{A}]_{ij}$ of the adjacency matrix, which defines the relaxed network topology. As stated before, for a given relaxed solution, the choice of a_{th} determines the final real topology. The solution $\hat{\mathbf{A}}$ that results from the projection procedure is feasible for the original problem, which it turns to be a lower bound in the lifetime gain to operate with full detection capabilities.The difference between the upper and the lower bounds is called the gap, see Figure 4.4.1. For example, for the particular setting used in Figure 4.4.1 (a), we can state that the optimal solution would lead to a gain from 2.9 to 4.4 times in the operational time of the network.

4.4.2 Distributed Parameter Estimation with Ensured Quality

Let us consider a random deployment of N nodes, each of which gets a noisy observation of an unknown parameter. Given this set of observations, the maximum likelihood estimator (MLE) of the parameter is simply the average of the N observations[5]. In particular, we assume the general linear observa-

[5]Provided that the noise is additive zero-mean Gaussian, with the same variance at each node, and spatially uncorrelated.

tion model:

$$\mathbf{x} = \theta + \mathbf{w}$$

where θ is the scalar parameter to be estimated, $\mathbf{x}$ is the vector with the N observations and $\mathbf{w}$ is the white gaussian noise of zero mean and variance $\sigma_{\mathbf{w}}^2$. It is well known that the MLE of θ given the observations [56] is:

$$\hat{\theta}_{ML} = \left(\mathbf{1}^T \sigma_{\mathbf{w}}^{-2} \mathbf{I}_N \mathbf{1}\right)^{-1} \mathbf{1}^T \mathbf{I}_N \sigma_{\mathbf{w}}^{-2} \mathbf{x} = \frac{1}{N} \mathbf{1}^T \mathbf{x}$$

where we recognize easily the average of the observations:

$$x_{\text{avg}} \triangleq \frac{1}{N} \sum_{i=1}^{N} x_i = \frac{1}{N} \mathbf{1}^T \mathbf{x}$$

Furthermore, for this simple setting, the variance of the estimator is:

$$\sigma_\theta^2 = \left(\mathbf{1}^T \sigma_{\mathbf{w}}^{-2} \mathbf{I}_N \mathbf{I}\right)^{-1} = \frac{1}{N} \sigma_{\mathbf{w}}^2 \tag{4.4.4}$$

Therefore, by computing the average of the initial observations $\mathbf{x}$, the error variance is reduced by a factor of N.

Performing average consensus to solve this problem in a distributed manner implies that after $\mathscr{L}_i$ processes have been performed, node i eventually runs out of battery, increasing the variance of the estimator in (4.4.4) as a consequence. In order to maintain a certain quality of the distributed estimator, such that its variance is kept below a specific threshold $\sigma_\theta^2 \leq \sigma_{th}^2 \leq \sigma_{\mathbf{w}}^2$, there is a maximum number of nodes that can run out of battery, which will be given by:

$$m = N - \frac{\sigma_{\mathbf{w}}}{\sigma_{th}} \tag{4.4.5}$$

which implies that we have $N - m$ survivor nodes still executing consensus.

The objective here is to obtain an optimal topology matrix $\mathbf{A}$, such that the lifetime of the m most energy demanding nodes that are executing the iterative consensus is maximized. If the variance of the observation noise remains constant over time, the minimum estimation quality based on subsequent consensus based estimations is ensured by the same value of m. Then, our goal is to perform a topology optimization that leads to maximizing the number of distributed estimations that can be performed with a certain guar-

anteed quality (error variance).

Let us arrange the elements of vector $\mathrm{p} = (\mathrm{p}_1, \mathrm{p}_2, \ldots, \mathrm{p}_N)$ in decreasing order and store them in the vector $\mathscr{O}(\mathrm{p})$, such that $\mathscr{O}_1(\mathrm{p}) \geq \mathscr{O}_2(\mathrm{p}) \geq \ldots \geq \mathscr{O}_N(\mathrm{p})$, then the sum corresponding to the power consumption of the m more demanding nodes can be expressed as follows[6]:

$$S_m(\mathrm{p}) = \sum_{i=1}^{m} \mathscr{O}_i(\mathrm{p}) \tag{4.4.6}$$

It can be shown [63] that (4.4.6) is equivalent to:

$$S_m(\mathrm{p}) = \frac{1}{N}\left(m \sum_{i=1}^{N} \mathrm{p}_i + g(\ell) \right) \tag{4.4.7}$$

where $(z)_+$ is defined as $(z)_+ = \max(z, 0)$ and $g(\ell)$ is:

$$g(\ell) = \min_{\ell \in \mathbb{R}} \sum_{i=1}^{N} [m(\ell - \mathrm{p}_i)_+ + (N-m)(\mathrm{p}_i - \ell)_+]$$

whose optimal solution is $\ell^* = \mathscr{O}_m(\mathrm{p})$, making $g(\ell^*) = N \cdot S_m(\mathrm{p}) - m\sum_{i=1}^{N} \mathrm{p}_i$.

By defining $\mathrm{p}_i^+ - \mathrm{p}_i^- = \mathrm{p}_i - \ell$, with $\mathrm{p}_i^+, \mathrm{p}_i^- \geq 0$, we can express (4.4.7) as follows:

$$\min_{\ell \in \mathbb{R}} \frac{1}{N}\left(\sum_{i=1}^{N} [m\mathrm{p}_i^- + (N-m)\mathrm{p}_i^+ + m\mathrm{p}_i] \right)$$

where we have exploited the fact that $(\ell - \mathrm{p}_i)_+$ and $(\mathrm{p}_i - \ell)_+$ are complementary in the sense that for a given value ℓ only one of the two terms is greater than zero.

Finally, if we make $\mathrm{p}_i^- = \mathrm{p}_i^+ - \mathrm{p}_i + \ell$, the expression in (4.4.6) is the solution of the following linear program:

$$\begin{array}{ll} \text{min.}_{\{\ell\}} & m\ell + \sum_{i=1}^{N} \mathrm{p}_i^+ \\ \text{s. t.} & \mathrm{p}_i^+ \geq \mathrm{p}_i - \ell,\ i = 1, \ldots, N \\ & \mathrm{p}_i^+ \geq 0,\ i = 1, \ldots, N \end{array}$$

Accordingly, the problem consisting on optimizing the network topology

[6] In order to include the energy budget C_i available at every node i, it is enough to redefine the vector p as $\mathrm{p}' = \left(\frac{\mathrm{p}_1}{\mathrm{C}_1}, \frac{\mathrm{p}_2}{\mathrm{C}_2}, \ldots, \frac{\mathrm{p}_N}{\mathrm{C}_N} \right)$.

to minimize the equivalent amount of energy can be cast as follows:

$$\begin{array}{ll} \min._{\{\ell,\mathbf{A}\}} & \frac{\left(m\ell+\sum_{i=1}^{N}\mathrm{p}_i^+\right)}{\lambda_2(\mathbf{L}(\mathbf{A}))} \\ \text{s. t.} & \mathrm{p}_i^+ \geq \mathrm{p}_i(\mathbf{A})-\ell,\ i=1,\ldots,N \\ & \mathrm{p}_i^+ \geq 0,\ i=1,\ldots,N \\ & \xi \leq \lambda_2(\mathbf{L}(\mathbf{A})) \\ & [\mathbf{A}]_{ij}=[\mathbf{A}]_{ji}\ \ \forall i,j\in\mathscr{V} \\ & 0\leq[\mathbf{A}]_{kl}\leq 1 \text{ if } r_{kl}\leq R_{\max}\ \ \forall k,l\in\mathscr{V} \\ & [\mathbf{A}]_{kl}=0 \text{ if } r_{kl}>R_{\max}\ \ \forall k,l\in\mathscr{V} \\ & [\mathbf{A}]_{kl}\geq[\mathbf{A}]_{kj} \text{ if } r_{kl}\leq r_{kj}\ \ \forall i\in\mathscr{V},\forall j,l\in\mathscr{N}_i \end{array}$$

Similarly to what is done to solve the problem **P1**, we introduce the following function:

$$h(\mu,\mathbf{A})=\min_{\ell,\mathbf{A}}\left\{\left(m\ell+\sum_{i=1}^{N}\mathrm{p}_i^+\right)-\mu\lambda_2(\mathbf{L}(\mathbf{A}))\right\}$$

where μ is a real positive parameter that controls the trade-off between $S_m(\mathrm{p})$ and the convergence time determined by $\lambda_2(\mathbf{L}(\mathbf{A}))$. Again, the values of $\mathbf{A}$ and ℓ that minimize $h(\mu,\mathbf{A})$ are the same ones that minimize our objective function. Furthermore, the value of μ that makes $h(\mu,\mathbf{A})=0$ is also the minimum of the cost function of the previous problem. Consequently, this previous optimization problem is equivalent to **P1b**:

$$\begin{array}{ll} \min._{\{\ell,\mathbf{A}\}} & \left(m\ell+\sum_{i=1}^{N}\mathrm{p}_i^+\right)-\mu\lambda_2(\mathbf{L}(\mathbf{A})) \\ \text{s. t.} & \mathrm{p}_i^+ \geq \mathrm{p}_i(\mathbf{A})-\ell,\ i=1,\ldots,N \\ & \mathrm{p}_i^+ \geq 0,\ i=1,\ldots,N \\ & \xi \leq \lambda_2(\mathbf{L}(\mathbf{A})) \\ & [\mathbf{A}]_{ij}=[\mathbf{A}]_{ji}\ \ \forall i,j\in\mathscr{V} \\ & 0\leq[\mathbf{A}]_{kl}\leq 1 \text{ if } r_{kl}\leq R_{\max}\ \ \forall k,l\in\mathscr{V} \\ & [\mathbf{A}]_{kl}=0 \text{ if } r_{kl}>R_{\max}\ \ \forall k,l\in\mathscr{V} \\ & [\mathbf{A}]_{kl}\geq[\mathbf{A}]_{kj} \text{ if } r_{kl}\leq r_{kj}\ \ \forall i\in\mathscr{V},\forall j,l\in\mathscr{N}_i \end{array}$$

which is a convex parametric problem, and whose solution is a function of the parameter μ. To solve that, we perform successive iterations as shown in **Algorithm 4**, which involves two different steps. In the first step, we obtain

a value of μ by computing our cost function for any valid matrix $\mathbf{A}$ and ℓ. In the second step, we solve the problem **P1b** for the value of μ obtained in the first step. We start the algorithm with any feasible solution and we iterate until a certain degree of accuracy is obtained, that is, the solution of problem **P1b** is close enough to the original one. The result of the algorithm are the entries $[\mathbf{A}]_{ij}$ of the adjacency matrix, which define the relaxed network topology. As stated before, for a given solution, the choice of a_{th} determines the final topology and this choice affects the energy consumption of nodes.

Algorithm 4

Require: ε, m
Ensure: $\left(m\ell + \sum_{i=1}^{N} \mathrm{p}_i^+\right) - \mu\lambda_2(\mathbf{L}(\mathbf{A})) \leq \varepsilon$

Set $\mathbf{A}, \ell$ as a feasible solution to **P1b**

while $\left(m\ell + \sum_{i=1}^{N} \mathrm{p}_i^+\right) - \mu\lambda_2(\mathbf{L}(\mathbf{A})) > \varepsilon$ **do**
set $\mu = \frac{\left(m\ell + \sum_{i=1}^{N} \mathrm{p}_i^+\right)}{\lambda_2(\mathbf{L}(\mathbf{A}))}$

Solve **P1b** with the current μ, obtaining $\mathbf{A}^*$ and ℓ^*
set $\mathbf{A} = \mathbf{A}^*$ and $\ell = \ell^*$
end while

4.4.3 Numerical Results

First, we evaluate the energy efficiency of a network of $N = 100$ nodes repeatedly executing a detection process based on consensus. Figure 4.4.2 shows the averaged results of applying **Algorithm 2** to 100 different random graphs. The x-axis represents the applied threshold a_{th} and the y-axis represents the corresponding magnitude of energy (expressed in μJules) needed to perform the detection process. The red solid lines (maximal connectivity) refer to the particular case $a_{th} = 0$, which is a common scenario in WSNs, in which every node uses the maximum transmission range $R_1 = R_2 = \ldots = R_N = R_{\max}$. Additionally, we have also compared with the optimal radius R_{opt} (common to all nodes), that is, $R_1 = R_2 = \ldots = R_N = R_{\text{opt}}$, which can be approximated by using the results in [76].

In particular, in Figure 4.4.2 (a), the total energy consumption for different values of the parameter κ is compared. As expected, the minimum value

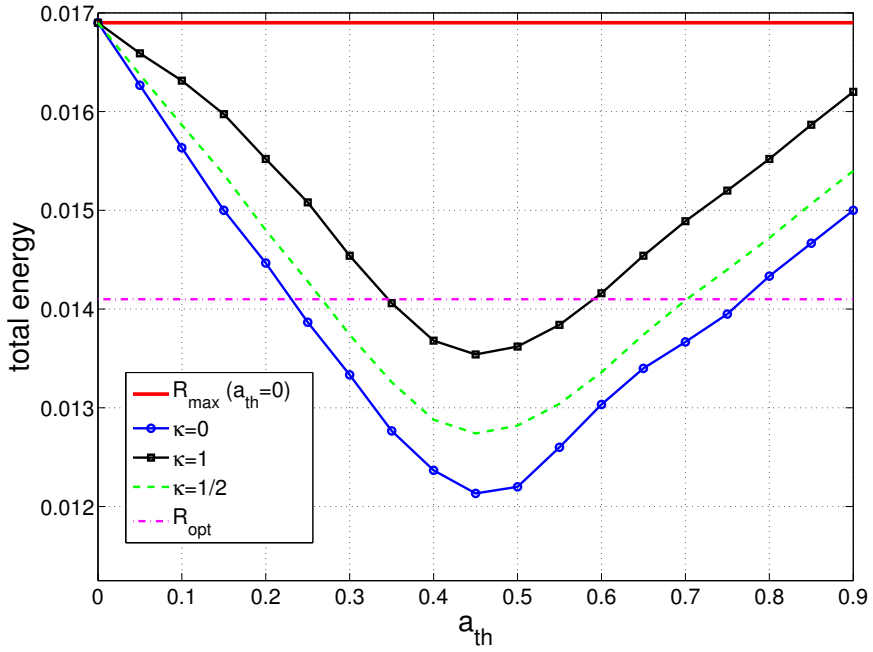
is obtained for $\kappa = 0$, while the worst performance is obtained for $\kappa = 1$. The rest of Pareto solutions lie between these two curves. It is interesting to point out that the value $\kappa = 1/2$, represented by the dashed curve, yields competitive maximum energy consumption values per node and quasi minimum total energy consumption. Similar results are obtained for the maximum energy consumption per node, as can be observed in Figure 4.4.2 (b). Both magnitudes can be reduced between the 33% and the 50% as compared to the maximal connectivity. This means that the distributed detection can be performed during longer periods of time before the batteries of one or several nodes should be changed.

Although the trade-off between the two previous magnitudes may characterize the energy efficiency of most tasks based on consensus, these provide no information about the power allocation among the different nodes of the network. In other words, we cannot conclude whether the energy can be reduced by saving power at one or several nodes, or by globally reducing the convergence time at the cost of extra power expenses. The power distribution of the nodes determines how homogeneous their lifetime is, which is crucial for maintaining a certain number of them alive. In particular, the variance of a distributed estimator based on consensus depends on the number of nodes executing this latter process. Thus, the power distribution of nodes is an additional parameter to evaluate.

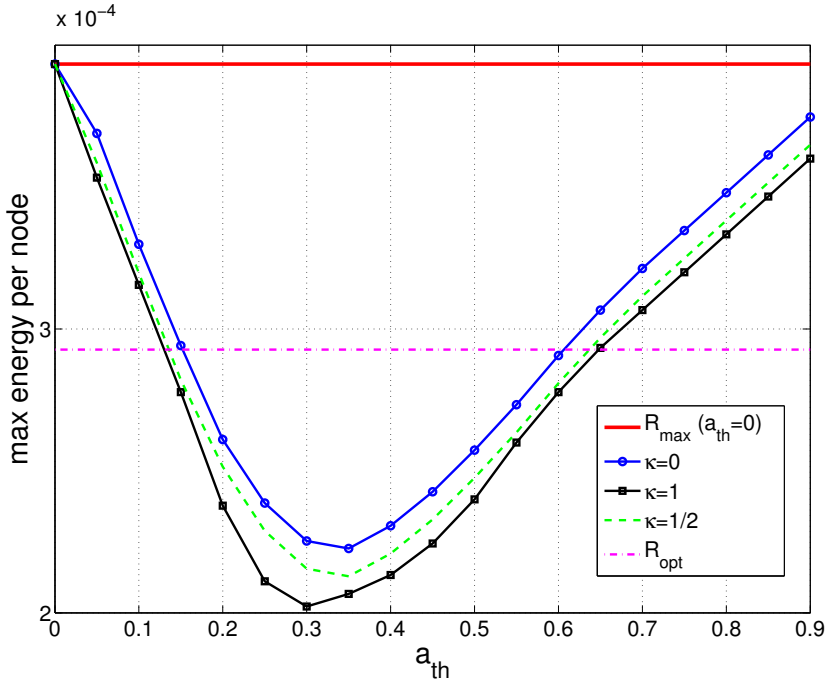
The effect of this power distribution can be better observed in Figure 4.3.5 for the specific case of $N = 30$, where the transmission ranges of all nodes are depicted. While the size of the circles is clearly determined by the parameter γ, it can be seen that the ratio between the maximum (red circle) and minimum (green circle) transmission ranges is affected by the choice of the number m of nodes involved in the optimization. More precisely, for $m = 1$ all transmission ranges have a similar length, so that every node has approximately the same power consumption per time step, while the convergence time is kept small. On the contrary, for $m = N$ the power consumption is very different for every node, and the total consumed power is significantly smaller than in the previous case.

Figure 4.4.3 (a) compares the function S_m for three different values of $m = 25, 50, 75$. This graph shows that similar curves can be obtained when different number of nodes are considered in the optimization criteria. The

ratio between the maximum and the minimum transmission range is crucial for the nodes lifetime. Small ratios imply similar power consumption for all nodes, so that most of them run out of battery after executing a similar number of estimation processes. The opposite occurs for large ratios, which implies that after one or several nodes run out of battery, the rest can still execute several processes. Additionally, for small ratios it takes longer to the first nodes to run out of battery than for larger ratios. This effect can be observed in Figure 4.4.3 (b), where we use the parameter m to rule the aforementioned ratio. This figure shows the number of remaining operative nodes in the network, as a function of the number of performed estimation processes and for different values of the parameter m. Note that in order to maximize the time until the first node run out of batteries, we must use $m = 1$ instead of $m = 30$ as intuition may suggest. This is because $m = 1$ ensures that the energy consumption of the most consuming node is minimum among all the possible values of m. The equivalent figure that shows the variance of the estimator as a function of the number of estimation processes executed is shown with the previous one.

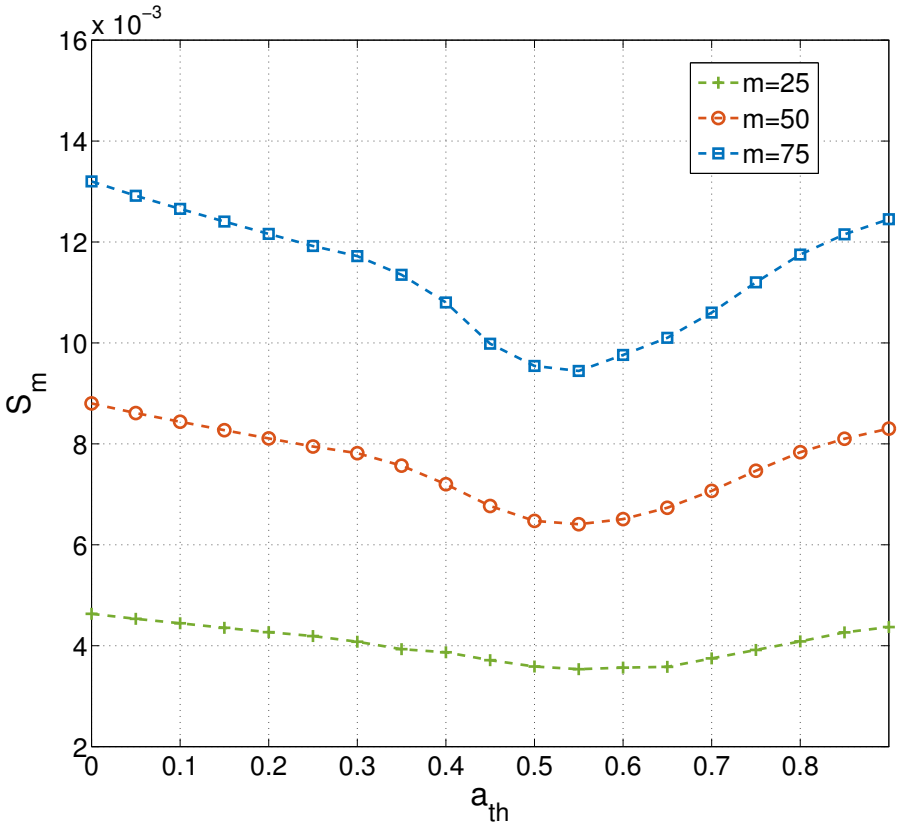


(a) Total energy consumption

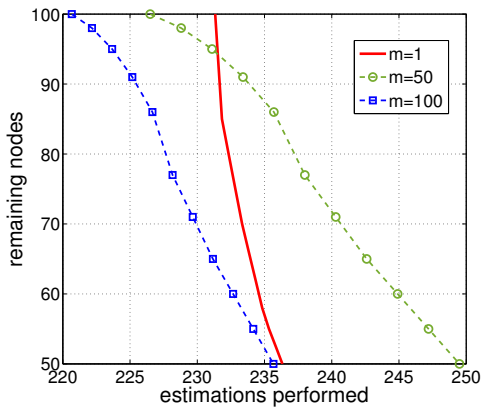


(b) Maximum energy consumption per node

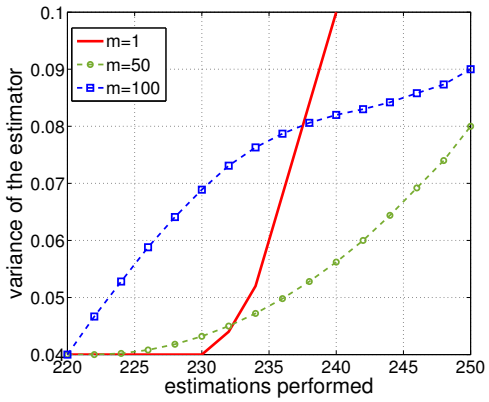
Figure 4.4.2: Total energy consumption and maximum energy consumption per node in a continuous system after the execution of a detection process based on consensus. The effect of different threshold values a_{th} and the trade-off parameter κ are compared. A common transmission range to all nodes with the values of $R_{\max}$ and R_{opt} are also shown in the graph. The value of $R_{\max}$ is obtained applying $a_{th} = 0$ to the solution topology (no optimization applied). The value of R_{opt} can be approximated using the result in [76].



(a) Summation of the m maximum consuming nodes



(b) Remaining nodes



(c) Variance of the estimator

Figure 4.4.3: The behavior of S_m for different values of the parameter m is shown for a continuous system. The parameter m also affects the number of nodes remaining in the network and hence to the variance of the estimator.

4.5 Conclusions of the Chapter

In this chapter, we have presented a new methodology to optimize the network topology in terms of several energy related functions. We first apply this methodology to three different settings that usually arise in industrial environments where extra situation awareness is needed. We also identify the constraints that impose each of these settings and relate them with the graph models that provide the best results in general. Thus, as opposite to previous works, which introduce unrealistic parameters to reproduce the formation of the different graph models, we identify several real parameters that affect the formation of these graphs. In particular, the formation of small word graphs (SWG), scale free graphs (SFG) and random geometric graphs (RGG) have been found to characterize the topology under certain real conditions. Moreover, motivated by two well known tasks in the field of signal processing, we also apply this methodology to improve their performance. In the particular case of the distributed detection problem, our optimization results on systems running full detection capabilities for longer periods of time. In the case of the distributed estimation problem, our optimization results on ensuring a minimum quality of the estimations for longer periods of time. Extensive numerical results have been presented to show the efficiency of our methodology in terms of the considered energy functions.

Chapter 5

Topology Optimization: Discrete Systems

5.1 Introduction

In the previous chapter, we have presented a new methodology for continuous systems to reach consensus in an energy efficient manner. This is based on a centralized method that performs an a priori design of the topology to minimize a set of cost functions that are based on convex combinations of the power vector defined in (2.4.1) multiplied by the convergence time in (3.1.2).

However, the sensor systems in practice are usually discrete. A discrete system, as opposite to a continuous system, involves digital components working with digital signals and having a countable number of states. For that reason, discrete systems have a direct relation to real sensor networks, since these networks are generally composed by digital devices (although integrated sensors may be analogic). In such systems, the moments in which the updates are performed are discrete time instants. In the particular case of a system executing consensus, these updates are performed whenever the nodes communicate with their neighbors and calculate their own new state with the data collected from them.

In addition, in the previous chapter, it is implicitly assumed the existence of a central entity in charge of obtaining the optimal topology. Once obtained, the solution is given to the deployed nodes, which communicate accordingly to reach consensus. However, since one of the main advantages of consen-

sus algorithms is that they operate without relying in such central entities, it is also of great interest to find new methods that can be implemented in a distributed manner, so that the network topology optimization can be collaboratively performed by the sensor nodes as the consensus algorithm itself. For this purpose, new low complexity methods to improve both the convergence time and the power consumption of consensus algorithms are necessary.

Another important issue to deal with is that the sensor network to be optimized may be already executing a consensus based application over an existing topology. This implies that a posteriori optimization methodologies, to be executed over the existing configuration, may be needed for such scenarios.

Although most of the results derived in Chapter 4 can be easily extended to discrete systems when a small step-size is considered, this is not always possible due to the time constraints that are present in the consensus-based applications. The main reason is that a small step-size can be far from efficient, since the convergence time is reduced by increasing the value of this parameter. Although the methodologies to be applied in a discrete system with a large step size are different from the ones for continuous systems, the energy parameters to be optimized are still determined by the convergence time and the power consumption of the nodes.

In this chapter, besides showing how to adapt the a priori topology design presented in Chapter 4, we also consider consensus algorithms executed over already formed topologies. The network topologies to be optimized are composed by computationally constrained discrete devices that run on batteries. This scenario makes the results of this chapter closer to reality than the previous one, which is generally theory-driven.

For simplicity and in order to not extend unnecessarily this chapter, we only consider homogeneous networks and unicast communications. Similar results can be obtained for the different network settings presented in the previous chapter. Thus, we firstly show how to adapt the topology optimization methodology proposed in Chapter 4 to discrete systems when a small enough step size is considered. We also identify the problems appeared when continuous and discrete systems are decoupled, finding, in this case, an alternative optimization method to improve the network topology. Finally, two distributed methodologies with different computational cost aimed to optimize the topology are proposed.

5.2 Topology Optimization: $\alpha \approx \Delta t$

As we have introduced in Chapter 4, if the value of the step-size $\alpha \approx \Delta t$ is small enough, the system evolution $\mathbf{x}(k+1) = (\mathbf{I} - \alpha\mathbf{L})\mathbf{x}(k)$ is equivalent to the continuous system described by the differential equation in (3.1.1). Given this equivalence, from expression (3.1.7), the energy consumption of node i for a broadcast scenario can be expressed as:

$$\begin{aligned}
\mathscr{E}_i(\mathbf{A},\mathbf{W}) &= t(\mathbf{A},\mathbf{W})\mathrm{p}_i(\mathbf{A}) \\
&= K\frac{\alpha(\mathbf{A})\mathrm{p}_i(\mathbf{A})}{1-\lambda_2(\mathbf{W})} \\
&= K\frac{\max\limits_{j\in\mathscr{V}}\{p_{ij}\cdot\alpha(\mathbf{A})[\mathbf{A}]_{ij}\}}{1-\lambda_2(\mathbf{W})} \\
&= K\frac{\max\limits_{j\in\mathscr{V}}\{p_{ij}\cdot[\mathbf{W}]_{ij}\}}{1-\lambda_2(\mathbf{W})} \\
&= K\frac{\mathrm{p}_i(\mathbf{W})}{1-\lambda_2(\mathbf{W})}
\end{aligned} \tag{5.2.1}$$

where we have utilized the relation between the off-diagonal entries of matrices $\mathbf{A}$ and $\mathbf{W}$. Thus, the energy $\mathscr{E}_i(\mathbf{A},\mathbf{W}) = \mathscr{E}_i(\mathbf{W})$ is totally characterized by the weight matrix $\mathbf{W}$.

For a unicast scenario, we can argue in the same terms with the particularity that $\mathrm{p}_i(\mathbf{W}) = \sum\limits_{j\in\mathscr{V}} p_{ij}\cdot[\mathbf{W}]_{ij}$.

Similarly, the lifetime of node i is given by:

$$\mathscr{L}_i(\mathbf{W}) = \left\lfloor \frac{\mathscr{C}_{\text{node}}}{\mathscr{E}_i(\mathbf{W})} \right\rfloor \tag{5.2.2}$$

where, again, $\mathscr{E}_i(\mathbf{W})$ can be substituted by the corresponding expression in both broadcast and unicast scenarios.

Thus, for a discrete system with small enough values of α, the problem of obtaining the optimal weight matrix that minimizes any energy related function can be cast as follows (**P0'**):

$$\begin{array}{ll} \min._{\{\mathbf{W}\}} & \frac{f(\mathbf{p}(\mathbf{W}))}{1-\lambda_2(\mathbf{W})} \\ \text{s. t.} & \mathbf{W1} = \mathbf{1} \\ & \mathbf{1}^T\mathbf{W} = \mathbf{1}^T \\ & \mathbf{W} = \mathbf{W}^T \\ & 0 \leq [\mathbf{W}]_{ij} \leq 1 \text{ if } condition1 \\ & [\mathbf{W}]_{ij} = 0 \text{ if } condition2 \end{array}$$

which is again a convex-concave fractional problem that can be solved by using the same methodology proposed for the continuous case. The result of problem **P0'** is the optimal weight matrix for each function considered in Table 1.

condition1: The entries of the weight matrix **W** should be between zero or one. Each entry is the weight that a node gives to the information coming from the corresponding neighbor. This condition is generally necessary to satisfy the doubly-stochasticity of this matrix.

condition2: Depending on the maximum power transmission of nodes p_{max}, some links cannot be established. Accordingly, some entries of the matrix **A** and consequently the corresponding entries of matrix **W** are set to zero, reducing considerably the number of variables present in the optimization problem.

Note that there exists a one-to-one correspondence between the solutions given by the relaxed version of the problem **P0** and the problem **P0'**. Once we have obtained the optimal matrix **A** from **P0** with all its entries between 0 and 1, we can always find a constant value of α to obtain the corresponding optimal matrix **W**.

5.3 Topology Optimization: $\alpha >> \Delta t$

The problem in **P0**' aims to minimize the same magnitudes than **P0**, but involving the weight matrix $\mathbf{W}$ instead of the topology matrix $\mathbf{A}$. This is only possible when the parameter α is maintained small enough. When this is no longer possible, we need to analyze the energy consumption of the discrete system in different terms.

Given the convergence time in (3.1.10) for a discrete system, we can redefine the energy consumption of any node i as follows:

$$\mathscr{E}_i(\mathbf{A},\mathbf{W}) = \frac{-\mathrm{p}_i(\mathbf{A})}{\log(\lambda_2(\mathbf{W}))}$$

which is the product between the power consumption and the convergence time. However, the denominator is no longer concave on $\mathbf{A}$, preventing us from formulating the energy-related problem as a convex-concave fractional one. Another important difference is that, given a broadcast transmission, the no-establishment of certain links under this transmission range may improve $\lambda_2(\mathbf{W})$ [12], reducing the convergence time as a consequence. Thus, the introduction of the step-size has two main implications:

- It prevents us from using the methodology utilized in the continuous case, since the cost function is no longer a fraction of convex and concave functions [11].
- A sparser network topology may lead to faster convergence times [12], since the dominant eigenvalue of the Laplacian may also be involved. This does not occur in the continuous case, where only the algebraic connectivity is needed to characterize the convergence time.

To solve the problem in a discrete system with a non-small step-size, we first introduce the following upper bound:

$$\mathscr{E}_i(\mathbf{A}) = \frac{\mathrm{p}_i(\mathbf{A})}{\sum_{q=1}^{+\infty} \frac{(\alpha\lambda_2(\mathbf{L}))^q}{q!}} \leq \frac{\mathrm{p}_i(\mathbf{A})\lambda_N(\mathbf{L})}{2\lambda_2(\mathbf{L})} \tag{5.3.1}$$

where we have used the Maclaurin serie of the function $\log(1-\alpha\lambda_2(\mathbf{L}))$ (with $|\alpha\lambda_2(\mathbf{L}))| < 1$), which is clearly lower bounded by $\frac{2\lambda_2(\mathbf{L})}{\lambda_N(\mathbf{L})}$.

5.3.1 Solution Based on Convex Optimization

Given the upper bound in (5.3.1) and due to the strong relation between the continuous and discrete problems, what we propose is to apply the problem **P1** introduced in Chapter 4 with two major modifications: i) the value of the largest eigenvalue $\lambda_N(\mathbf{L})$ of the Laplacian matrix is limited to a maximum value, which is determined by the maximum degree of the network[1] and ii) the entries of the matrix $\mathbf{A}$ are no longer sorted by distance. These two changes open the possibility of reaching sparse network topologies that maintain good values of $\frac{\mathrm{p}_i(\mathbf{A})}{\lambda_2(\mathbf{L}(\mathbf{A}))}$, while limiting the value of $\lambda_N(\mathbf{L})$. The complete methodology is summarized in **Algorithm 5**, which is based on solving the following problem **P1**' for different values of α:

$$
\begin{array}{ll}
\min._{\{\mathbf{A}\}} & f(\mathbf{p}(\mathbf{A})) - \mu\lambda_2(\mathbf{L}(\mathbf{A})) \\
\text{s. t.} & \xi \leq \lambda_2(\mathbf{L}(\mathbf{A})) \\
& [\mathbf{A}]_{ij} = [\mathbf{A}]_{ji} \ \ \forall i,j \in \mathscr{V} \\
& 0 \leq [\mathbf{A}]_{ij} \leq 1 \ \text{ if } \ r_{ij} \leq R_{\max} \ \ \forall i \in \mathscr{V} \\
& [\mathbf{A}]_{ij} = 0 \ \text{ if } \ r_{ij} > R_{\max} \ \ \forall i \in \mathscr{V} \\
& \sum_{j=1}^{N} [\mathbf{A}]_{ij} \leq \alpha^{-1} \ \ \forall i \in \mathscr{V}
\end{array}
$$

where $R_{\max}$ denotes the maximum transmission range of nodes.

In the problem **P1**', we have introduced the function $h(\mu,\mathbf{A}) = \mathbf{p}(\mathbf{A})) - \mu\lambda_2(\mathbf{L}(\mathbf{A})$ and the parameter $\alpha = \frac{1}{d_{\max}}$. The formulation of **P1**' can be easily adapted to solve the energy-related problems proposed for the continuous case, providing reasonable good results.

Although this centralized approach provides an important benchmark for comparison, unfortunately, computing all of this in a centralized fashion is not always satisfactory, since distributed and computationally constrained settings are commonly found in practice. It is natural to ask whether the design of the topology can also be performed in a decentralized fashion.

In the following sections, we consider two decentralized approaches with different computational requirements for improving both the convergence time and the power consumption based on redesigning the original network.

[1] It is well known that $\lambda_N(\mathbf{L}) \leq d_{\max}$. Accordingly we are applying a step-size of the form $\alpha = \frac{1}{d_{\max}}$.

Algorithm 5

Require: $d_{\max}$
  $d_{\text{th}} = d_{\max}$
  while $d_{\text{th}} \geq 1$ AND $\lambda_2(\mathbf{L}(\mathbf{A})) > \xi$ **do**
    $\alpha = 1/d_{\text{th}}$
    Set matrix $\mathbf{A}$ as a feasible solution
    while $f(\mathbf{p}(\mathbf{A})) - \mu\lambda_2(\mathbf{L}(\mathbf{A})) > \varepsilon$ **do**
      set μ as $\frac{-f(\mathbf{p}(\mathbf{A}))}{\lambda_2(\mathbf{L}(\mathbf{A}))}$
      Solve **P1'** with the current (μ,α), obtaining $\mathbf{A}^*$
      set $\mathbf{A} = \mathbf{A}^*$
    end while
    $d_{\text{th}} = d_{\text{th}} - 1$
  end while
  choose $\mathbf{A}(a_{\text{th}}, d_{\text{th}})$ that minimizes $\frac{-f(\mathbf{p}(\mathbf{A}))}{\log(\lambda_2(\mathbf{W}))}$

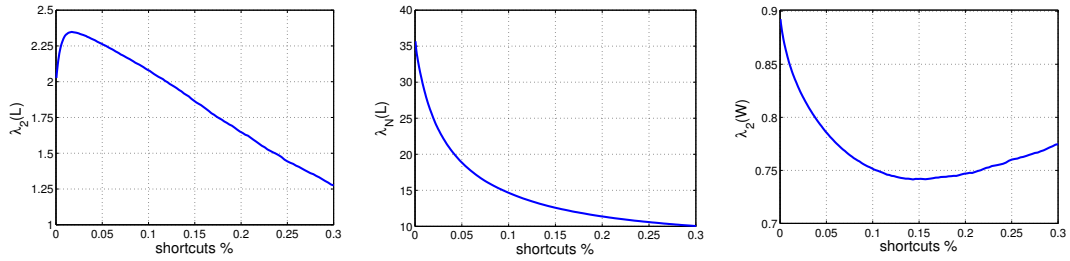


Figure 5.3.1: Average results over 1000 different network topologies composed by 100 nodes that are randomly deployed. These graphs have been generated by using the solution based on the spectrum of Laplacian matrix proposed in this chapter. The different parameters $\{\lambda_2(\mathbf{L}), \lambda_N(\mathbf{L}), \lambda_2(\mathbf{W})\}$ as a function of the percentage of added shortcuts are shown from left to right.

5.3.2 Solution Based on the Spectrum of Laplacian Matrix

In this section, we describe a method for improving the convergence of the average consensus algorithm based on the dominant eigenvector and the Fiedler eigenvector [46]. Our main idea consists of adding n links[2], one in each iteration of the algorithm, and removing in total m links, but in this case, a different number at each iteration. The number m of removed links depends on the energy associated to the additional links. Since, in this paper, we are interested in an energy efficient method, the number of removed links should be large enough to compensate the extra power consumption that results from adding the n links. Note that the added links require more power than the existing ones, which means that $m \geq n$. This implies that the resulting network is sparser than the original one.

If we use the weight structure defined in (3.1.5) and the optimal value of α defined in (3.1.9), the minimum convergence time is achieved when $\frac{\lambda_2(\mathbf{L})}{\lambda_N(\mathbf{L})}$ is maximum. This criterion is given by the existing relation between the eigenvalues of $\mathbf{L}$ and $\mathbf{W}$, that is, $\lambda_i(\mathbf{W}) = 1 - \alpha\lambda_i(\mathbf{L})$. The only way to reduce this quotient is by changing the underlaying topology. Moreover, it is also well known that the eigenvalues of the Laplacian matrix cannot decrease when a link is added and they cannot increase when a link is removed [48]. Then, this motivates the idea of finding the group of existing links such that removing them, $\lambda_N(\mathbf{L})$ is reduced as much as possible and the group of shortcuts that can be added to the original network in order to increase $\lambda_2(\mathbf{L})$ as much as possible. These two simultaneous effects lead to reduce $\lambda_2(\mathbf{W})$.

If $\mathbf{v_2}$ is an eigenvector with unit norm corresponding to $\lambda_2(\mathbf{L})$, then $\mathbf{v_2}\mathbf{v_2}^T$ is a supergradient of $\lambda_2(\mathbf{L})$. If $\lambda_2(\mathbf{L})$ is isolated, then $\lambda_2(\mathbf{L})$ is an analytic function of $\mathbf{L}$ and the supergradient is the gradient. Therefore, when $\lambda_2(\mathbf{L})$ is isolated, $([\mathbf{v_2}]_i - [\mathbf{v_2}]_j)^2$ gives the first order approximation of the increase in $\lambda_2(\mathbf{L})$, when the links $e_{ij}, e_{ji} \notin \mathscr{E}$ are added to the graph. A similar reasoning can be applied to the dominant eigenvalue $\lambda_N(\mathbf{L})$ and its corresponding eigenvector. Then, we add links based on maximizing the quotient $\frac{\left([\mathbf{v_2}]_i - [\mathbf{v_2}]_j\right)^2}{\left([\mathbf{v_N}]_i - [\mathbf{v_N}]_j\right)^2}$. However, since we are interested in a power efficient method, we also take into account the power required to create the new shortcut, that is, $p_{ij} = r_{ij}^{\gamma}$.

[2]We call these links shortcuts because of its similarity to the concept introduced in Chapter 4 for Small World Networks

Then, the final decision for adding a link, at each step of the process, is based on maximizing the following quotient:

$$\zeta_1 = \frac{([\mathbf{v_2}]_i - [\mathbf{v_2}]_j)^2}{([\mathbf{v_N}]_i - [\mathbf{v_N}]_j)^2 p_{ij}} \tag{5.3.2}$$

In a similar way, before executing a new iteration of the algorithm, we remove enough links to not increase the total power consumption. In this case, it is based on maximizing:

$$\zeta_2 = \frac{([\mathbf{v_N}]_i - [\mathbf{v_N}]_j)^2}{([\mathbf{v_2}]_i - [\mathbf{v_2}]_j)^2 p_{ij}} \tag{5.3.3}$$

Note that in this second case the power consumption associated to the link removed is included within the denominator. This means that to reduce this second quotient, the shorter links are removed first. Then the total number of links is more decreased, which results in a sparser network topology.

Our perturbation method requires only computing two eigenvectors, which is much less computational demanding than solving the optimization problem proposed in the previous section. The whole procedure is summarized in **Algorithm 6**.

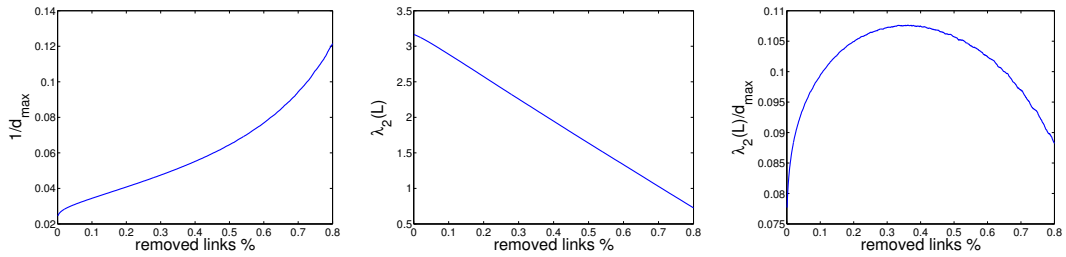


Figure 5.3.2: Average results over 1000 different topologies that are randomly deployed and each of them are composed by 100 nodes. These graphs have been generated by always removing the links of the nodes with maximum degree. The different parameters $\left\{\frac{1}{d_{max}}, \lambda_2(\mathbf{L}), \frac{\lambda_2(\mathbf{L})}{d_{max}}\right\}$ as a function of the percentage of removed links are shown from left to right.

5.3.3 Solution Based on Degree of Nodes

As explained before, removing links from high degree nodes can actually give rise to a neat positive effect in the convergence time. Figure 5.3.2 shows that

Algorithm 6 Spectrum-based algorithm

Require: $\mathrm{p}_{\mathrm{total}}$
Ensure: $d_i = d_i - 1, d_j = d_j - 1 \;||\; active\_flag = false$
  $\mathrm{p}_{\mathrm{before}} = \mathrm{p}_{\mathrm{total}}$
  while energy is reduced **do**
    Obtain ζ_1
    Add link e_{ij} based on ζ_1
    $\mathrm{p}_{\mathrm{now}} = \mathrm{p}_{\mathrm{before}} + 2p_{ij}$
    while $\mathrm{p}_{\mathrm{now}} > \mathrm{p}_{\mathrm{before}}$ **do**
      Obtain ζ_2
      Remove link e_{lm} based on ζ_2
      $\mathrm{p}_{\mathrm{now}} = \mathrm{p}_{\mathrm{now}} - 2p_{lm}$
    end while
    $\mathrm{p}_{\mathrm{before}} = \mathrm{p}_{\mathrm{now}}$
  end while

the parameter $\lambda_2(\mathbf{W})$ can be reduced by removing approximately 35% of the links in a random geometric network where connectivity is ensured with high probability [50]. In Figure 5.3.2, it is assumed that the entire set of nodes with maximum degree is known at each iteration step, which implies that some global information is being used. Our goal in this section is to obtain similar results, but only using local information and distributed computations.

Assuming a Random Geometric Graph, the degree distribution for a randomly deployed network can be approximated by a binomial distribution, or in the limit for large scale networks, by a Poisson distribution. Then, the probability of any node having degree k is $p(k) = \frac{e^{-d_{\mathrm{avg}}} d_{\mathrm{avg}}^k}{k!}$ and the degree distribution is simply given by $Np(k)$.

The main idea behind this distributed approach is to find a degree d_{th} such that all the nodes with degree greater than or equal to this parameter are chosen for losing links between them. This is done by combining the information given by Figure 5.3.2 and the degree distribution. In order to remove a percentage c of the total number of links M, if we assume that each node i having a degree $d_i \geq d_{\mathrm{th}}$ loses $(d_i - d_{\mathrm{th}})$ links, we can approximate the parameter d_{th} by using the following expression:

$$N \left[\sum_{i=d_{\mathrm{th}}}^{d_{\mathrm{max}}} p(i)(i - d_{\mathrm{th}}) \right] = cM$$

where d_{avg} can be approximated asymptotically by $\frac{M}{N}$. Then, if we divide the summation in two terms, we have:

$$\sum_{i=d_{\text{th}}}^{d_{\max}} p(i)i - d_{\text{th}} \left(\sum_{i=d_{\text{th}}}^{d_{\max}} p(i) \right) = cd_{\text{avg}}$$

The first summation is the expression for the expectation of a Poisson distribution where the first d_{th} terms have been removed. Therefore, this summation is always lower or equal than d_{avg} resulting in the following inequality:

$$d_{\text{th}} \left(\sum_{i=d_{\text{th}}}^{d_{\max}} p(i) \right) \leq (1-c)d_{\text{avg}} \tag{5.3.4}$$

We are looking for the minimum value of c that ensures $\text{prob}(d_i < cd_{\text{avg}}) \leq 0.05$ and allow us to simplify the left hand side of (5.3.4) by using $\sum_{i=d_{\text{th}}}^{d_{\max}} p(i) \approx 1$. Since the Poisson distribution has almost all the probability accumulated around the parameter d_{avg}, the range of values that allow us to make this approximation in a dense deployed network is obtained for values of $c \geq 0.35$. In other words, in this range of values, both the number of nodes having low degree and the associated probability are small. Moreover, since the resulting expression for the parameter d_{th} is proportional to the average degree of the network, it can be calculated in a distributed way before executing one or several realizations of the consensus algorithm because its inherent simplicity. Then, each node can use **Algorithm 7**, at each iteration step, to achieve a proper solution. Figure 5.3.2 shows that using this parameter d_{th}, we can achieve the best convergence when $d_{\text{th}} \approx 0.65 d_{\text{avg}}$, that is, by removing approximately 35% of the total number of links in the network.

In conclusion, this distributed approach consists in removing links from nodes having high degree. These nodes are chosen by using the parameter d_{th}, which can be obtained in a distributed way.

5.4 Numerical Results

In this section, we compare the results of the methods proposed in this chapter for the optimization of the network topology. We test our algorithms using 1000 different randomly deployed networks of $N = 100$ nodes, where con-

Algorithm 7 Degree-based approach

Require: $d_1, ..., d_M$
Ensure: $d_i = d_i - 1, d_j = d_j - 1 \;||\; active\_flag = false$
  if $d_i > d_{th}$ **then**
    $j = index(\max(d_1, d_2, ..., d_M))$
    if $d_j > d_{th}$ **then**
      $[\mathbf{A}]_{ij} = 0$, $[\mathbf{A}]_{ji} = 0$
      $d_i = d_i - 1, d_j = d_j - 1$
    else
      $active\_flag = false$
  else
    $active\_flag = false$

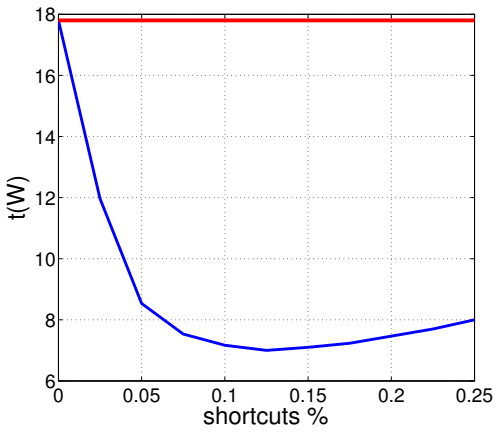
nectivity is ensured with high probability.

Figure 5.4.1 shows that our spectrum-based method reduces $\lambda_2(\mathbf{W})$ by maximizing $\frac{\lambda_2(\mathbf{L})}{\lambda_N(\mathbf{L})}$. The resulting networks are sparser than the original ones due to the reduction in the connectivity of the network. Figure 5.4.1 also shows that there exists an optimal topology for a given power consumption that minimizes $\lambda_2(\mathbf{W})$, beyond which, it is not possible to improve this parameter.

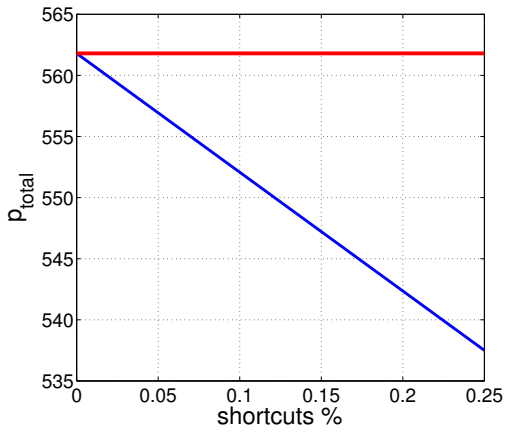
Figure 5.4.1 shows the behavior of two of the main parameters presented in this paper, the convergence time, the power consumption and the total energy consumption, as a function of the percentage of shortcuts included in the network, that is, the number of links added by our method divided by the total number of links. These graphs show clearly the benefits of using our approach in terms of the parameters considered in the study.

Figure 5.4.2 shows the chosen value of c giving, on average, the fastest convergence time when applying our distributed method and saving, at the same time, a large amount of power by reducing the connectivity of the network.

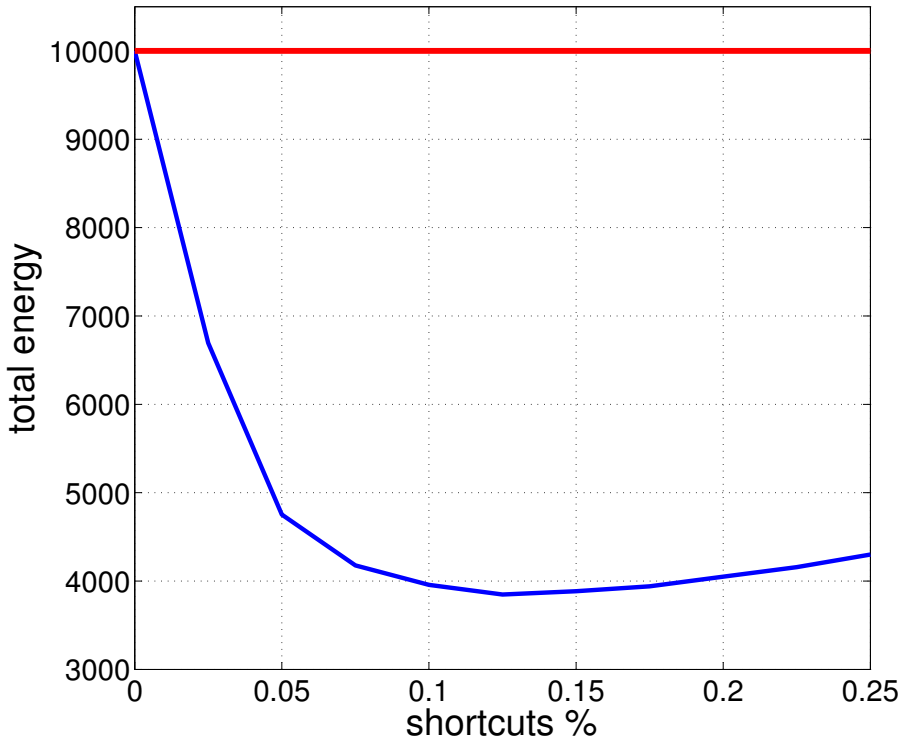
Finally, Figure 5.4.3 shows a comparison in terms of the maximum deviation from the average. Here, we compare the evolution of the topology that results from applying the methods proposed in this chapter. It is easy to see that our distributed methods obtain good results when compared with the centralized benchmark in terms of convergence time, while keeping a low power consumption.



(a) convergence time

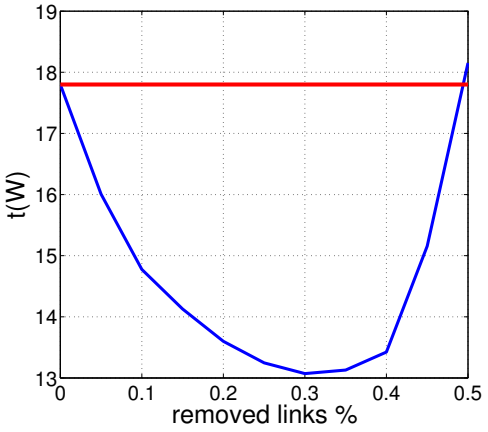


(b) power consumption

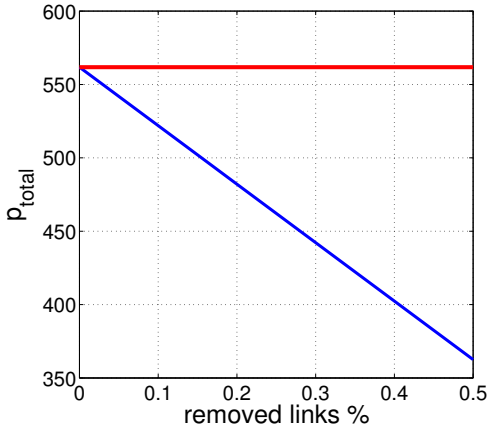


(c) energy consumption

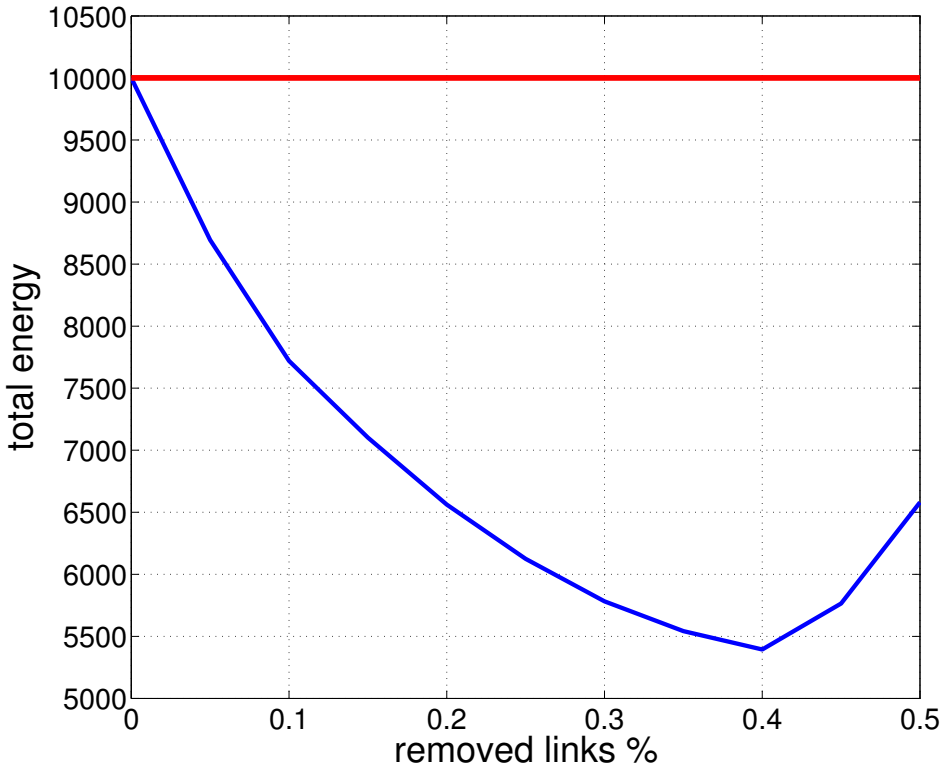
Figure 5.4.1: Convergence time, total power consumption and total energy consumption of the topologies resulted from applying the spectrum-based method as a function of the percentage of shortcuts. The red solid line represents the value of the parameters using the original graph.



(a) convergence time



(b) power consumption



(c) energy consumption

Figure 5.4.2: Convergence time, total power consumption and total energy consumption of the topologies resulted from applying the degree-based method as a function of the percentage of removed links. The different parameters values are shown as a function of c and generated by removing links from nodes with degree greater or equal than $\gamma = (1-c)\lambda$. The red line is the value of the parameters when no links are removed.

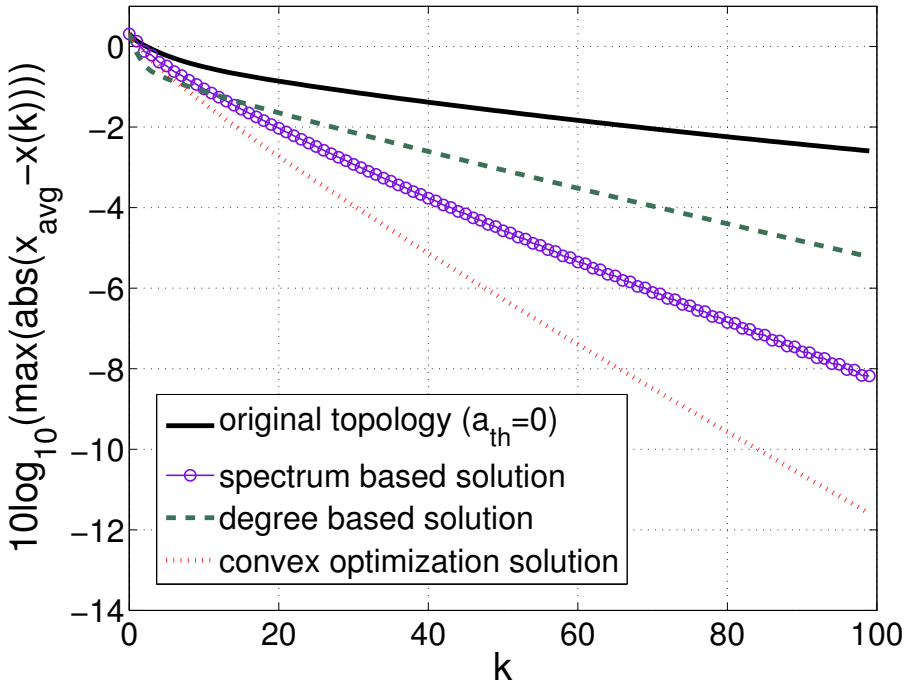


Figure 5.4.3: Comparison of the maximum deviation as a function of the iteration number between the different methods proposed in this chapter for discrete systems when the step-size is not small enough.

5.5 Conclusions of the Chapter

In this chapter, we have proposed several methodologies to optimize the network topology, where the step size determines which to use. When the step size is small enough or independent of the topology, the optimization problem proposed in the previous chapter may be used here by applying a simple reformulation involving the weight matrix. However, when the step-size is not small enough or it is dependent of the topology, we have presented a totally different methodology to optimize the power consumption of the nodes and the convergence time needed to achieve consensus. In particular, we have proposed a centralized and two distributed approaches to reduce the parameters considered, which present different computational cost and can be utilized in different network configurations.

Part III

Consensus in Time-variying Graphs

Chapter 6

Random Directed Graphs: Asymmetric Gossip

6.1 Introduction

In the presence of random communications, since every instantaneous topology is randomly generated, the optimization methodologies proposed in the two previous chapters are no longer valid. In practice, random communications imply that the power vector $\mathrm{p}(k)$ in each iteration k and the convergence time $t(\mathbf{W})$ are both random[1]. If the number of iterations needed to reach convergence is random, then the number of instantaneous topologies generated during the consensus process is also random. Thus, in order to obtain the energy consumption of node i, we should calculate $\mathscr{E}_i = \sum_{k=1}^{t(\mathbf{W})} \mathrm{p}_i(k)$. Any method seen in previous chapters to optimize the topology requires a priori knowledge of how the communications are performed. Then, in this random setting, the value of $\mathscr{E}_i$ cannot be properly optimized, since we do not know neither the number nor the structure of the instantaneous topologies $\mathbf{A}(k)$ that are generated. Finally, the conditions used in the previous chapters to ensure convergence are not either valid here. This means that new conditions are required for time-varying graphs and new consensus schemes must be derived consequently.

[1]Note that for continuous systems despite essential progress in consensus over time-varying graphs, some issues in this area remains open. This partly pertains to a gap between necessary and sufficient conditions for consensus in the part concerned with random interaction topology. Since for discrete systems this gap was recently filled, when in this work it is referred to consensus in (random) time-varying topologies, this is always consensus in discrete systems.

In the presence of random communications, the simplest scheme that ensures average consensus almost surely is the gossip algorithm presented in Boyd et al. [31], which focuses in a low complexity implementation. In this scheme, the non-zero entries of $\mathbf{W}(k)$ are generated by randomly activating, at each time instant k, an undirected edge, which determines the nodes that pairwise exchange and mix information at that iteration. The information is mixed according to a fix real number $\alpha \in (0,1)$, which is again the step size of the process. This scheme leads to the nodes i and j to produce a new state according to the equations:

$$\begin{aligned} x_i(k+1) &= (1-\alpha)x_i(k) + \alpha x_j(k) \\ x_j(k+1) &= \alpha x_i(k) + (1-\alpha)x_j(k) \end{aligned} \tag{6.1.1}$$

The state of the rest of nodes remains unaltered. Note that this scheme is implicitly using instantaneous weight matrices of the form $\mathbf{W}(k) = \mathbf{I} - \alpha\mathbf{L}(k)$.

In order to provide almost sure convergence to the average value, this protocol is forcing to ensure the following:

- A strongly connected undirected graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ in which all the instantaneous random subgraphs $\mathscr{G}(k)$ are included $\mathscr{G}(k) \subset \mathscr{G}$.
- Every instantaneous weight matrix is doubly-stochastic, that is, $\mathbf{W}(k)\mathbf{1} = \mathbf{1}$ and $\mathbf{1}^T\mathbf{W}(k) = \mathbf{1}^T$.

The first condition is ensured by having a sufficiently large power transmission at the nodes, while the second one is satisfied by ensuring that every matrix $\mathbf{W}(k)$ is a $N \times N$ symmetric matrix of the form $\mathbf{W}(k) = \mathbf{I} - \alpha\mathbf{L}(k)$. Accordingly, the symmetric gossip algorithm defined by equation (6.1.1) provides, by only using local communications, almost sure convergence to the average:

$$\mathbf{x}(k) \to \mathbf{x}_{\text{avg}}, \quad \text{almost surely} \tag{6.1.2}$$

where $\mathbf{x}_{\text{avg}}$ is a vector that contains N entries equal to the average μ of the initial data of nodes, that is, $\mathbf{x}_{\text{avg}} = \mu\mathbf{1}$.

As explained in Chapter 1, the linear iteration given by (6.1.1) generally takes many iterations to converge. Several alternative schemes of communications have been proposed in order to accelerate consensus. Some important examples are the works of [41], [26] and [22]. In particular, the work in [41] makes use of geographic information to accelerate the gossip process, which requires to establish long routes, making it difficult to ensure instantaneous symmetric communications. This work was improved in [26], by allowing path averaging. Finally, the work in [22] proposes the use of asymmetric broadcast communications, which provides fast convergence to some particular value, but it significantly differs from the average.

In practice, ensuring symmetric communications is difficult due to the existence of wireless interferences and other environmental factors. The majority of the existing related works assume symmetric exchanges of data to reach average consensus almost surely, with the notable exceptions of [25] and [53], which focus on asymmetric unicast and asymmetric broadcast communications respectively.

The main problem of using asymmetric communications is that the summation of the initial data is not preserved along the iterations. When an asymmetric packet is sent, the receiving node updates its state producing a deviation from the target state (average consensus in our case). We call the difference between two consecutive states the residual. The intuition behind our approach is to exploit this residual generated in each asymmetric data exchange, to preserve the summation of the system and accelerate the consensus process.

Our methodology has two important advantages over previous methods [25] and [53]. Firstly, the matrices used during the process are row-stochastic instead of column-stochastic, facilitating the analysis of the protocol. In addition, the use of the residuals allows to use local information to the nodes to guide the communications so that the convergence is faster. This means that although the activations of the nodes that perform a transmission are random, the receivers are chosen using local knowledge.

6.2 Residual gossip

We propose an asymmetric gossip algorithm that ensures average consensus almost sure under unicast and broadcast communications. In this asymmetric scenario, the instantaneous graphs $\mathscr{G}(k) \subset \mathscr{G}$ are generally directed. The weight matrix $\mathbf{W}(k)$ at each iteration k is asymmetric and only the condition $\mathbf{W}(k)\mathbf{1} = \mathbf{1}$ is fulfilled. Finally, the data is still mixed according to a fix real number $\alpha \in (0,1)$. Let us consider first the following gossip scheme:

$$x_j(k+1) = \alpha x_i(k) + (1-\alpha)x_j(k),\ \forall j \in \mathscr{N}_i(k) \tag{6.2.1}$$

where the remaining nodes in the network, including the transmitter node i, do not update their state value, that is:

$$x_\ell(k+1) = x_\ell(k),\ \forall \ell \notin \mathscr{N}_i(k) \tag{6.2.2}$$

This is the simplest asymmetric gossip scheme, which is studied in [45]. It ensures convergence to a common state value almost surely, but this state is not necessarily the average (see Figure 6.2.1). In other words, there exists a random variable a such that $\forall i \in \mathscr{V}$ it is accomplished that $\lim_{k\to\infty} x_i(k) = a$, almost surely.

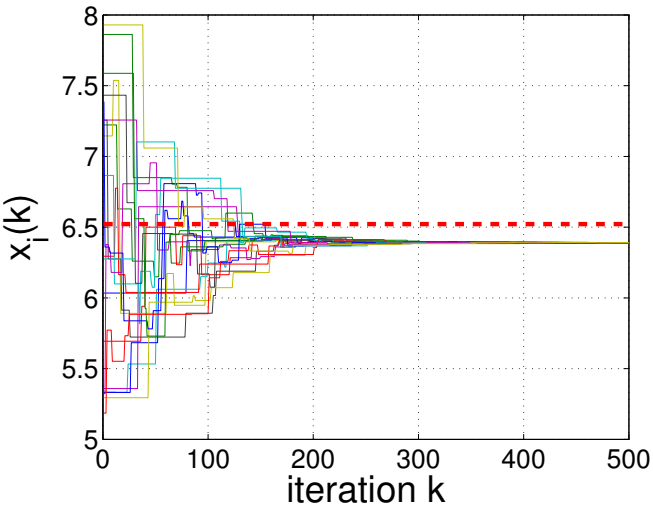


Figure 6.2.1: Example of the convergence of $N = 20$ nodes with an asymmetric gossip algorithm. The process converges to a certain value that clearly differs from the initial average. This initial target value is represented by the red dashed line in the figure.

At each step of the asymmetric process, a residual component is ignored. Disregarding it is what causes the final deviation from the average. In particular, the following information is ignored:

$$\Delta_j = x_j(k) - (\alpha x_i(k) + (1-\alpha)x_j(k)),\ \forall j \in \mathcal{N}_i(k) \tag{6.2.3}$$

which coincides with the current variation of node j.

What we propose in our residual gossip algorithm is to include in the packet transmitted through each link e_{ij}, together with the current state of the transmitter node i, its residual $r_i(k)$ accumulated until iteration k. Accordingly, the system evolves following the new equation:

$$x_j(k+1) = \alpha x_i(k) + (1-\alpha)(x_j(k) + r_i(k)),\ \forall j \in \mathcal{N}_i(k) \tag{6.2.4}$$

Note that in order to evaluate the state of each node i, we require to calculate $x_i(k) + r_i(k)$. Then, consensus is reached whenever $\lim_{k\to\infty}(x_i(k) + r_i(k)) = a$, almost surely. Later in this chapter, we restrict the residuals to be $r_i(k) = 0$, $\forall i \in \mathcal{V}$, so that the convergence condition is the usual one, that is, $\lim_{k\to\infty} x_i(k) = a$, almost surely.

Lemma 6.1: Given the system evolution in (6.2.4) and in order to ensure the summation of the process is preserved along the iterations, the residual of any receiver j must evolve as follows:

$$r_j(k+1) = r_j(k) + x_j(k) - x_j(k+1) + r_i(k),\ \forall j \in \mathcal{N}_i(k) \tag{6.2.5}$$

while the residual of any transmitter i is set to zero $r_i(k+1) = 0$ and rest of residuals remain unaltered.

Proof. In order to preserve the summation of the process in this new scheme, it must be accomplished that the summation of the state of the nodes and its residual at both iterations k and $k+1$ are the same, that is, $x_i(k) + x_j(k) + r_i(k) + r_j(k) = x_i(k+1) + x_j(k+1) + r_i(k+1) + r_j(k+1)$, $\forall j \in \mathcal{N}_i(k)$. By substituting $r_i(k+1) = 0$ and $r_j(k+1) = r_j(k) + x_j(k) - x_j(k+1) + r_i(k)$ in the right hand side of the previous equality, it can be easily seen that both terms are exactly the same. □

This alternative asymmetric scheme defined by (6.2.4) and (6.2.5) pre-

serves the summation of the initial data along iterations (states plus residuals). Then, if this scheme is still able to converge to a common state, it is surely that this is the average of the initial data.

Equation in (6.2.4) can be expressed in matrix form as follows:

$$\mathbf{x}(k+1) = \mathbf{W}(k)(\mathbf{x}(k) + \mathbf{s}(k)) \tag{6.2.6}$$

where the the residual vector $\mathbf{s}$ is given by:

$$[\mathbf{s}(k)]_j = \begin{cases} r_i(k) & \text{if } j \in \mathcal{N}_i(k) \\ 0 & \text{otherwise} \end{cases}$$

and the weight matrix $\mathbf{W}(k)$ is given by:

$$[\mathbf{W}(k)]_{\ell j} = \begin{cases} 1-\alpha & \text{if } j = \ell,\ \ell \in \mathcal{N}_i(k) \\ \alpha & \text{if } j \neq \ell,\ j = i,\ \ell \in \mathcal{N}_i(k) \\ 1 & \text{if } j = \ell,\ \ell \notin \mathcal{N}_i(k) \\ 0 & \text{otherwise} \end{cases}$$

where node i is chosen to send data in the current iteration k.

The weight matrix and the residual vector allow us to express (6.2.4) as follows:

$$\begin{aligned}
\mathbf{x}(1) &= \mathbf{W}(0)(\mathbf{x}(0) + \mathbf{s}(0)) \\
&= \mathbf{W}(0)\mathbf{x}(0) + \mathbf{W}(0)\mathbf{s}(0) \\
\mathbf{x}(2) &= \mathbf{W}(1)(\mathbf{x}(1) + \mathbf{s}(1)) \\
&= \mathbf{W}(1)\mathbf{W}(0)(\mathbf{x}(0) + \mathbf{s}(0)) + \mathbf{W}(1)\mathbf{s}(1) \\
&\vdots \\
\mathbf{x}(k+1) &= \mathbf{W}(k)(\mathbf{x}(k) + \mathbf{s}(k)) \\
&= \mathbf{W}(k)\mathbf{W}(k-1)\ldots\mathbf{W}(1)\mathbf{W}(0)\mathbf{x}(0) + \\
&\quad + \mathbf{W}(k)\mathbf{W}(k-1)\ldots\mathbf{W}(1)\mathbf{W}(0)\mathbf{s}(0) + \\
&\quad + \mathbf{W}(k)\mathbf{W}(k-1)\ldots\mathbf{W}(1)\mathbf{s}(1) + \\
&\quad \vdots \\
&\quad + \mathbf{W}(k)\mathbf{s}(k)
\end{aligned}$$

where $\mathbf{s}(0)$ is a vector with all of its entries equal to zero. This system evolution can be expressed as follows:

$$\mathbf{x}(k+1) = \prod_{i=0}^{k} (\mathbf{W}(i)\mathbf{x}(0)) + \sum_{j=0}^{k} \left(\prod_{\ell=j}^{k} (\mathbf{W}(\ell)\mathbf{s}(j)) \right) \tag{6.2.7}$$

Proposition 6.1: The system defined by (6.2.7) reaches average consensus almost surely $\lim_{k\to\infty} \mathbf{x}(k) = \mathbf{x}_{\text{avg}}$, provided that $\lim_{k\to\infty} r_i(k) = 0 \; \forall i \in \mathscr{V}$.

Proof. The first term of (6.2.7), given by $\prod_{i=0}^{k} (\mathbf{W}(i)\mathbf{x}(0))$, defines an asymmetric consensus as the one studied in [45]. Since every $\mathbf{W}(k)$ is row stochastic, the product of these matrices and the initial vector $\mathbf{x}(0)$ leads to consensus to a scalar random variable a almost surely. The second term can be seen as the summation of several asymmetric consensus. Let us denote $k = C$ the first iteration in which $r_i(k) \leq \xi \; \forall i \in \mathscr{V}$, for an arbitrarily small value of ξ. When this consensus iteration $k = C$ is reached, we can continue the process Δ iterations more, that is, $\sum_{j=0}^{C+\Delta} \left(\prod_{\ell=j}^{C+\Delta} (\mathbf{W}(\ell)\mathbf{s}(j)) \right)$. If Δ is large enough, all the products, started before the iteration $k = C$, are able to achieve consensus to a scalar random variable a_i almost surely. Formally, our system provides:

$$\lim_{k\to\infty} \mathbf{x}(k) = a\mathbf{1} + \sum_{i=0}^{C} a_i\mathbf{1} + \sum_{i=C+1}^{C+\Delta} a_i\mathbf{1}, \quad \text{almost surely}$$

where $a_i \leq \xi$ if $i > C$.

Therefore, since we are ensuring that the summation (states plus residuals) is preserved along the iterations, our scheme provides average consensus $\lim_{k\to\infty} \mathbf{x}(k) = a\mathbf{1} + \sum_{i=0}^{C} a_i\mathbf{1} = \mathbf{x}_{\text{avg}}$ almost surely.

□

As stated at the beginning of this section, the process defined by (6.2.7) is suitable for both unicast and broadcast communications. The only difference between these two schemes is the number of nodes that receive the information and the way in which the residuals are disseminated.

Our overall algorithm is described as follows: When a new iteration is started, a node i is randomly chosen. This node wakes up and sends a unicast or a broadcasts packet containing its own state value and the residual divided

by certain quantity $1 \leq \beta \leq d_i$. The packet is received by the destination node or the whole neighborhood. Once it is received, the corresponding nodes update their information according to (6.2.4). Therefore at each iteration k, we have the following:

- Node i sends a packet containing $x_i(k)$ and $\frac{r_i(k)}{\beta}$ and sets $r_i(k) = 0$. How to choose β is explained later.
- The packet is successfully received by the destination node in the unicast scheme, or by every node $j \in \mathcal{N}_i(k)$ in the broadcast scheme.
- The nodes receiving the packet from node i update its own state according to (6.2.4), which generates an update of its own residual by using (6.2.5). The state of the rest of the nodes remains unaltered.

This algorithm reaches average consensus almost surely, if $\lim_{k\to\infty} r_i(k) = 0 \; \forall i \in \mathcal{V}$, as shown in **Proposition 6.1**.

6.2.1 Condition for the residuals to vanish

A general condition for the residuals to vanish is that after certain number of exchanges[2] κ, any extra transmission from any node i to any node j at any time instant $k > \kappa$ satisfies the following:

$$\sum_{n=1}^{N} |r_n(k)| \geq \sum_{n=1}^{N} |r_n(k+1)| + \varepsilon \tag{6.2.8}$$

where ε is an arbitrarily small real number, which determines how fast the residuals vanish. The condition (6.2.8) clearly implies that $\lim_{k\to\infty} \sum_{n=1}^{N} |r_n(k)| = 0$, that is, all the residuals tend to zero.

For simplicity let us assume that in each iteration only one node i communicates with only one node j. In this particular case, condition (6.2.8) becomes:

$$|r_i(k)| + |r_j(k)| > |r_j(k+1)| \tag{6.2.9}$$

[2]Note that residuals can be initialized to zero, so they first need to grow in absolute value. Once these have a value different from zero, these can be used to guide the process to average consensus. In following sections we explain efficient methodologies to initialize the residuals with suitable values.

where we have noted that it is always accomplished that $|r_i(k+1)| = 0$.

Proposition 6.2: A transmission from node i to node j at time instant k satisfies condition (6.2.8) if $|x_j(k) - x_i(k)| < \frac{1-\alpha-\varepsilon}{\alpha}|r_i(k)|$.

Proof. Lets first develop the expression of $r_j(k+1)$:

$$\begin{aligned} r_j(k+1) &= r_i(k) + r_j(k) + x_j(k) - x_j(k+1) \\ &= r_i(k) + r_j(k) + x_j(k) - (\alpha x_i(k) + (1-\alpha)(x_j(k) + r_i(k))) \\ &= r_j(k) + \alpha(r_i(k) + x_j(k) - x_i(k)) \end{aligned}$$

then, it is accomplished that:

$$\begin{aligned} |r_j(k+1)| &= |r_j(k) + \alpha(r_i(k) + x_j(k) - x_i(k))| \\ &\leq |r_j(k)| + |\alpha(r_i(k) + x_j(k) - x_i(k))| \end{aligned}$$

which allows to express the condition in (6.2.8) as follows:

$$|\alpha(r_i(k) + x_j(k) - x_i(k))| + \varepsilon < |r_i(k)|$$

which is accomplished if:

$$\alpha|r_i(k)| + \alpha|(x_j(k) - x_i(k))| + \varepsilon < |r_i(k)|$$

and noting that $0 \leq \alpha \leq 1$, we can express the previous inequality as:

$$|x_j(k) - x_i(k)| < \frac{1-\alpha-\varepsilon}{\alpha}|r_i(k)|$$

which concludes the proof. □

Thus, if the residual of a node i sending the information is equal to zero $r_i(k)$, the exchange cannot take place without violating Proposition 6.2. However, convergence to the average is still possible, whenever the variation produced by these transmissions is equal to zero.

Corollary: The difference $|r_j(k+1)| - |r_j(k)|$ between two consecutive residuals in a receiving node j, produced by the transmission of a transmitting node i with residual $r_i(k) = 0$, tends to vanish, that is, $\lim_{k\to\infty}(|r_j(k+1)| -$

$|r_j(k)|) = 0$.

Proof. Given a node i with residual $r_i(k) = 0$, its transmission to node j does not affect the resulting residual $r_j(k+1)$, if consensus has been reached. To prove this, note that $r_j(k+1) = r_j(k) + (x_j(k) - x_j(k+1)) + r_i(k)$, since we are considering here any transmitter node i with $r_i(k) = 0$ and noting that $\lim_{k\to\infty}(x_j(k) - x_j(k+1)) = 0$, we have the following:

$$\lim_{k\to\infty}(|r_j(k+1)| - |r_j(k)|) = 0$$

□

Thus, it is important to differentiate the following two cases:

- The residual of the node that is sending the information is zero. This means that there is no extra information to guide the process towards consensus. In this case, we assume that the nodes are in a state equivalent to the original state and can exchange information normally, generating a residual in the receiver. Note that the summation of the process is preserved and the difference between two consecutive residuals tends to vanish.
- The residual of the node that is sending the information is not zero. In this case, the transmission always occurs and the receiving node decides what to do with the information in order to satisfy Proposition 6.1. These transmissions tend to reduce the value of the residuals, making them to vanish.

In the following sections, we show different heuristics to make our residual gossip efficient forth both unicast and broadcast schemes of communication.

6.2.2 Unicast scheme of communications

When a unicast scheme of communications is considered, only one node receives the information from the origin node i. We want to minimize the value of the residual along the iterations, which intuitively leads to ensure

$\lim_{k\to\infty} r_i(k) = 0 \; \forall i \in \mathscr{V}$. There are two ways to operate in this scenario: i) the origin node i randomly chooses a neighbor j from $\mathscr{N}_i$, which updates its state if **Proposition 6.2** is fulfilled, or node j directly updates its residual with r_i otherwise or ii) the origin node decides to which node is best to send the information according to some local criteria. We focus on the second methodology, which leads us to the following heuristic:

Min-max approach (h1): This heuristic chooses the neighbor with minimum state value if the current residual of the origin node i is positive and it chooses the neighbor with maximum state value if the current residual of the origin node i is negative. This local information can be estimated during the process, updating the minimum and the maximum when the neighbors send their own information.

This approach ensures that the residual $r_i(k)$ and the difference between states $x_j(k) - x_i(k)$ are of opposite sign. As a consequence, we can upper bound $|\alpha(r_i(k) + x_j(k) - x_i(k))|$ by $\max(\alpha|r_i(k)|, \alpha|x_j(k) - x_i(k)|)$, leading to the condition:

$$|x_j(k) - x_i(k)| < \frac{1}{\alpha}|r_i(k)| \tag{6.2.10}$$

which is less conservative than the one stated in **Proposition 6.2** for a general case.

Notice that both processes, namely, the nodes reaching consensus and the residuals approaching to zero, are closely related. In fact, if the residual sent from the transmitter to the receiver makes their difference among states to remain equal or to become smaller, the difference among states and the resulting residual are both decreased. This reduction is explained by the manner in which the state of the nodes and the residuals are updated.

6.2.3 Broadcast scheme of communications

When a broadcast scheme of communications is considered, every node $j \in \mathscr{N}_i(k)$ receives the information from the origin node i. The proposed heuristics in this scheme are:

Equitable approach (h2): This first method is based on equally distributing the residuals. For this purpose, the origin node i sends $(x_i(k), \frac{r_i(k)}{d_i})$ to every node $j \in \mathscr{N}_i(k)$, that computes its new state (6.2.4) and residual (6.2.5)

if the condition stated in **Proposition 6.2** is fulfilled.

Large-small approach (h3): This heuristic updates the neighbors with smaller state value than node i if its current residual is positive or it updates the state of the neighbors with larger state value otherwise. The value of β is obtained accordingly to the resulting $\mathcal{N}_i(k)$. The rest of neighboring nodes only update their residual. Similarly to the unicast case, the condition in **Proposition 6.2** can be also relaxed here.

6.2.4 Toy example

Let us define a network example composed by $N = 5$ nodes, where, after k iterations, the state of the nodes is $\mathbf{x}(k) = [6\ 4\ 2\ 8\ 5]^T$ and the residuals $\mathbf{r}(k) = [-3\ -1\ 2\ 2\ 0]^T$. Now imagine that node $i = 1$ has been chosen to broadcast its value to its neighbors 2, 3 and 4, which only 2 and 4 satisfy **Proposition 6.2**. In this case, if we use $\alpha = 1/2$, then the entries of the matrix $\mathbf{W}$ are $W_{21} = W_{31} = W_{41} = 1/2$, $W_{11} = W_{55} = 1$, $W_{22} = W_{33} = W_{44} = 1/2$ and the rest are zero. Finally, if we consider the equitable approach (h3), the residual is spread as $r_i(k)/d_i = -3/3$, thus vector $\mathbf{s}$ is $\mathbf{s} = [0\ -1\ -1\ -1\ 0]^T$.

The result is $\mathbf{x}(k+1) = [6\ \frac{9}{2}\ 2\ \frac{13}{2}\ 5]^T$ and $\mathbf{r}(k+1) = [0\ -\frac{5}{2}\ 1\ \frac{5}{2}\ 0]^T$. Note that the summation of the process $\sum_{i=1}^{5}(x_i(k) + r_i(k)) = 25$ is preserved.

6.3 Numerical results

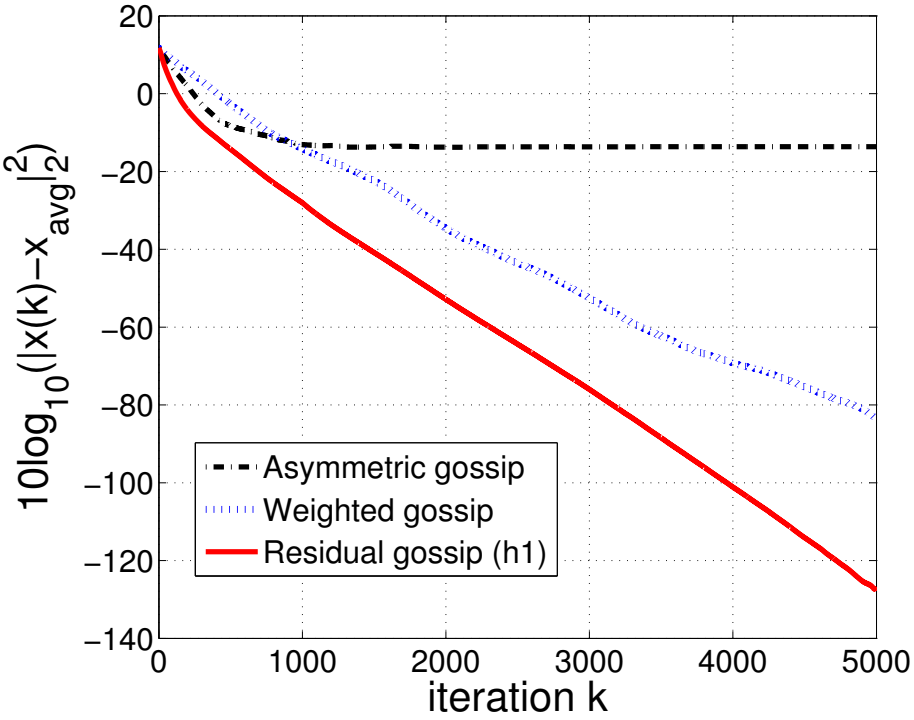
For the experiments conducted in this chapter, we have used a randomly deployed WSN of $N = 50$ nodes inside a $2D$ unit square area. The information is mixed as described in (6.2.4) where the instantaneous topology determines which data is mixed. We average our results over 100 different topologies. The size of the network do not affect the convergence results and our method is totally scalable with the network size in terms of computational resources.

Figure 6.3.1 shows that our residual gossip algorithm converges to the average faster than some existing methods in the related literature. In particular, we evaluate the convergence in terms of the deviation from the average of the initial data. In the unicast scheme of communications, we compare our heuristic h1 with the existing methods of [45] and [25]. In the broadcast scheme of communications, we compare our two heuristics h2 and h3 with

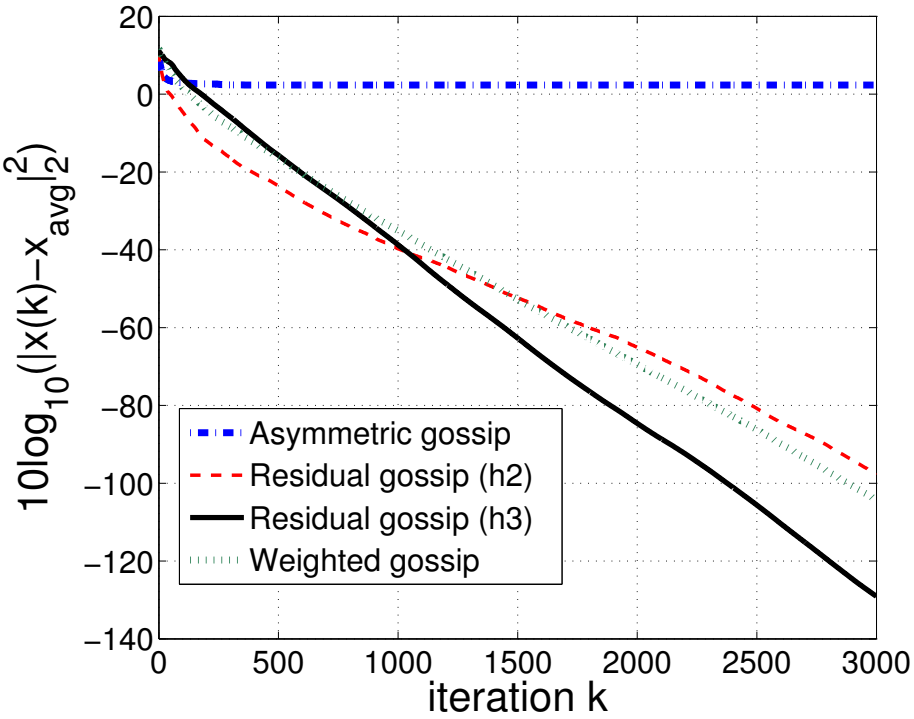
the existing methods of [22] and [53]. Note that the broadcast scheme takes less iterations to converge with similar error. Note that since the convergence time is improved and the exact same links are activated per communication step, the energy is consequently improved with our methodology.

Figure 6.3.2 shows the evolution of the residuals originated by our three heuristics. The norm of the vector $\mathbf{r}(k)$ is evaluated along the iterations. Since the vector $\mathbf{r}(0)$ has all of its entries equal to zero, the norm of this vector grows along the first iterations, until a maximum is reached and from which it starts to rapidly decrease. It is interesting to notice that the method that provides the faster convergence is the one that produces the largest residuals at the beginning of the process. As expected, all the residuals vanish as the iteration number k increases.

Finally, in order to reduce more the convergence time, we could add or subtract some suitable value δ_i to each node state and initialize the residuals with it, such that $x_i = x_i - \delta_i$ and $r_i(0) = \delta_i$.



(a) unicast communications



(b) broadcast communications

Figure 6.3.1: Convergence of $N = 50$ nodes in several randomly deployed networks. (a) Comparison between our unicast residual gossip and the works of [45] and [25]. (b) Comparison between our broadcast residual gossip and the works of [22] and [53].

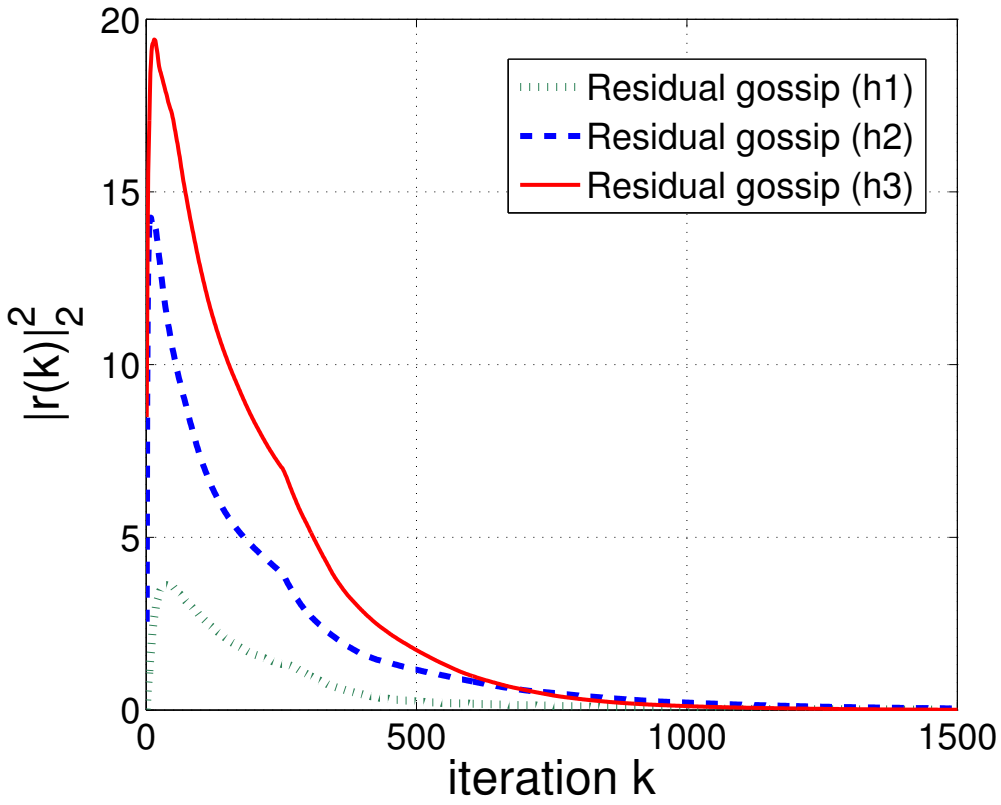


Figure 6.3.2: Evolution of the residuals with the iteration number k for the three heuristics proposed in this chapter. The broadcast heuristic (h3) that provides, in general, the faster convergence is the one that produces the largest residuals at the beginning of the process. As expected, all the residuals vanish as the iteration number k increases.

6.4 Conclusions of the Chapter

In this chapter, we have proposed a new asymmetric gossip algorithm for obtaining almost sure average consensus with fast convergence speed. We have shown that our protocol ensures that the average is preserved along the iterations. Then, we have derived the conditions for our protocol to reach consensus. Both results ensure that our protocol ensures average consensus almost surely. Finally, we have compared the convergence results of our new algorithm with several existing approaches, showing the superior performance of our asymmetric algorithm in both the unicast and broadcast schemes of communication.

Chapter 7

Random Directed Graphs: MSE consensus

7.1 Introduction

Gossip algorithms are a clear example of the inherent capacity for distributed processing that Wireless Sensor Networks (WSNs) possess. As explained in the previous chapter, in these algorithms [31][41], nodes randomly cooperate with each other by means of communications that are generally assumed to be symmetric in order to reach a global common solution. This cooperative technique becomes specially important when the implementation of estimation [81][5] and detection [80][4] tasks must be tackled in a distributed manner under real communications. These are tasks that require a fast response time in order to ensure the correct operation of the upper layer application, which implies, among other requirements, an efficient use of the wireless channel through multiple concurrent transmissions. However, an overuse of the channel might lead to interferences, collisions and other undesired effects, which affect negatively the convergence of the gossip algorithm. In this setting, ensuring bidirectional communications implies that any node j, after a transmission of data to any other node i, should wait until the acknowledgment, including the data from i, is received. In addition, this latter node i should also wait for the acknowledgment from j. Then, every node waiting for an acknowledgment cannot mix information with any other node until the current data exchange is finished. Thus, while a node is waiting, the data re-

ceived from other nodes, must be stored and processed after the current one. This procedure requires a control mechanism that introduces both a communications overhead and an uncontrolled delay in gossip algorithms, which is prohibitive in many real applications.

As explained in the previous chapter, executing gossip algorithms under asymmetric communications ensures, almost surely, consensus to a common (random) value, but this may significantly diverge from the average value. Solutions such as the residual gossip algorithm proposed in the previous chapter can be used to partially solve this problem. However, if the asymmetry in the communications occurs due to uncontrolled collisions and packet losses, certain deviation may still be present. Some conditions for an asymmetric gossip algorithm to reach consensus almost surely have been derived in [82] and an upper-bound on the mean squared error (MSE) has been proposed in [71]. Depending on both the deviation from the average and the error tolerance of the gossip-based application, the response of the whole system may be compromised. Thus, gossip schemes with a satisfactory statistical response under the presence of random collisions and packet losses are also necessary.

Independently of the gossip algorithm proposed [22][45], the order and the frequency of the data exchanges are determined by the chosen Medium Access Control (MAC) protocol. In the context of WSNs, two types of MAC protocols have been widely utilized to control the channel access and deal with the interference. When nodes decide whether to transmit or not based on the presence or absence of certain amount of energy in the channel, the protocol can be classified as a Carrier Sense Multiple Access (CSMA) protocol [47][39][7][37]. Alternatively, when nodes transmit according to certain time-division multiplexing approach, the protocol can be classified as a Time Division Multiple Access (TDMA) protocol [44][23][40][28]. Schemes combining the two previous approaches have recently appeared in the related literature [54][62][73]. These hybrid protocols are aimed at incorporating the main advantages of both traditional schemes, namely, the distributed nature of CSMA protocols, where nodes only use local information, and the time-division methodology proposed by link scheduling protocols, where collisions are completely avoided.

Although the previous MAC alternatives [54][62][73] present good per-

formance in terms of throughput and collision avoidance, if no other control mechanism is considered, the resulting connectivity patterns are asymmetric and random, and the agreement value reached by a gossip algorithm executed over them is a random variable. To the best of our knowledge, the only works considering a MAC protocol for satisfying certain consensus requirements at the application level have been proposed in [85] and [84]. However, these are based on applying the MAC protocol and the gossip algorithm in two differentiated phases, which makes them unsuitable for many real applications.

When designing a new MAC protocol, using accurate radio signal propagation and interference models is fundamental to ensure that collisions are minimized. The interference model is directly related to the complexity of the protocol, being important the distinction between considering only primary interference and taking also into account secondary interference. On the one hand, considering only primary interference implies that two links interfere with each other if and only if they share the receiver. On the other hand, considering also secondary interference implies that every link can affect each other even if they do not share a receiver, which is a much more realistic model. In the first case, the scheduling can be solved in polynomial time [51], while in the second case, it becomes NP-hard [49].

In this chapter, we consider a cross-layer scheme for distributed scheduling based on a carrier sense strategy, that allows an asymmetric gossip algorithm to reach unbiased average consensus in the presence of secondary interference. Based only on local information, such as the sensed power in the channel, nodes are able to self-organize in a time division basis, in such a way that, under the assumption of continuous time, free-collision patterns with a probabilistic symmetric structure are generated. More specifically, every iteration of the gossip algorithm involves several scheduling steps, each of which implies the activation of a link and, following a specific criteria, the inhibition of several other links around it. This criteria, which is the key aspect of the proposed MAC protocol, is designed in a cross-layer basis by considering the performance of the random consensus.

The remainder of this chapter is structured as follows: some necessary background on graph theory and on consensus problems is presented in Section 7.2. In Section 7.3, we present a cross-layer link scheduling protocol based on carrier sense that ensures unbiased average consensus. We then

present, in Section 7.4, some numerical results about the performance of our approach. Finally, the conclusions of the chapter are summarized in Section 7.5.

7.2 Problem definition and background

Previous works [31][41] in gossip algorithms aimed at computing the average of the initial values of the nodes assume that the underlying graph is undirected, namely, the communications between every pair of nodes are symmetric. However, in a realistic scenario, subject to interference, path loss, fading and packet losses, the arising instantaneous topologies are totally random, hence the final consensus value is actually a random variable. In this chapter, we propose a setting where the gossip algorithm is executed over a new cross-layer MAC protocol specially designed to keep the random consensus value unbiased, namely its expectation matches the average of the initial values of the nodes. In order to ensure this property, an appropriate connectivity pattern is created by our MAC protocol before each iteration of the gossip algorithm takes place. As much of the previous work in the related literature [54][62][73], we assume continuous time, such that two transmissions can not start exactly at the same time.

In this section we first introduce the interference model, which provides the framework that rules the design of our proposed MAC protocol. Then, we show the importance of the average consensus in the distributed design of global statistical inference tasks, and present the conditions under which random consensus is unbiased.

7.2.1 Interference model

We consider a network composed of N nodes, each one equipped with an omni-directional antenna, and arbitrary deployed in a square area of L square meters following a uniform distribution. The nodes perform half-duplex communication using a common transmission power P_t. Each pair of nodes is linked by a single user channel affected by the corresponding fading. The work in [7] shows that, in practice, a simplified path loss model is usually enough to capture the essence of the fading effect. The channel gain between

a transmitter node j and a receiver node i is approximated by $\frac{1}{d_{ij}^{\gamma}}$, where $\gamma \geq 2$ is the path-loss exponent and $d_{ji} = d_{ij}$ is the distance that separates nodes i and j.

Additionally, we assume the SINR interference model according to which the successful reception of a packet sent by node j to node i depends on the SINR at node i, that is, a packet between j and i is correctly received if and only if:

$$\frac{\frac{P_t}{d_{ji}^{\gamma}}}{\sum_{v \in \mathcal{V}, v \neq j} \frac{P_t}{d_{iv}^{\gamma}} + N_0} \geq \beta \tag{7.2.1}$$

where N_0 is the background noise and β is a constant threshold. In practice, β is chosen to guarantee a certain quality in the communications. If inequality in (7.2.1) is satisfied for every scheduled link, it is said that a *feasible transmission pattern* has been formed. In this model, all the simultaneous transmissions are considered when evaluating whether a single transmission is valid or not.

In the sequel, we introduce some definitions concerning the different areas that are associated to a transmission.

Definition 1. *The transmission range R_{max} is the maximum distance up to which a packet can be correctly received in absence of interference.*

Taking into account (7.2.1) and in the absence of interference, the maximum transmission radius $R_{\max}$ can be expressed as:

$$R_{\max} = \left(\frac{P_t}{N_0 \beta} \right)^{\frac{1}{\gamma}} \tag{7.2.2}$$

Since a transmission to a node at distance equal to $R_{\max}$ implies that no other link can be simultaneously scheduled without collision, we assume that the length of the links to be scheduled should be shorter than $R_{\max}$. Accordingly, as in [7] and [37], we define also a circular area of radius $R_\rho < R_{\max}$:

Definition 2. *The intended transmission range R_ρ of a node j defines the circular area containing all the neighbors that node j aims at communicating with.*

Therefore, this intended transmission range can be expressed as:

$$R_\rho = \rho R_{\max} = \rho \left(\frac{P_t}{N_0 \beta} \right)^{\frac{1}{\gamma}} \tag{7.2.3}$$

with $0 < \rho < 1$. The value of ρ must be large enough for the network to be connected, namely, in order to guarantee a multi-hop path between every pair of nodes with high probability. It has been shown [50] that the critical radius for connectivity is $L\sqrt{\frac{\log N}{\pi N}}$, so we have that $L\sqrt{\frac{\log N}{\pi N R_{\max}^2}} < \rho < 1$.

Definition 3. *The collision area associated to a specific link between transmitter node j and receiver node i is the circular area of radius Υ_{ji} and centered at i, inside which, no other node can transmit without corrupting the transmission from j to i.*

From (7.2.1), and considering that a collision occurs at node i if SINR $< \beta$, the following expression can be readily obtained:

$$\Upsilon_{ji} = \left(\frac{P_t}{\frac{P_t}{\beta d_{ji}^{\gamma}} - N_0} \right)^{\frac{1}{\gamma}} \tag{7.2.4}$$

All the areas and radii described before ($R_{\max}$, R_ρ, Υ_{ji}) are shown in Figure 7.2.1.

Finally, when a node j performs a transmission, and in order to protect the corresponding link, other nodes around it might decide not to transmit. This concept leads us to the following definition:

Definition 4. *The inhibition area S_{inh}^j of a transmitter node j is defined as the area that includes all nodes that are inhibited due to its transmission.*

The main objective of this area, generally defined as a circle centered at a transmitter node j, is to protect the transmission from node j to any neighbor i, by inhibiting as many nodes from the collision area Υ_{ji} as possible. However, since each area is, in general, centered at a different point, this objective entails a balance between the number of interfering nodes that are not inhibited (hidden terminal), and the number of non-interfering nodes that are inhibited (exposed terminal). As opposed to Υ_{ji}, whose value is completely

determined by the wireless medium and the node transceiver sensibility, the inhibition area S_{inh}^{j} is a design parameter of the protocol, and its exact determination is a key point in the design of any CSMA policy. This issue is tackled in 7.3.2 for our specific protocol design.

7.2.2 Graph Theory

The sequence of instantaneous connectivity patterns that result from applying a MAC protocol can be modeled as a time-varying graph $\mathscr{G}(k) = (\mathscr{V}, \mathscr{E}(k))$, consisting of a constant set $\mathscr{V}$ of N nodes and a set $\mathscr{E}(k) \subset \mathscr{E}$ of directed links that changes at each iteration k. We denote a directed link from node j to node i as e_{ji}, which indicates that there exists a directed information flow from node j to node i. The set $\mathscr{E} = \{e_{ji} \| d_{ji} \leq R_\rho\}$ denotes all the links that are susceptible to be scheduled, that is, the links between pairs of nodes that are at distance lower or equal than R_ρ. While the set $\mathscr{E}$ must include enough links to ensure that the underlaying graph results in a strongly connected graph, as we will explain later, each subset of links $\mathscr{E}(k)$ may correspond to a disconnected one.

Given a time-varying graph $\mathscr{G}(k)$, we can assign an $N \times N$ adjacency matrix $\mathbf{A}(k)$ where an entry $[\mathbf{A}(k)]_{ji}$ is equal to 1 if $e_{ji} \in \mathscr{E}(k)$ and 0 otherwise. The set of neighbors of a node i is defined as $\mathscr{N}_j(k) = \{i \in \mathscr{V} : e_{ji} \in \mathscr{E}(k)\}$ and the degree matrix $\mathbf{D}(k)$ is a diagonal matrix whose entries are given by $[\mathbf{D}(k)]_{jj} = |\mathscr{N}_j(k)|$. Then, the Laplacian of a graph $\mathscr{G}(k)$ is a matrix defined as $\mathbf{L}(k) = \mathbf{D}(k) - \mathbf{A}(k)$, whose second smallest eigenvalue[1] $\lambda_2(\mathbf{L}(k))$ is the so-called algebraic connectivity. In general, and specifically for asymmetric gossip schemes, instantaneous matrices $\mathbf{A}(k)$ and $\mathbf{L}(k)$ are non symmetric. Besides, if we denote by $[\mathbf{P}]_{ji}$ the probability of establishing a link from node j to node i, we have that $\mathbb{E}[\mathbf{A}(k)] = \bar{\mathbf{A}} = \mathbf{P}$, where $\mathbf{P}$ is the connection probability matrix. The main objective of this chapter is to obtain, despite the asymmetry of instantaneous matrices $\mathbf{A}(k)$, a symmetric matrix $\mathbf{P}$. As it is explained in the next subsection, this condition guarantees that the consensus process is unbiased, i.e., it converges in expectation.

[1] We are assuming that the eigenvalues of a instantaneous Laplacian matrix are arranged in increasing order, i.e., $0 = \lambda_1(\mathbf{L}(k)) \leq \lambda_2(\mathbf{L}(k)) \leq ... \leq \lambda_N(\mathbf{L}(k))$.

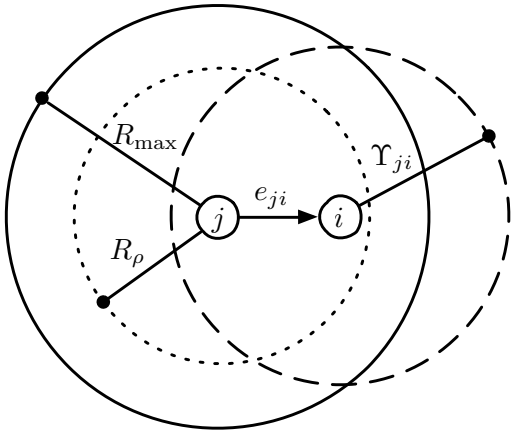


Figure 7.2.1: Relation between the different areas and radii presented in this chapter. In this example, it is assumed that the link between the transmitting node j and the receiving node i is chosen to be activated.

7.2.3 Distributed computation and average consensus

As stated in the previous chapter, at each iteration k of the gossip algorithm, some nodes send their values to randomly chosen neighbors. Thus, every node i that receives a packet from a neighbor j updates its current value according to the expression:

$$x_i(k+1) = (1-\alpha)x_i(k) + \alpha x_j(k) \tag{7.2.5}$$

where α is the step size of the algorithm chosen to ensure convergence. The other nodes remain unchanged.

By using matrix notation, we can write (7.2.5) in a more compact form as:

$$\mathbf{x}(k+1) = \mathbf{W}(k)\mathbf{x}(k) \tag{7.2.6}$$

where:

$$\mathbf{W}(k) = \mathbf{I} - \alpha\mathbf{L}(k) \tag{7.2.7}$$

is the instantaneous weight matrix at the k-th iteration. Therefore, if we form the vector $\mathbf{x}(0) = [x_1(0)\dots x_N(0)]$ with the initial values of the nodes, at each iteration k of the gossip algorithm we have that:

$$\mathbf{x}(k) = \mathbf{W}(k)\dots\mathbf{W}(0)\mathbf{x}(0)$$

The final objective of this gossip algorithm is that all nodes of the network asymptotically compute the average of the initial values of the nodes. In this way, if we define $x_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} x_i(0)$, the goal of the process is that the following holds:

$$\lim_{k\to\infty} \mathbf{x}(k) = x_{\text{avg}}\mathbf{1} \tag{7.2.8}$$

At each iteration k of the algorithm, the deviation is given by the mean square error (MSE), defined as:

$$\text{MSE}(k) = \frac{1}{N}\mathbb{E}\left[\left\|\mathbf{x}(k) - x_{\text{avg}}\mathbf{1}\right\|_2^2\right] \tag{7.2.9}$$

where the expectation is computed over all the realizations of the gossip algorithm.

The matrices $\mathbf{W}(k)$ are random and independent at each iteration. By considering (7.2.7), it becomes clear that $\mathbf{W}(k)$ and $\mathbf{L}(k)$ have the same eigenvectors, and that their eigenvalues are related as: $\lambda_i(\mathbf{W}) = 1 - \alpha\lambda_{N-i+1}(\mathbf{L})$. Therefore, the largest eigenvalue of $\mathbf{W}(k)$ is equal to 1, and the associated right eigenvector is the all-one vector $\mathbf{1}$. However, and due to the random nature of $\mathbf{W}(k)$, nothing can be stated about the associated left eigenvector, which becomes a random vector and is different, in general, for each $\mathbf{W}(k)$. Therefore, and provided that the graph is connected on average, namely $\lambda_2\left(\overline{\mathbf{L}}\right) \neq 0$, where $\overline{\mathbf{L}} = \mathbb{E}[\mathbf{L}(k)]$, we have that [82]:

$$\lim_{k\to\infty} \mathbf{W}(k)\ldots\mathbf{W}(0) = \mathbf{1}\mathbf{m}^T$$

where $\mathbf{m}$ is a random vector with $\sum_{i=1}^{N} m_i = 1$ and $0 \leq m_i \leq 1$. It means that the product of the weight matrices asymptotically converges to a rank-one matrix with equal rows, and (7.2.8) becomes:

$$\lim_{k\to\infty} \mathbf{x}(k) = \mathbf{1}\mathbf{m}^T\mathbf{x}(0)$$

which implies that the nodes reach consensus in the common random value:

$$\hat{x}_{\text{avg}} = \mathbf{m}^T\mathbf{x}(0) = \sum_{i=1}^{N} m_i x_i(0) \tag{7.2.10}$$

Then, by performing the dynamics in (7.2.5), each node is able to compute an estimator $\hat{x}_{\text{avg}}$ of x_{avg}, whose value depends on the left eigenvectors associated to eigenvalue 1 of the weight matrices applied during the process. The performance of the estimator is given by the asymptotic value of the MSE in (7.2.9):

$$\text{MSE}(\hat{x}_{\text{avg}}) = \lim_{k\to\infty} \text{MSE}(k) = \mathbb{E}\left[\left(x_{\text{avg}} - \hat{x}_{\text{avg}}\right)^2\right] \tag{7.2.11}$$

which can be decomposed as follows:

$$\text{MSE}(\hat{x}_{\text{avg}}) = \left(\mathbb{E}\left[x_{\text{avg}} - \hat{x}_{\text{avg}}\right]\right)^2 + \mathbb{E}\left[\left(\hat{x}_{\text{avg}} - \mathbb{E}\left[\hat{x}_{\text{avg}}\right]\right)^2\right]$$

where the first term on the right side denotes the bias of the estimator, and the second term represents its variance. The estimator converges in expectation if the following holds:

$$\mathbb{E}\left[x_{\text{avg}} - \hat{x}_{\text{avg}}\right] = 0$$

that is, its bias is equal to zero. In this case, we say that the estimator is unbiased.

Lemma 7.1 *The estimator $\hat{x}_{avg}$ is unbiased if and only if $\mathbf{P1} = \mathbf{1P}^T$, that is, the sum of the columns of the probability matrix $\mathbf{P}$ is the same as the sum of its rows.*

Proof. If we compute the expectation of the value of $\mathbf{x}(k)$ in the limit, we have the following:

$$\mathbb{E}[\hat{x}_{\text{avg}}\mathbf{1}] = \mathbb{E}[\lim_{k\to\infty} \mathbf{W}(k)\ldots\mathbf{W}(0)\mathbf{x}(0)] = \lim_{k\to\infty} \overline{\mathbf{W}}^k\mathbf{x}(0)$$

where we have applied the independence between the matrices $\mathbf{W}(k)$ and also from vector $\mathbf{x}(0)$. By applying the Perron-Frobenius Theorem, and considering that $\mathbf{1}$ is the right eigenvector of $\overline{\mathbf{W}}$ associated to eigenvalue 1, we have that:

$$\lim_{k\to\infty} \overline{\mathbf{W}}^k = \frac{\mathbf{1}\overline{\mathbf{m}}^T}{\overline{\mathbf{m}}^T\mathbf{1}} \tag{7.2.12}$$

where $\overline{\mathbf{m}}$ is the left eigenvector of $\overline{\mathbf{W}}$ associated to the eigenvalue 1. If we express $\overline{\mathbf{L}} = \overline{\mathbf{D}} - \mathbf{P}$, and by considering that $\mathbf{1}$ is the right eigenvector of $\overline{\mathbf{L}}$

associated to eigenvalue 0, we have that:

$$\mathbf{P1} = \overline{\mathbf{D}}\mathbf{1} \tag{7.2.13}$$

Moreover, since $\overline{\mathbf{W}}$ and $\overline{\mathbf{L}}$ have the same eigenvectors, the following holds:

$$\overline{\mathbf{m}}^T\mathbf{P} = \overline{\mathbf{m}}^T\overline{\mathbf{D}} \tag{7.2.14}$$

From (7.2.13) and (7.2.14), we have that if the sum of the rows of $\mathbf{P}$ is equal to the sum of the columns, then $\mathbf{1P}^T = \mathbf{P1} = \overline{\mathbf{D}}\mathbf{1} = \mathbf{1}\overline{\mathbf{D}}^T$. Therefore, $\overline{\mathbf{m}}$ is the all-one vector, which from (7.2.12) implies that:

$$\mathbb{E}[\hat{x}_{\text{avg}}\mathbf{1}] = \frac{1}{N}\mathbf{1}\mathbf{1}^T\mathbf{x}(0) = x_{\text{avg}}\mathbf{1}$$

which means that the estimator is unbiased.

On the other hand, if the estimator is unbiased, then $\overline{\mathbf{m}}$ must be the all-one vector, which implies that $\mathbf{1P}^T = \mathbf{1}\overline{\mathbf{D}}^T = \overline{\mathbf{D}}\mathbf{1} = \mathbf{P1}$, and the sum of the columns of $\mathbf{P}$ is the same as the sum of its rows. □

Corollary 7.1 *If the probability matrix* $\mathbf{P}$ *is symmetric, the estimator* $\hat{x}_{avg}$ *is unbiased.*

Additionally, from the results in [82], it can be easily proven that the variance of the estimator $\hat{x}_{\text{avg}}$ does not depend on the symmetry of the matrix $\mathbf{P}$. Thus, since a symmetric probability matrix $\mathbf{P}$ guarantees a bias equal to zero, it always results in a smaller value of the MSE.

By considering all the above, the purpose of the present chapter is the design of a MAC protocol that guarantees a symmetric probability connection matrix $\mathbf{P}$, so that the final consensus value computed by the gossip algorithm is an unbiased estimator of the average of the initial values.

7.3 Description of the protocol

This section is devoted to a detailed description of the proposed MAC protocol. After outlining the main features of the scheduling procedure, we present the design rule that guarantees symmetric connection probabilities. Then, we explain how the inhibition area of the nodes can be tuned in such a way that

this design rule is fulfilled. Finally, we show how to compute a closed form expression for the maximum number of simultaneous links that are scheduled at each iteration.

Our proposed scheme involves the simultaneous execution of the link scheduling protocol and the average gossip algorithm. At every iteration k of the gossip algorithm, all the links in $\mathscr{E}$ are initially labeled as unclassified, defining the link demand[2]. At each scheduling step n, a randomly chosen node j sends a packet with its current value to a randomly chosen neighbor i, who mixes both values by using (7.2.5). Accordingly, link e_{ji} is marked as activated, which causes all the unclassified links inside the inhibition area S_{inh}^{j} of transmitter node j to be marked as inhibited. The subsequent repetition of this activation-transmission-inhibition step leads to the following concept:

Definition 5. *The length of a particular feasible transmission pattern, denoted by η, is the number of simultaneous links that has been scheduled in a particular realization while satisfying the feasibility condition (7.2.1) for all of them.*

Therefore, an iteration of the gossip algorithm involves η transmissions that are the result of the scheduled activation of the corresponding η links, such that $|\mathscr{E}(k)| = \eta$.

Accordingly, at each step n of the scheduling protocol, $0 \leq n \leq \eta$, every $e_{ji} \in \mathscr{E}$ is classified into one of the following sets:

- ACTIVATED $\mathscr{A}(n)$: contains all links marked as activated up to the n-th step of the link scheduling protocol. Initially, $\mathscr{A}(0) = \emptyset$. Then, only a new link is marked as activated at each step n, so that, $|\mathscr{A}(n)| = n$. After the η scheduling steps, we have that $\mathscr{E}(k) = \mathscr{A}(\eta)$.

- INHIBITED $\mathscr{I}(n)$: contains all links marked as inhibited up to the n-th step of the link scheduling protocol. At the initial step, we have that $\mathscr{I}(n) = \emptyset$. After the η scheduling steps, we have that $\mathscr{I}(\eta) = \mathscr{E} \setminus \mathscr{E}(k)$.

- UNCLASSIFIED $\mathscr{U}(n)$: contains the links not belonging to any of the previous sets. The initial set is given by $\mathscr{U}(0) = \mathscr{E} = \{e_{ji} | d_{ji} \leq R_\rho\}$.

[2]The link demand at iteration $k+1$ is reestablished after a subset $\mathscr{E}(k)$ of links is activated at iteration k. This means that this demand is the same for every gossip iteration and not all of the links included in this demand has to be satisfied by the scheduling protocol.

From (7.2.5) and according to the previous classification of links, it follows that only the receiving nodes of the links that have been activated after the last η scheduling steps update their values in the k iteration of the gossip algorithm, that is:

$$x_i(k+1) = \begin{cases} (1-\alpha)x_i(k) + \alpha x_j(k) & \text{if } e_{ji} \in \mathscr{A}(\eta) \\ x_i(k) & \text{otherwise} \end{cases}$$

Given this scheduling procedure, in the next subsection, we derive the condition under which symmetric connection probabilities are obtained.

7.3.1 Symmetric connection probabilities

Given a link e_{ji}, if we denote by $\mathscr{S}_{ji}(n)$ the set of unclassified links at step n whose activation imply its inhibition, and based on the previous description of the link scheduling protocol, we have the following result:

Lemma 7.2 *If the number of unclassified links that can inhibit link e_{ji} is the same as the number of those that can inhibit link e_{ij} at every scheduling step l, then we have symmetric probabilities of communication. In other words, if $\left|\mathscr{S}_{ji}(l)\right| = \left|\mathscr{S}_{ij}(l)\right|$ $\forall l = 1 \dots \eta$, then matrix P is symmetric.*

Proof. The probability $P_{ji}^{A}(n)$ of activating a link e_{ji} at the n-th step of the link scheduling is the combination of two events:

- The link e_{ji} remains unclassified after $n-1$ scheduling steps.
- The link e_{ji} is chosen for activation at step n.

Therefore, we can write the following:

$$P_{ji}^{A}(n) = p\left\{e_{ji} \in \mathscr{U}(n-1)\right\} p\left\{e_{ji} \in \mathscr{A}(n) | e_{ji} \in \mathscr{U}(n-1)\right\}$$

The link e_{ji} belongs to $\mathscr{U}(n-1)$ if it does belong neither to $\mathscr{A}(n-1)$ nor to $\mathscr{I}(n-1)$:

$$\begin{aligned} & p\left\{e_{ji} \in \mathscr{U}(n-1)\right\} = \\ & = \left(1 - p\left\{e_{ji} \in \mathscr{A}(n-1)\right\}\right)\left(1 - p\left\{e_{ji} \in \mathscr{I}(n-1)\right\}\right) \end{aligned}$$

The link e_{ji} does not belong to $\mathscr{A}(n-1)$ if it has not been activated in any of the previous $n-1$ steps:

$$\left(1-p\left\{e_{ji}\in\mathscr{A}(n-1)\right\}\right)=\prod_{l=1}^{n-1}\left(1-P_{ji}^{A}(l)\right)$$

Similarly, the link e_{ji} does not belong to $\mathscr{I}(n-1)$ if it has not been inhibited in any previous step. Furthermore, the link e_{ji} is inhibited at step l if a link from $\mathscr{S}_{ji}(l)$, among all candidates contained in $\mathscr{U}(l)$, is chosen for activation. Since both $\left|\mathscr{S}_{ji}(l)\right|$ and $|\mathscr{U}(l)|$ are random variables, we have the following:

$$\begin{aligned}
&\left(1-p\left\{e_{ji}\in\mathscr{I}(n-1)\right\}\right)=\\
&=\prod_{l=1}^{n-1}\left(1-\sum_{\tau,\upsilon}\frac{\tau}{\upsilon}p\left\{\left|\mathscr{S}_{ji}(l)\right|=\tau,|\mathscr{U}(l)|=\upsilon\right\}\right)=\\
&=\prod_{l=1}^{n-1}\left(1-\mathbb{E}\left[\frac{\left|\mathscr{S}_{ji}(l)\right|}{|\mathscr{U}(l)|}\right]\right)
\end{aligned}$$

The probability that link e_{ji} is chosen for activation at scheduling step n, provided that it has remained unclassified during the previous $n-1$ steps, can be expressed as:

$$\begin{aligned}
&p\left\{e_{ji}\in\mathscr{A}(n)|e_{ji}\in|\mathscr{U}(n)|\right\}=\\
&=\sum_{\upsilon}\frac{1}{\upsilon}p\left\{|\mathscr{U}(n)|=\upsilon\right\}=\\
&=\mathbb{E}\left[\frac{1}{|\mathscr{U}(n)|}\right]
\end{aligned}$$

Accordingly, the probability of activation of a specific link e_{ji} at the n-th step of the scheduling process is given by the following recursive expression:

$$P_{ji}^{A}(n)=\mathbb{E}\left[\frac{1}{|\mathscr{U}(n)|}\right]\cdot\prod_{l=1}^{n-1}\left(1-\mathbb{E}\left[\frac{\left|\mathscr{S}_{ji}(l)\right|}{|\mathscr{U}(l)|}\right]\right)\left(1-P_{ji}^{A}(l)\right)$$

Thus, it follows that if $\left|\mathscr{S}_{ji}(l)\right|=\left|\mathscr{S}_{ij}(l)\right|$ for $l=1\ldots\eta$, then $P_{ji}^{A}(n)=P_{ij}^{A}(n)$ for $n=1\ldots\eta$. Then, since $[\mathbf{P}]_{ij}=\sum_{n=1}^{\eta}P_{ij}^{A}(n)$, it is accomplished that $[\mathbf{P}]_{ij}=[\mathbf{P}]_{ji}$, which concludes the proof.

□

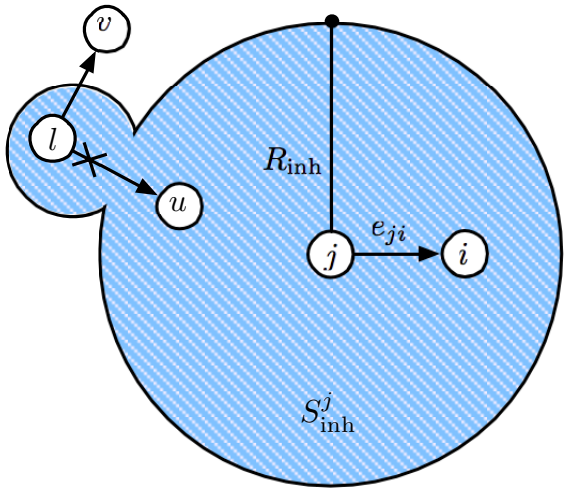


Figure 7.3.1: Inhibition area S^j_{inh} of node j, as it is defined in **Proposition 7.1**. S^j_{inh} covers every link with an endpoint inside the circle of radius R_{inh} centered at j. Consequently, any node u located at a distance d_{ju} shorter than R_{inh} from node j is inhibited for both sending and receiving.

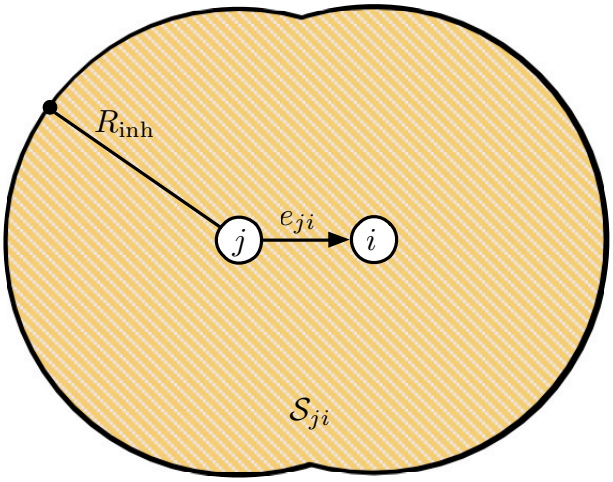


Figure 7.3.2: Set of nodes $\mathcal{S}_{ji}$ whose activation would cause the inhibition of the transmission from node j to i. Clearly, this set is the same as the one that includes the nodes whose activation would inhibit the transmission from i to j.

7.3.2 Design of the inhibition area

The inhibition area S^j_{inh} of a node j, namely the nodes that are inhibited for transmitting while this node j is performing a transmission, is a key design

aspect in any CSMA protocol, since it entails a trade-off between the hidden and the exposed terminal problems. As explained in Section 7.3, in order to completely avoid collisions and generate feasible transmission patterns, the inhibition area S^j_{inh} of a node j should cover the collision area Υ_{ji} of node i, inhibiting the transmission of any neighbor of node i. However, in our specific setting, the inhibition area takes an additional role: it must be designed in such a way that **Lemma 7.2** holds, so that a symmetric probability matrix **P** is ensured. Accordingly, and assuming continuous time so that two nodes cannot transmit exactly at the same time, we propose the following result:

Proposition 7.1 *Given any active link e_{ji}, if every other link with an endpoint at a distance from the transmitter j shorter than the inhibition radius R_{inh} given by:*

$$R_{inh} = \left(\frac{(\eta - 1)P_t}{\frac{N_0}{\rho^\gamma} - N_0} \right)^{\frac{1}{\gamma}} + R_\rho \tag{7.3.1}$$

is inhibited, then feasible transmission patterns with an associated symmetric probability matrix **P** *are generated.*

Proof. We first show that symmetric probabilities of connection are generated. From what is stated in **Proposition 7.1**, link e_{ji} is inhibited if either node j or node i are located at a distance shorter than (7.3.1) from an active transmitter u. Therefore, the set of links $\mathscr{S}_{ji}(n)$, whose activation implies the inhibition of link e_{ji}, becomes:

$$\mathscr{S}_{ji}(n) = \left\{ e_{uv} : \min\left\{ d_{uj}, d_{ui} \right\} \leq R_{\text{inh}} \right\} \tag{7.3.2}$$

Since the inhibition radius in (7.3.1) is a common value to all links, then $\mathscr{S}_{ji}(n) = \mathscr{S}_{ij}(n)$, which, by **Lemma 7.2**, implies that $P^A_{ji}(n) = P^A_{ij}(n)$, for $n = 1 \ldots \eta$.

Now, we prove that, with the defined strategy, collision free patterns are obtained. Since the maximum number of simultaneous active links is η, the collision area for a link between j and i defined by (7.2.4) must be redefined for the case of $\eta - 1$ interferers. Besides, since the distance between those $\eta - 1$ interferers and the receiver i cannot be predicted, we must assume the worst case scenario, that is, the $\eta - 1$ interferers are located at the same

distance, and as close as possible to node i. By considering this worst case scenario, the radius for this new collision area can be computed from (7.2.1) as:

$$\Upsilon_{ji}^{\eta} = \left(\frac{\beta(\eta-1)P_t}{\frac{P_t}{d_{ji}^{\gamma}} - N_0\beta} \right)^{\frac{1}{\gamma}} \tag{7.3.3}$$

This collision area is protected if the transmission from node j to node i inhibits the transmission of any other node in a circular area centered at j with radius $\Upsilon_{ji}^{\eta} + d_{ji}$. As explained before, a condition to ensure symmetric probabilities of communication is that the inhibition radius R_{inh} is common to every node j, hence the associated inhibition area S_{inh}^{j} should cover the collision area of the longest link in the network. Since $R_{\rho} \geq \max_{e_{ji} \in \mathscr{E}} \{d_{ji}\}$, the expression for R_{inh} in (7.3.1) implies that $R_{\text{inh}} \geq \max_{e_{ji} \in \mathscr{E}} \{\Upsilon_{ji} + d_{ji}\}$, and a feasible pattern is generated as a consequence, while symmetric probabilities of communication are ensured. □

Remark 1. *In the proposed strategy, the concept of inhibition area is reformulated, based on the inhibition of links instead of nodes. Figure 7.3.1 shows the inhibition area S_{inh}^{j} of node j based on the radius R_{inh}. Node l is inhibited to transmit to u, but it is still able to transmit to node v.*

Remark 2. *Based on this strategy, a transmission from node j to node i is inhibited not only by the activation of any node inside a circular area of radius R_{inh} centered at j, but also by the activation of any node inside the same area centered at the intended receiver i (Figure 7.3.2). It is straightforward to see that the previous set of nodes is the same one that inhibits the transmission from node i to node j.*

Our algorithms works as follows, the sets $\mathscr{A}(0)$ and $\mathscr{I}(0)$ are initialized with no links on them, while the set $\mathscr{U}(0)$ is initialized with every link in $\mathscr{E}$. At each scheduling step, a link between a transmitting node j and a receiving node i is randomly chosen from the set $\mathscr{U}(n)$ and this unique link is added to the set of active links $\mathscr{A}(n)$ at scheduling step n. Then, the activation of this new link implies that no other transmission is simultaneously scheduled in the inhibition area around the receiving node i. Hence, these links are added to the set $\mathscr{I}(n)$ and are removed from the set $\mathscr{U}(n)$. Given a

feasible set of links, let us denote $\eta_{\max}$ the maximum value of η that ensures that no other link could be additionally scheduled. Since the parameter η is estimated, there are two scenarios for finishing the link scheduling: 1) if the estimated value $\hat{\eta}$ is larger than the maximum one $\eta_{\max}$, the algorithm stops when the set $\mathscr{U}(n)$ is empty and 2) if the estimated value $\hat{\eta}$ is smaller than the maximum one $\eta_{\max}$, the link scheduling protocol is finalized when $\hat{\eta}$ simultaneous links are obtained. Both stoping conditions satisfy the feasibility of the transmission pattern.

Algorithm 8 Link scheduling described in terms of $\mathscr{A}$, $\mathscr{I}$, $\mathscr{U}$

Require: $\mathscr{A}(0)$, $\mathscr{I}(0)$, $\mathscr{U}(0)$, $\hat{\eta}$
Ensure: $\mathscr{U}(n)$ is empty OR $n = \hat{\eta}$
  $n = 0$
  while $\mathscr{U}(n)$ is not empty AND $n \leq \hat{\eta}$ **do**
    e_{ji} = choose uniformly at random a link from $\mathscr{U}(n)$
    add e_{ji} to $\mathscr{A}(n)$
    remove e_{ji} from $\mathscr{U}(n)$
    for $u = 1$ to N **do**
      for $v = 1$ to N **do**
        if $d_{uj} \leq R_{\text{inh}}$ OR $d_{vj} \leq R_{\text{inh}}$ **then**
          add e_{uv} to $\mathscr{I}(n)$
          remove e_{uv} from $\mathscr{U}(n)$
    $n = n + 1$
  end while

7.3.3 Computation of the length of the scheduling

Having a good approximation of the maximum number of simultaneous links is needed for an optimal operation of our link scheduling protocol. If the number of simultaneous links $\hat{\eta}$ estimated by our protocol is too large, the inhibition radius becomes longer than necessary and the resulting number of simultaneous links is reduced as a consequence. Oppositely, if this number $\hat{\eta}$ is too small, the protocol stops after $\hat{\eta}$ links are scheduled, so that, it is still able to ensure that collisions are avoided.

In this section, we propose an iterative method for obtaining an approximation to both the number of simultaneous links η and the inhibition radius R_{inh}. For this purpose, we assume the densest packing of interferers, which corresponds to the hexagon packing [47], as shown in Figure 7.3.3. This

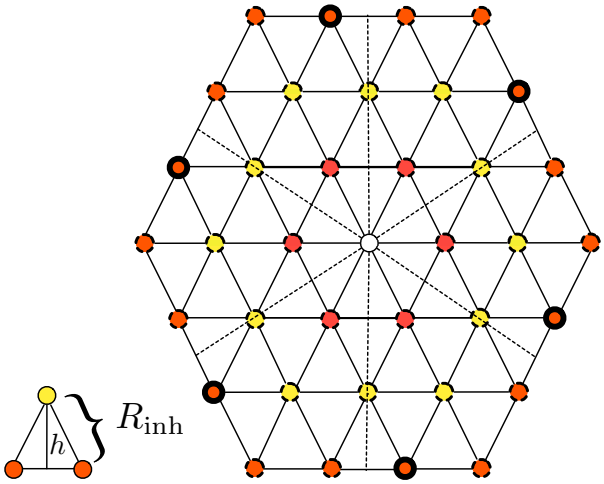


Figure 7.3.3: The densest packing of interferers around a particular transmitter (white circle). We differentiate between odd levels (red circles) and even levels (yellow circles). We argue in terms of the equilateral triangles created, where R_{inh} is their side and h denotes their height.

packing leads to multiple interferers located in odd and even levels around the receiver. The maximum number of levels, denoted by l_{MAX}, depends of the value of R_{inh} and can be obtained as $l_{\text{MAX}} = \sqrt{2}L/R_{\text{inh}}$, where L is the side of the deployment area. The objective is to obtain the number of virtual transmitters located at distance R_{inh} that would produce the equivalent interference to the transmitters appeared in all levels $l = 1, 2, \ldots, l_{\text{MAX}}$ (see Figure 7.3.3).

Proposition 7.2 *For a given value of l_{MAX} and considering the hexagon packing, the maximum value of η is given by:*

$$\eta = 6 \sum_{\ell=1}^{l_{\text{MAX}}} \sum_{m=0}^{\lfloor \ell/2 \rfloor} \left((\ell/2 - \lfloor \ell/2 \rfloor + m)^2 + (\ell h)^2 \right)^{-\frac{\gamma}{2}} + \sum_{m=2}^{\lfloor \ell/2 \rfloor} \left((\ell/2 - \lfloor \ell/2 \rfloor + m - 1)^2 + (\ell h)^2 \right)^{-\frac{\gamma}{2}} + \left((1/4) + (\ell h)^2 \right)^{-\frac{\gamma}{2}} \quad (7.3.4)$$

Proof. According to **Proposition 7.2**, we need to calculate the maximum number of interferers that can be placed at distance R_{inh} from a particular transmitter without producing a collision. However, since these interferers must be placed in such a way that they do not produce collisions to each other, these are placed in levels that are at distance proportional to R_{inh}, as

shown in Figure 7.3.3. In order to express everything as a function of R_{inh}, we make the calculations for odd and even levels separately. All the equations are expressed as a function of the squared triangles of side lengthR_{inh}.

First, the interference produced by the dotted nodes that belong to odd levels in Figure 7.3.3 is determined by the following expression:

$$\sum_{m=0}^{\lfloor \ell/2 \rfloor} \frac{6}{\left(\left(\frac{1}{2}+m\right)^2 + (\ell h)^2 \right)^{\frac{\gamma}{2}}} \tag{7.3.5}$$

where m gives the relative position of the current node in terms of triangles, ℓ denotes the level being considered and $h = \sqrt{3/4}$. Note that we have omitted the dependence of all terms on R_{inh}, e.g., the value of h would be $h = \sqrt{3/4}R_{\text{inh}}$. The equivalent expression for the even level is:

$$\sum_{m=0}^{\ell/2} \frac{6}{\left(m^2 + (\ell h)^2 \right)^{\frac{\gamma}{2}}} \tag{7.3.6}$$

Combining equations (7.3.5) and (7.3.6), we obtain:

$$\sum_{m=0}^{\lfloor \ell/2 \rfloor} \frac{6}{\left((\ell/2 - \lfloor \ell/2 \rfloor + m)^2 + (\ell h)^2 \right)^{\frac{\gamma}{2}}} \tag{7.3.7}$$

We also have to consider the nodes that appear in the odd levels drawn with a solid line in Figure 7.3.3. The expression that relates the distance of these nodes with R_{inh} is the following:

$$\sum_{m=1}^{\lfloor \ell/2 \rfloor} \frac{6}{\left(\left(m-\frac{1}{2}\right)^2 + (\ell h)^2 \right)^{\frac{\gamma}{2}}} \tag{7.3.8}$$

The equivalent expression for the even level is:

$$\sum_{m=2}^{\ell/2} \frac{6}{\left((m-1)^2 + (\ell h)^2 \right)^{\frac{\gamma}{2}}} \tag{7.3.9}$$

Combining equations (7.3.8) and (7.3.9), we obtain:

$$\sum_{m=2}^{\lfloor \ell/2 \rfloor} \frac{6}{\left((\ell/2 - \lfloor \ell/2 \rfloor + m - 1)^2 + (\ell h)^2\right)^{\frac{\gamma}{2}}} + \frac{6}{((1/2)^2 + (\ell h)^2)^{\frac{\gamma}{2}}} \tag{7.3.10}$$

Combining equations (7.3.7) and (7.3.10), we obtain the final equation for the parameter η given by (7.3.4). □

Since l_{MAX} depends on R_{inh}, which is also unknown, we propose **Algorithm 2** to iteratively calculate the values of R_{inh} and $\hat{\eta}$.

Algorithm 9 $\hat{\eta}$ and R_{inh} calculation

Require: γ
Ensure: R_{inh} and $\hat{\eta}$ are valid in the hexagon packing
  $\hat{\eta} = 1$
  while $\hat{\eta}$ changes its value **do**
    obtain R_{inh} using expression (7.3.1)
    $l_{\mathrm{MAX}} = \sqrt{2}L/R_{\mathrm{inh}}$
    update $\hat{\eta}$ using expression (7.3.4)
  end while

Algorithm 2 iteratively obtains an approximation of the parameters η and R_{inh} by successively refining both estimations. For example, if we initialize the value of $\hat{\eta} = 1$, the corresponding value of R_{inh} is also small because a small number of simultaneous transmitters is expected. Consequently, a large value of l_{MAX} is obtained because several level fit in the deployment area with the packing shown in Figure 7.3.3. Then, applying (7.3.4), we obtain a larger value of $\hat{\eta}$, which is then used again to obtain R_{inh} and so on.

7.4 Numerical results

In this section, we numerically evaluate the performance of the asymmetric gossip algorithm under the cross-layer link scheduling protocol presented in this chapter. Our main goal is to evaluate the MSE and the convergence rate for different parameters: P_t, η, α and ρ, showing, in addition, that an unbiased consensus value reduces the MSE.

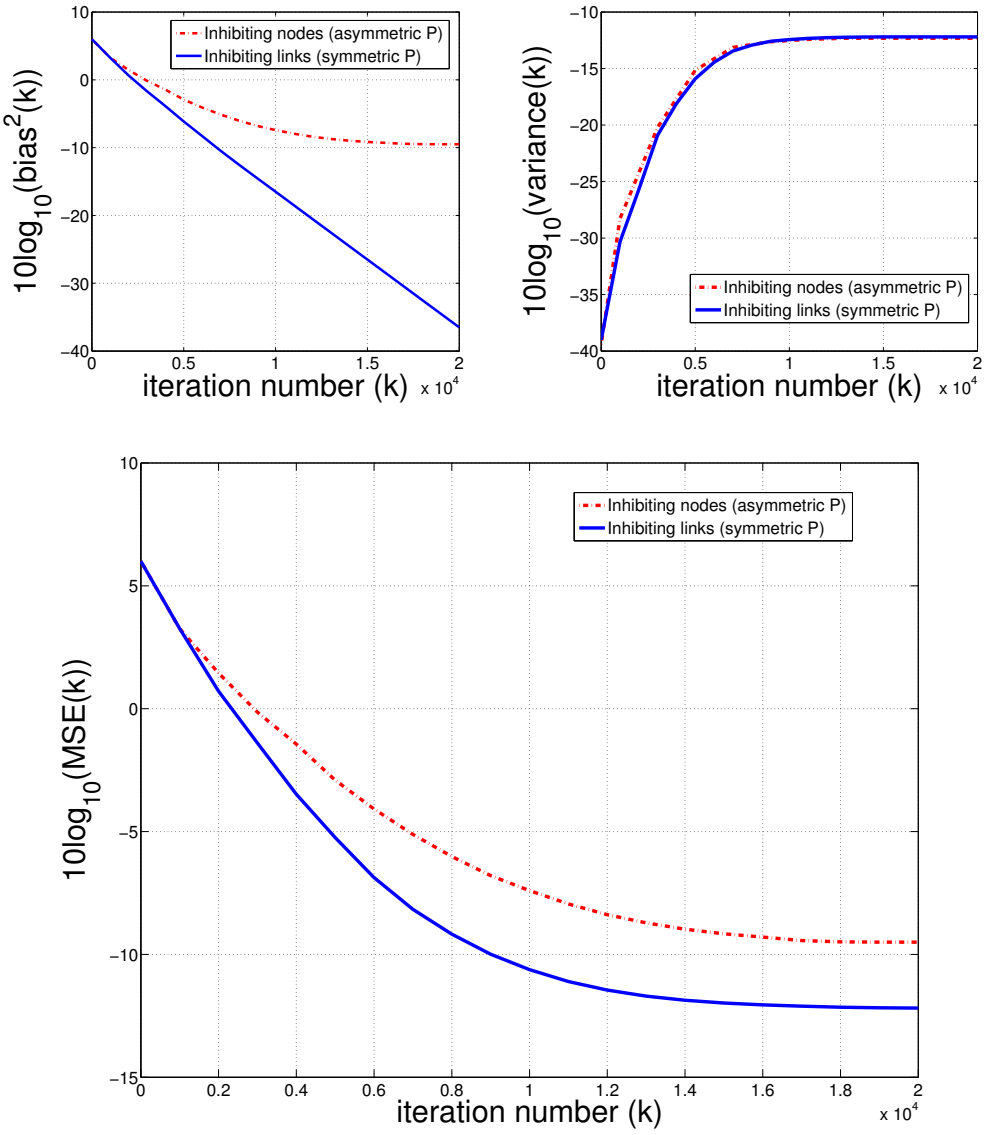


Figure 7.4.1: Bias, variance and MSE of the cross-layer scheme proposed in this chapter when both nodes and links are inhibited. The network setting used is: $P_t = -2$ dBm, $\rho = 0.5$ and $\alpha = 0.005$.

7.4.1 Simulation Scenario

We model a WSN as a uniformly random deployed network of $N = 1000$ nodes inside a $2D$ unit square area. The information is mixed as described in (7.2.5), where the instantaneous topology at the k-th iteration determines which data is mixed. Additionally, channel gains are computed based on node positions, and on the radio propagation model. Radio signal propagation is assumed to follow log-normal shadowing, with path loss exponent $\gamma = 3$. For a given background noise $N_0 = 10^{-9}$mW per meter and a given value of $\beta = 10$, we choose a combination of the values P_t and ρ that ensures connectivity on average, that is, the set of edges $\mathscr{E}$ contains enough links to ensure the existence of a multi-hop path between every pair of nodes. Finally, since the value α plays a crucial role, we analyze the consensus performance for different values of this parameter.

7.4.2 Numerical Results

Figure 7.4.1 shows the bias, the variance and the MSE for the cross-layer methodology proposed in this chapter. In order to illustrate the importance of obtaining a symmetric matrix $\mathbf{P}$, we have compared two different settings that lead to different values of the MSE. In the first one, we have used the traditional approach of inhibiting nodes (under the inhibition radius R_{inh}), leading to an asymmetric matrix $\mathbf{P}$. In the second one, we have used, the results of **Proposition 7.1**, inhibiting links and ensuring a symmetric matrix $\mathbf{P}$. When the matrix $\mathbf{P}$ is ensured to be symmetric, the asymmetric gossip converges in expectation, namely, the term of the bias tends to zero. Accordingly, the MSE converge to the value of the variance. Notice that the variance depends on several factors, such as the variance of the initial data. Oppositely, when the matrix $\mathbf{P}$ is asymmetric, the bias does not vanishes and it dominates the value of the MSE, since it present much larger values than the variance.

Figure 7.4.2 shows that the convergence speed and the final MSE value are slightly influenced by ρ and P_t. This is explained by the fact that our link scheduling protocol produces denser connectivity patterns when the average length of links is decreased by a reduction on the values of P_t or ρ and vice versa. It means that, in both cases, the information is mixed at similar rates,

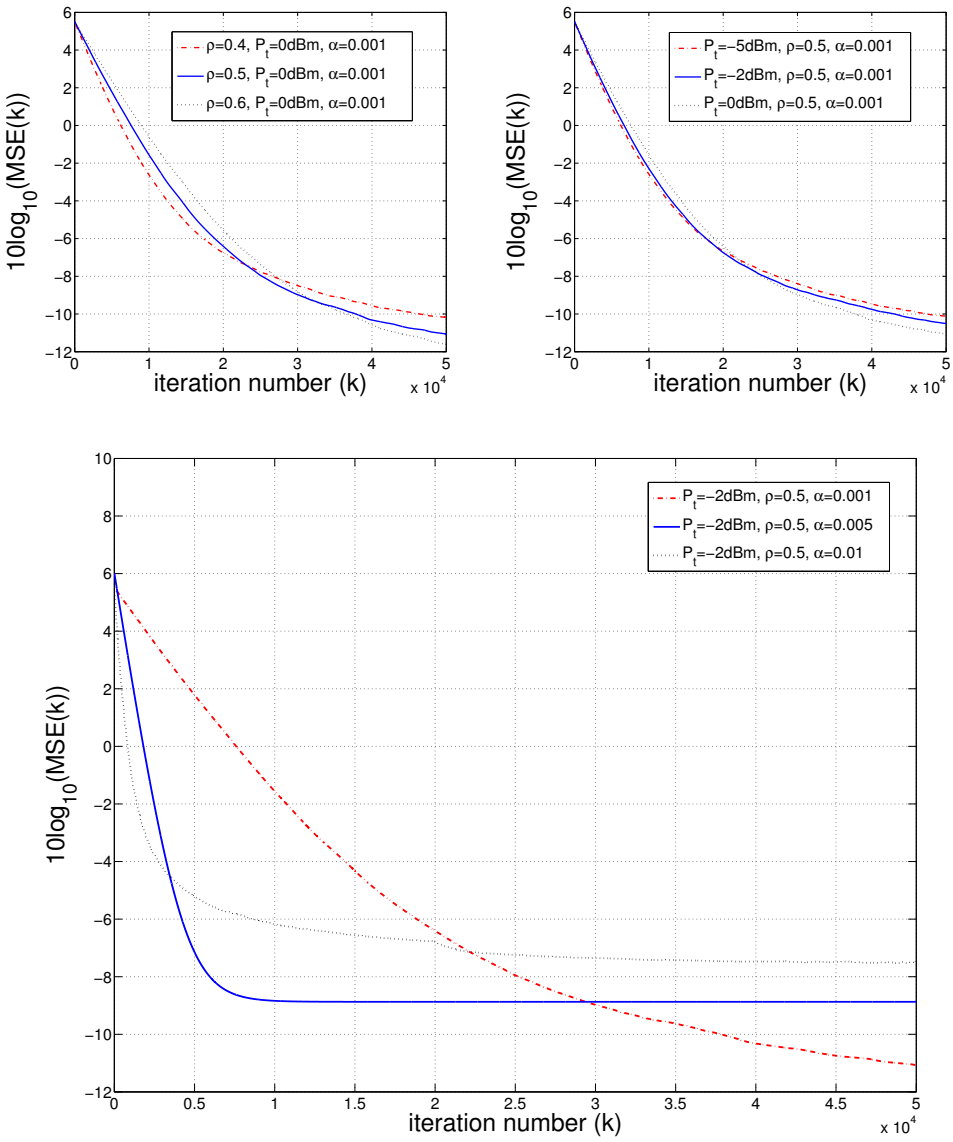


Figure 7.4.2: Evolution of the MSE value for different combinations of the parameters: α, P_t and ρ.

namely, when denser patterns are generated, there are more data exchanges between nodes, but these exchanges are more locally done than in the case of having less but larger links. As a consequence, the consensus performance in terms of convergence speed and accuracy is almost independent of the parameters ρ and P_t. Note that for a given value of these parameters P_t and ρ, as larger the resulting value of η is, the faster convergence we obtain. Therefore, for a fixed average link size, the value of η is a measure about how good the link scheduling is for the convergence rate of the consensus process. Figure 7.4.2 also shows the influence of α in the existing trade-off between the convergence rate and the accuracy of the consensus process. It is clear from this figure that larger values of α lead to faster convergence speed but also to smaller precision reflected in larger values of the MSE, and vice versa.

Figure 7.4.3 shows that the convergence speed clearly depends on the number of simultaneous links scheduled. Note that we are ensuring to have the same average link length in all the curves plotted, that is, the different number of simultaneous links is not altered by P_t or ρ, but it depends on the medium access protocol being used. Accordingly, faster convergence rate is obtained when the medium access protocol is able to schedule more links at each iteration k of the consensus process. Moreover, the resulting MSE is the same for the three average values of η, which ensures that we are not having any side effect that worsens the accuracy of the gossip algorithm.

Finally, Figure 7.4.4 shows the influence of the medium access control protocol in the convergence rate and the accuracy of the consensus process. We execute two versions of our protocol, one ensuring collision avoidance and symmetric probabilities of communication and another ensuring only collision avoidance. We also compare these two approaches with a constant threshold technique, which is widely used in the hardware of real motes, where a node is allowed to transmit as long as the measured signal strength RSSI ≤ -90 dBm. It is clear from Figure 7.4.4 that ensuring an unbiased consensus value tend to reduce the MSE.

7.4.3 Toy example

In this section, we show a complete example of the operation of our protocol. In this example, **Lemma 7.2** is fulfilled by applying the result in **Proposition 7.1**, so that a symmetric matrix **P** is obtained.. For the sake of simplicity, we consider a ring of $N = 1000$ nodes inside a square area of side $L = 1000$ m. The nodes are equally spaced and transmit with a power of $P_t = -10$dBm. We assume the path loss exponent $\gamma = 3$, the background noise $N_0 = 10^{-9}$ and $\rho = 0.2$.

In this setting, the maximum distance to which a node can transmit a message is $\rho R_{\max} = 43$m. **Algorithm 2** gives in the first iteration $R_{\text{inh}} = 136$m and $\eta = 65$. In the second iteration, $R_{\text{inh}} = 417$m and $\eta = 18$. After a few more iterations, we reach the final values of $R_{\text{inh}} = 317$m and $\eta = 25$. Accordingly, the perimeter of the ring is 31416m, which means that the distance that separates two consecutive nodes is 3.1416m.

These values mean that each node can communicate with 26 neighbors, so that, there are $M = 26000$ links that can be scheduled at each iteration k of the consensus algorithm. This implies that $\mathscr{U}(1)$ and $S_{ij}(1)$ contain 26000 and 399 links respectively. The probability of activation of any link in the first scheduling step is:

$$P_{ij}^{A}(1) = \mathbb{E}\left[\frac{1}{|\mathscr{U}(1)|}\right] = \frac{1}{26000}$$

In the second scheduling step, the number of links in $\mathscr{U}(2)$ is $26000 - 399 - 1 = 25600$ (399 inhibited and 1 activated) with probability equal to one. Thus, the activation probability of any ink in the network in the second scheduling step is:

$$\begin{aligned} P_{ij}^{A}(2) &= \mathbb{E}\left[\frac{1}{|\mathscr{U}(2)|}\right]\left(1 - \mathbb{E}\left[\frac{|\mathscr{S}_{ij}(1)|}{|\mathscr{U}(1)|}\right]\right)(1 - P_{ij}^{A}(1)) \\ &= \frac{1}{25600}\left(1 - \frac{399}{26000}\right)\left(1 - \frac{1}{26000}\right) \end{aligned}$$

Similar calculations can be made for $n = 1 \ldots \eta$.

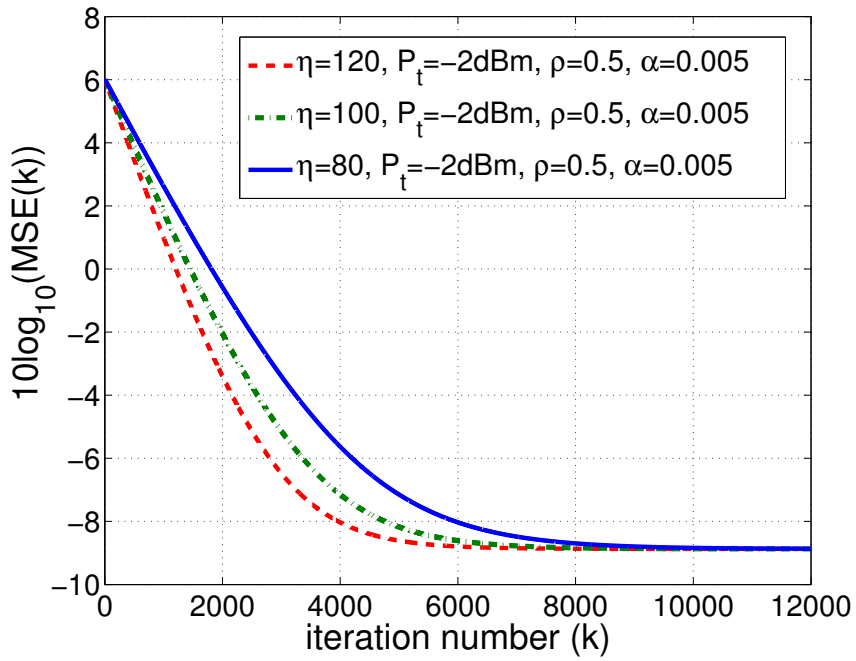


Figure 7.4.3: Evolution of the MSE value for different values of the parameter η.

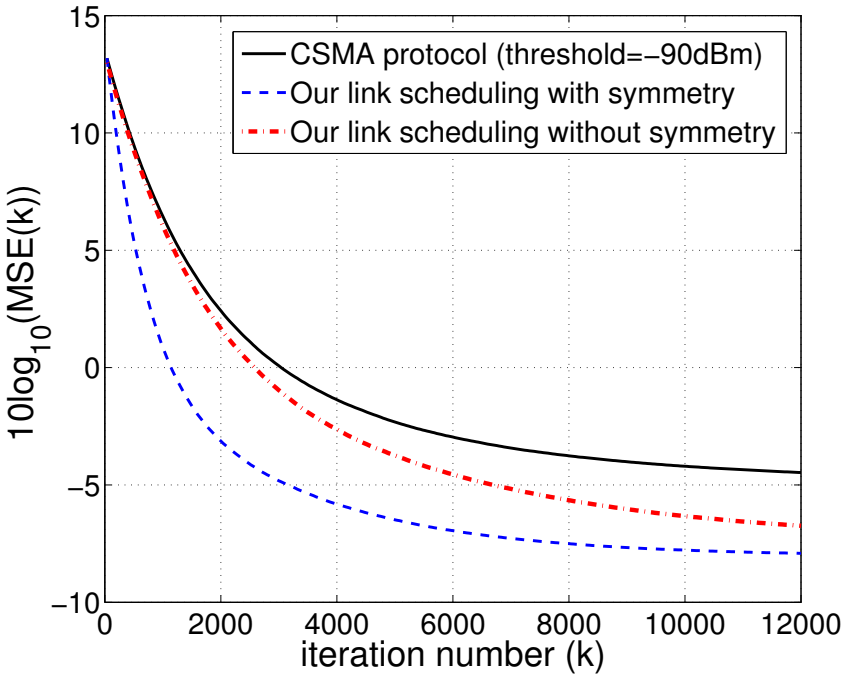


Figure 7.4.4: Evolution of the MSE value for different medium access control protocols.

7.5 Conclusions of the Chapter

In this chapter, we have considered the problem of average consensus in WSNs under an accurate interference model, in which correct packet reception at a receiving node depends on the SINR. We propose a decentralized link scheduling that leads to connectivity patterns that are free of collisions. Since these are randomly generated and the interferences constrain the set of simultaneous links, the resulting connectivity patterns at each iteration are asymmetric. We have analyzed the performance of the consensus algorithm under a probabilistic point of view and we have devised a probabilistic model where connection between any pair of nodes is symmetric on average, ensuring unbiased average consensus. Moreover, we numerically show how the network parameters affect the convergence rate and the error of the consensus algorithms.

Part IV

Conclusions and open problems

Chapter 8

Conclusions and Open Problems

The work of this dissertation was set out to make several consensus-based tasks efficient in terms of several parameters and under different communication and network settings. The main performance parameters considered are the convergence time[1], the associated power consumption of the nodes and the deviation from the target average value. These parameters are crucial since they determine the energy consumption and the accuracy of the algorithm. On the one hand, the energy consumption is important for the energy constrained sensor networks widely found in practice. On the other hand, the accuracy of the algorithm is crucial for the operation of the consensus-based tasks, since a large error may compromise the response of the whole system.

In the presence of static networks, where convergence is easily ensured, we have devised several methodologies to optimize the underlying network topology in terms of several energy related functions that are relevant in the area of sensor networks. Alternatively, when the connectivity of the network varies over time, since the convergence conditions for the previous case does not longer apply, we design new methodologies that affect the instantaneous network topologies to reach certain accuracy and performance.

Although the methodologies in static and random scenarios are different, both are mainly based on designing how the communications between neighboring nodes are performed.

[1]The value of convergence can be the deterministic average value or a random value.

8.1 Conclusions

This dissertation focuses on ensuring certain energy efficiency when tasks based on consensus are executed over static sensor networks. While most of the related work focuses on the convergence time of consensus algorithms, we also improve the power consumption of individual nodes to improve several energy related functions that maximize the lifetime of the network when consensus-based tasks are executed. There are many practical advantages of improving consensus algorithms. First, a reduced convergence time allows to obtain the estimation of needed data with certain accuracy before it may be required by certain time-constrained applications, such as target tracking applications. Second, energy efficiency may reduce the maintenance cost, since it reduces the battery changes per unit time. The devices may be located in spots of difficult access and the cost of mobilizing personal to replace the batteries may be high. Finally, the consensus-based application may run for longer periods of time with the required accuracy. We also apply these new optimization methodologies to different network settings, showing that depending on certain parameters, several well-known complex networks may appear as the best energy efficient solutions for consensus-based applications. In particular, the parameters that lead to the appearance of Random Geometric Networks, Small World Networks and Scale Free Networks have been studied in this dissertation. While most of the existing results about complex networks are structural, we have investigated the spatial characterisation of these networks filling in part this existing gap.

When the communications are random and the network topology varies over time, we mainly focus on ensuring that the average consensus algorithm converges with certain degree of accuracy to the average value. This alternative case, also known as the gossip algorithm, reaches a random value that may differ from the average when symmetric communications cannot be satisfied in every instantaneous topology. To solve this issue, we propose a new algorithm that ensures to reach the average value almost surely. Certain amount of accuracy is important since the consensus-based task may need it to operate normally. Moreover, we show that our algorithm is faster than existing works in the related literature, reducing the energy consumption of this process and ensuring a longer lifetime of sensor networks that are energy

constrained.

Finally, in the presence of random interferences, where collisions must be avoided while ensuring consensus with certain degree of accuracy, we propose a new cross-layer MAC protocol to satisfy both requirements. In this setting,we propose a new distributed link scheduling protocol that ensures convergence to the average in the mean square sense. This last scenario is commonly found in sensor networks, where complex control mechanisms are not always possible and probabilistic solutions must be chosen instead. In order to improve the convergence time in this setting, we also focus on maximizing the spatial reuse of the channel. This increases the number of correct communications per time step, which intuitively tends to reduce the energy consumption of the process. As stated before, fast and energy efficient networks are desired for reducing the maintenance cost and increasing the operation time.

8.2 Future Work

In this work, we have proposed several methodologies to improve the performance of consensus and gossip algorithms in terms of different parameters, under different scenarios and different communication settings. While for static network graphs most of the open problems about energy efficiency of consensus-based tasks have been tackled in this work, for random network graphs there is still much work to do. In particular, the definition of the expected energy consumption and a well defined methodology for optimizing this parameter is still to be investigated. Besides the randomness of every instantaneous topology, the major difficulty behind this research line is that the total number of these topologies is also random because the convergence time becomes a random variable.

In the case of static graphs an immediate extension of this work is to apply our methodology to the Sensor Selection problem, such that not only the best link candidates are selected, but also the nodes that contribute the most to the consensus-based task being executed. Similarly, it can be also considered the setting in which multiple tasks are simultaneously executed over the network, having to design a new optimization criteria to minimize the summation of the

energy functions related to the multiple tasks, which are executed by subsets of nodes. For this purpose, packet aggregation and node clustering can be applied in order to reduce the energy consumption of the overall process.

Finally, Graph Signal Processing has recently attracted a great amount of research work as a promising technique to extract information from a static signal on a static, weighted, undirected graph. An interesting extension to our work is to study how to apply our methodologies to the case of Graph Signal Processing in order to make this methodology fast and energy efficient. In the case, of time-varying graphs, the theoretical framework must be derived first and then several methodologies to understand the effect of real communication would be necessary.

Bibliography

[1] Hydrobionets: Autonomous control of large-scale water treatment plants based on self-organized wireless biomem sensor and actuator networks. http://www.hydrobionets.eu.

[2] Réka Albert. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

[3] Daniel Alonso-Román, César Asensio-Marco, Eugenio Celada-Funes, and Baltasar Beferull-Lozano. Consensus based distributed estimation of biomass concentration in reverse osmosis membranes. In *Proceedings of the 1st ACM International Workshop on Cyber-Physical Systems for Smart Water Networks*, pages 1–5, April 2015.

[4] Daniel Alonso-Román and Baltasar Beferull-Lozano. Adaptive consensus based distributed detection in wsn with unreliable links. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4438–4442, March 2016.

[5] Daniel Alonso-Román and Baltasar Beferull-Lozano. Adaptive consensus based distributed kalman filter for wsns with random link failures. In *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 187–192, May 2016.

[6] Daniel Alonso-Román, Fernando Camaró-Nogués, César Asensio-Marco, and Baltasar Beferull-Lozano. Distributed subspace projection over wireless sensor networks with unreliable links. In *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 345–349, June 2013.

[7] Daniel Alonso-Román, Eugenio Celada-Funes, César Asensio-Marco, and Baltasar Beferull-Lozano. Improving reliability and efficiency of communications in wsns under high traffic demand. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 268–273, July 2013.

[8] César Asensio-Marco, Daniel Alonso-Román, and Baltasar Beferull-Lozano. Achieving energy-efficient distributed consensus in wireless scale free networks. In *2014 IEEE International Conference on Communications (ICC)*, pages 2436–2442, August 2014.

[9] César Asensio-Marco, Daniel Alonso-Roman, and Baltasar Beferull-Lozano. Ensuring high performance of consensus based estimation by lifetime maximization in wsns. In *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 108–114, June 2015.

[10] César Asensio-Marco, Daniel Alonso-Román, and Baltasar Beferull-Lozano. Cross-layer mac protocol for unbiased average consensus under random interference. *submitted to IEEE/ACM Transactions on Networking*, August 2016.

[11] César Asensio-Marco, Daniel Alonso-Roman, Fernando Camaro-Nogues, and Baltasar Beferull-Lozano. Topology optimization for a trade off between energy cost and network lifetime in average consensus. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4623–4627, May 2013.

[12] César Asensio-Marco and Baltasar Beferull-Lozano. Accelerating consensus gossip algorithms: Sparsifying networks can be good for you. In *2010 IEEE International Conference on Communications (ICC)*, pages 1–5, May 2010.

[13] César Asensio-Marco and Baltasar Beferull-Lozano. Finding sparse connectivity patterns in power-constrained ad-hoc networks for accelerating consensus algorithms. In *18th European Signal Processing Conference (EUSIPCO)*, pages 179–183, Aug 2010.

[14] César Asensio-Marco and Baltasar Beferull-Lozano. A greedy perturbation approach to accelerating consensus algorithms and reducing its power consumption. In *2011 IEEE Statistical Signal Processing Workshop (SSP)*, pages 365–368, June 2011.

[15] César Asensio-Marco and Baltasar Beferull-Lozano. Asymmetric average consensus under sinr-based interference. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 2571–2575, Aug 2012.

[16] César Asensio-Marco and Baltasar Beferull-Lozano. Link scheduling in sensor networks for asymmetric average consensus. In *Signal Processing Advances in Wireless Communications (SPAWC), 2012 IEEE 13th International Workshop on*, pages 319–323, June 2012.

[17] César Asensio-Marco and Baltasar Beferull-Lozano. Network topology optimization for accelerating consensus algorithms under power constraints. In *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 224–229, May 2012.

[18] César Asensio-Marco and Baltasar Beferull-Lozano. Fast average gossiping under asymmetric links in wsns. In *2014 Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*, pages 131–135, Sept 2014.

[19] César Asensio-Marco and Baltasar Beferull-Lozano. Optimal topology design for energy efficient consensus in broadcast wireless sensor networks. In *IEEE International Conference on Communications (ICC)*, pages 269–275, June 2014.

[20] César Asensio-Marco and Baltasar Beferull-Lozano. Energy efficient consensus over complex networks. *IEEE Journal of Selected Topics in Signal Processing*, 9(2):292–303, March 2015.

[21] César Asensio-Marco and Baltasar Beferull-Lozano. Optimal topology design for energy efficient consensus in broadcast wireless sensor networks. *submitted to IEEE Transactions on Wireless Communications*, September 2016.

[22] Tuncer Can Aysal, Mehmet Ercan Yildiz, Anand D Sarwate, and Anna Scaglione. Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal Processing,*, 57(7):2748–2761, 2009.

[23] Leonardo Badia, Alessandro Erta, Luciano Lenzini, and Michele Zorzi. A general interference-aware framework for joint routing and link scheduling in wireless mesh networks. *IEEE network*, 22(1):32–38, 2008.

[24] Sergio Barbarossa, Gesualdo Scutari, and Ananthram Swami. Achieving consensus in self-organizing wireless sensor networks: The impact of network topology on energy consumption. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 841–847, April 2007.

[25] Florence Bénézit, Vincent Blonde, Patrick Thiran, John Tsitsiklis, and Martin Vetterli. Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In *IEEE International Symposium on Information theory proceedings*, pages 1753–1757, 2010.

[26] Florence Bénézit, Alexandros Dimakis, Patrick Thiran, and Martin Vetterli. Gossip along the way: Order-optimal consensus through randomized path averaging. In *Allerton*, 2007.

[27] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.

[28] Douglas M Blough, Giovanni Resta, and Paolo Santi. Approximation algorithms for wireless link scheduling with sinr-based interference. *IEEE/ACM Transactions on networking*, 18(6):1701–1712, 2010.

[29] Francesca Boem, Yuzhe Xu, Carlo Fischione, and Thomas Parisini. Distributed fault detection using sensor networks and pareto estimation. In *2013 European Control Conference (ECC)*, pages 932–937. IEEE, 2013.

[30] Andrej Bogdanov, Elitza Maneva, and Samantha Riesenfeld. Power-aware base station positioning for sensor networks. In *Twenty-third*

AnnualJoint Conference of the IEEE Computer and Communications Societies (INFOCOM), March 2004.

[31] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking*, 14(SI):2508–2530, 2006.

[32] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[33] Kai Cai and Hideaki Ishii. Quantized consensus and averaging on gossip digraphs. *IEEE Transactions on Automatic Control*, 56(9):2087–2100, 2011.

[34] Fernando Camaro-Nogues, Daniel Alonso-Roman, César Asensio-Marco, and Baltasar Beferull-Lozano. Reducing the observation error in a wsn through a consensus based subspace projection. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 3643–3648, April 2013.

[35] Mihaela Cardei, My T Thai, Yingshu Li, and Weili Wu. Energy-efficient target coverage in wireless sensor networks. In *24th annual joint conference of the ieee computer and communications societies.*, volume 3, pages 1976–1984, March 2005.

[36] Ruggero Carli, Alessandro Chiuso, Luca Schenato, and Sandro Zampieri. Distributed kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26(4):622–633, 2008.

[37] Eugenio Celada-Funes, Daniel Alonso-Román, César Asensio-Marco, and Baltasar Beferull-Lozano. A reliable csma protocol for high performance broadcast communications in a wsn. In *IEEE Global Communications Conference (GLOBECOM)*, pages 473–479, December 2014.

[38] Jae-Hwan Chang and Leandros Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 12(4):609–619, 2004.

[39] Chi-Kin Chau, Minghua Chen, and Soung Chang Liew. Capacity of large-scale csma wireless networks. In *Proceedings of the 15th annual international conference on Mobile computing and networking (MOBICOM)*, pages 97–108, September 2009.

[40] Rene L Cruz and Arvind V Santhanam. Optimal routing, link scheduling and power control in multihop wireless networks. In *Twenty-second annual joint conference of the IEEE computer and communications (INFOCOM)*, volume 1, pages 702–711, April 2003.

[41] Alexandros DG Dimakis, Anand D Sarwate, and Martin J Wainwright. Geographic gossip: Efficient averaging for sensor networks. *IEEE Transactions on Signal Processing*, 56(3):1205–1216, 2008.

[42] Werner Dinkelbach. On nonlinear fractional programming. *Management Science*, 13(7):492–498, 1967.

[43] Prabal Dutta, Mike Grimmer, Anish Arora, Steven Bibyk, and David Culler. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 70. IEEE Press, 2005.

[44] Tamer ElBatt and Anthony Ephremides. Joint scheduling and power control for wireless ad hoc networks. *IEEE transactions on wireless communications*, 3(1):74–85, 2004.

[45] Fabio Fagnani and Sandro Zampieri. Randomized consensus algorithms over large scale networks. *IEEE Journal on Selected Areas in Communications*, 26(4):634–649, 2008.

[46] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[47] Liqun Fu, Soung Chang Liew, and Jianwei Huang. Effective carrier sensing in csma networks under cumulative interference. *IEEE Transactions on Mobile Computing*, 12(4):748–760, 2013.

[48] Chris Godsil and Gordon F Royle. *Algebraic graph theory*, volume 207. Springer Science & Business Media, 2013.

[49] Olga Goussevskaia, Yvonne Anne Oswald, and Rogert Wattenhofer. Complexity in geometric sinr. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, pages 100–109, September 2007.

[50] Piyush Gupta and Panganmala R Kumar. The capacity of wireless networks. *IEEE Transactions on information theory*, 46(2):388–404, 2000.

[51] Bruce Hajek and Galen Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34(5):910–917, 1988.

[52] Gustavo Hernandez-Penaloza, César Asensio-Marco, and Baltasar Beferull-Lozano. Distributed estimation of statistical correlation measures for spatial inference in wsns. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 699–703, August 2012.

[53] Franck Iutzeler, Philippe Ciblat, Walid Hachem, and Jérémie Jakubowicz. New broadcast based distributed averaging algorithm over wireless sensor networks. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3117–3120. IEEE, 2012.

[54] Libin Jiang and Jean Walrand. A distributed csma algorithm for throughput and utility maximization in wireless networks. *IEEE/ACM Transactions on Networking*, 18(3):960–972, 2010.

[55] Soummya Kar and Jose MF Moura. Consensus+ innovations distributed inference over networks: cooperation and sensing in networked systems. *IEEE Signal Processing Magazine*, 30(3):99–109, 2013.

[56] Steven M Kay. Fundamentals of statistical signal processing, volume i: estimation theory. 1993.

[57] Steven M Kay. Fundamentals of statistical signal processing: Detection theory, vol. 2, 1998.

[58] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003.*, pages 482–491. IEEE, 2003.

[59] Javad Lavaei and Richard M Murray. Quantized consensus by means of gossip algorithm. *IEEE Transactions on Automatic Control*, 57(1):19–32, 2012.

[60] Lun Li, David Alderson, John C Doyle, and Walter Willinger. Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Mathematics*, 2(4):431–523, 2005.

[61] Kay Soon Low, Win Nu Nu Win, and Meng Joo Er. Wireless sensor networks for industrial environments. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 2, pages 271–276. IEEE, 2005.

[62] Jian Ni and R Srikant. Distributed csma/ca algorithms for achieving maximum throughput in wireless networks. In *Information Theory and Applications Workshop*. Citeseer, 2009.

[63] Wlodzimierz Ogryczak and Arie Tamir. Minimizing the sum of the k largest functions in linear time. *Information Processing Letters*, 85(3):117–122, 2003.

[64] Reza Olfati-Saber. Ultrafast consensus in small-world networks. In *American Control Conference, 2005*, pages 2371–2378. IEEE, 2005.

[65] Reza Olfati-Saber, Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[66] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[67] Boris N Oreshkin, Tuncer C Aysal, and Mark J Coates. Distributed average consensus with increased convergence rate. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2285–2288. IEEE, April 2008.

[68] Pangun Park, Carlo Fischione, Alvise Bonivento, Karl H Johansson, and Alberto Sangiovanni-Vincent. Breath: an adaptive protocol for industrial control applications using wireless sensor networks. *IEEE Transactions on Mobile Computing*, 10(6):821–838, 2011.

[69] Mathew Penrose. *Random geometric graphs*, volume 5. Oxford University Press Oxford, 2003.

[70] Silvana Silva Pereira, Roberto Lopez-Valcarce, and Alba Pagès-Zamora. A diffusion-based em algorithm for distributed estimation in unreliable sensor networks. *IEEE Signal Processing Letters*, 20(6):595–598, 2013.

[71] Silvana Silva Pereira and Alba Pagès-Zamora. Mean square convergence of consensus algorithms in random wsns. *IEEE Transactions on Signal Processing*, 58(5), 2010.

[72] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys)*, pages 95–107, November 2004.

[73] Dajun Qian, Dong Zheng, Junshan Zhang, Ness B Shroff, and Changhee Joo. Distributed csma algorithms for link scheduling in multihop mimo networks under sinr model. *IEEE/ACM Transactions on Networking*, 21(3):746–759, 2013.

[74] Venkatesh Rajendran, Katia Obraczka, and Jose Joaquin Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12(1):63–78, 2006.

[75] Ram Ramanathan and Regina Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings (INFOCOM)*, volume 2, pages 404–413, March 2000.

[76] Stefania Sardellitti, Sergio Barbarossa, and Ananthram Swami. Optimal topology control and power allocation for minimum energy consump-

tion in consensus networks. *IEEE Transactions on Signal Processing*, 60(1):383–399, 2012.

[77] Stefania Sardellitti, Massimiliano Giona, and Sergio Barbarossa. Fast distributed average consensus algorithms based on advection-diffusion processes. *IEEE Transactions on Signal Processing*, 58(2):826–842, 2010.

[78] Alberto Speranzon, Carlo Fischione, and Karl Henrik Johansson. Distributed and collaborative estimation over wireless sensor networks. In *Proceedings of the 45th IEEE Conference on Decision and Control (CDC)*, pages 1025–1030, December 2006.

[79] Alberto Speranzon, Carlo Fischione, and Karl Henrik Johansson. A distributed estimation algorithm for tracking over wireless sensor networks. In *IEEE International Conference on Communications (ICC)*, pages 3088–3093, June 2007.

[80] Srdjan S Stanković, Nemanja Ilić, Miloš S Stanković, and Karl Henrik Johansson. Distributed change detection based on a consensus algorithm. *IEEE Transactions on Signal Processing*, 59(12):5686–5697, 2011.

[81] Srdjan S Stankovic, Miloš S Stankovic, and Dušan M Stipanovic. Decentralized parameter estimation by consensus based stochastic approximation. *IEEE Transactions on Automatic Control*, 56(3):531–543, 2011.

[82] Alireza Tahbaz-Salehi and Ali Jadbabaie. A necessary and sufficient condition for consensus over random networks. *IEEE Transactions on Automatic Control*, 53(3):791–795, 2008.

[83] Tijs Van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys)*, pages 171–180, November 2003.

[84] Sundaram Vanka, Vijay Gupta, and Martin Haenggi. Power-delay analysis of consensus algorithms on wireless networks with interference.

International Journal of Systems, Control and Communications, 2(1-3):256–274, 2010.

[85] Sundaram Vanka, Martin Haenggi, and Vijay Gupta. Convergence speed of the consensus algorithm with interference and sparse long-range connectivity. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):855–865, 2011.

[86] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393(6684):440–442, 06 1998.

[87] Feng Xiao and Long Wang. Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays. *IEEE Transactions on Automatic Control,*, 53(8):1804–1816, 2008.

[88] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[89] Lin Xiao, Stephen Boyd, and Sanjay Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Fourth International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 63–70, April 2005.

[90] Shuguang Xiong, Jianzhong Li, and Lei Yu. Maximize the lifetime of a data-gathering wireless sensor network. In *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2009.*, pages 1–9. IEEE, 2009.

[91] Ramon Xulvi-Brunet and Igor M Sokolov. Evolving networks with disadvantaged long-range connections. *Physical Review E*, 66(2):026118, 2002.

[92] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1567–1576, April 2002.

[93] Peng Zeng, Chuanzhi Zang, and Haibin Yu. Bounding the lifetime of target tracking sensor networks. In *IEEE International Conference on Communications (ICC)*, volume 8, pages 3444–3449, June 2006.

[94] Haotian Zhang, José MF Moura, and Bruce Krogh. Dynamic field estimation using wireless sensor networks: Tradeoffs between estimation error and communication cost. *IEEE Transactions on Signal Processing*, 57(6):2383–2395, 2009.

[95] Ya Zhang and Yu-Ping Tian. Consensus of data-sampled multi-agent systems with random communication delay and packet loss. *IEEE Transactions on Automatic Control*, 55(4):939–943, 2010.