KTH Electrical Engineering

Consensus Algorithms in Dynamical Network Systems

HÅKAN TERELIUS

Licentiate Thesis
Stockholm, Sweden 2013

KTH School of Electrical Engineering
TRITA-EE 2013:043 Automatic Control
ISSN 1653-5146 SE-100 44 Stockholm
ISBN 978-91-7501-935-2 SWEDEN

Akademisk avhandling som med tillstånd av Kungl. Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen i elektro- och systemteknik fredagen den 29 november 2013, klockan 10.15 i sal L1, Kungliga Tekniska högskolan, Drottning Kristinasväg 30, Stockholm.

Tryck: Universitetsservice US AB

Abstract

Dynamical network systems are complex interconnected systems describing many real world problems. The current trend is to connect more and more systems together, and at the same time requiring continuous availability. To this end, it is crucial to understand the dynamic behaviors of networked systems. This thesis makes three contributions in this area.

First, we study the important problem of gathering data that are distributed among the nodes in a network. Two specific tasks are considered: to estimate the size of the network, and to aggregate the distribution of local measurements generated by the nodes. We consider a framework where the nodes require anonymity, and restricted computational resources. We propose probabilistic algorithms with low resource requirements, that quickly generate arbitrarily accurate estimates. For dynamical networks, we improve the accuracy through a regularization term which captures the trade-off between the gathered data and a-priori assumptions on the dynamics.

In the second part of this thesis, we consider a dynamical network system where one node is misbehaving due to a failure. We specifically seek robustness conditions that guarantee that the entire network system is still functional. The nodes' dynamics is governed by consensus updates, and we present thresholds on the interaction strengths that determines if the system will reach consensus, or if the system will diverge.

Finally, a peer-to-peer network is utilized to improve a live-streaming media application. In particular, we study how an overlay network, constructed from simple preference functions, can be used to build efficient topologies that reduce both network latency and interruptions. We present necessary and sufficient convergence conditions, as well as convergence speed estimates, and demonstrate the improvements for a real peer-to-peer video streaming application.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Karl Henrik Johansson, for your continuous support, always being cheerful, while providing me with enthusiasm and guidance.

I would like to thank my co-supervisor Ather Gattami, whom I worked closely with during the first years. Thank you for all the insight you provided me with, and our great discussions.

To my current co-supervisor Alexandre Proutiere, for your excitement, inspiration and great knowledge.

Damiano Varagnolo, for hosting my visit in Padova, our successful collaborations, and your friendship. Guodong Shi, for our interesting collaborations and your technical advise.

To my friends and colleagues at the automatic control department, both current and former. It has been a great pleasure to work and spend time together with you: Assad Alam, Behdad Aminian, Martin Andreasson, Mariette Annergren, José Araujo, Bart Besselink, Niclas Blomberg, Themistoklis Charalambous, Phoebus Chen, Burak Demirel, Piergiuseppe Di Marco, Niklas Everitt, Farhad Farokhi, Hamid Feyzmahdavian, Oscar Flärdh, Euhanna Ghadimi, António Gonga, Meng Guo, Erik Henriksson, Aitor Hernandez, Per Hägg, Martin Jakobsson, Aziz Khakulov, Omid Khorsand, Jeffrey Larson, Christian Larsson, Kuo-Yun Liang, Magnus Lindhé, Pangun Park, Chithrupa Ramesh, Alireza Seyed Ahmadi, Iman Shames, Christopher Sturk, Sadegh Talebi, André Teixeira, Ubaldo Tiberi, Olle Trollberg, Valerio Turri, Patricio Valenzuela, Yuzhe Xu and Zhenhua Zou.

A special thanks to the administrators Hanna, Kristina, Anneli and Karin for always being helpful, and creating a great office atmosphere.

I am very grateful to the Swedish Research Council, the Knut and Alice Wallenberg Foundation and the KTH School of Electrical Engineering, through the Program of Excellence, for their financial support of my studies.

Finally, I would like to thank my friends and family for your encouragement and patience through this adventure. In particular, I would like to express my gratitude to my parents and my brother for all their support throughout my life.

Thank you!

Håkan Terelius
Stockholm, November 2013.

Contents

Contents vi

Notations ix

Abbreviations xi

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Preliminaries . . . 7
- 1.3 Problem Formulation . . . 11
- 1.4 Related Work . . . 11
- 1.5 Thesis Outline and Contributions . . . 14

2 Estimation in Anonymous Networks 17
- 2.1 Size Estimation Problem . . . 18
- 2.2 Network Size Estimation Algorithm . . . 19
- 2.3 Properties under a Markovian P2P Model . . . 23
- 2.4 Evaluation of the Size Estimation Algorithm . . . 26
- 2.5 Estimation of Probability Mass Functions Problem . . . 30
- 2.6 PMF Estimators Based on Consensus Protocols . . . 31
- 2.7 Statistical Characterization of the Max Consensus PMF Estimator . 35
- 2.8 Evaluation of PMF Estimators . . . 38
- 2.9 Estimating Time Dependent PMFs . . . 44
- 2.10 Conclusions . . . 46

3 Faulty Nodes in Consensus Protocols 47
- 3.1 Problem Statement . . . 47
- 3.2 Convergence Results . . . 49
- 3.3 Illustration of Convergence Results . . . 61
- 3.4 Conclusions . . . 65

4 Convergence in Peer-to-Peer Networks 67
- 4.1 The Gradient Topology Problem . . . 67

4.2 Convergence Analysis 70
4.3 Convergence Simulation 75
4.4 Evaluating Live-Streaming using the Gradient Topology 77
4.5 Conclusions 84

5 Conclusions and Future Work **85**
5.1 Conclusions 85
5.2 Future Work 86

Bibliography **89**

Notations

$\doteq$	Definition
$\mathbb{N}$	Set of natural numbers, $0, 1, 2, \ldots$
$\mathbb{R}$	Set of real numbers
$\mathbb{C}$	Set of complex numbers
$\mathrm{Re}\,(z)$	Real part of complex number z
$\mathrm{Im}\,(z)$	Imaginary part of complex number z
A^{T}	Transpose of matrix A
$\lambda\,(A)$	Eigenvalues of matrix A
$\mathbb{1}$	Column vector with all elements equal to one
$\mathrm{diag}\,(v)$	Diagonal matrix, with elements from vector v
$\mathcal{G}$	Graph consisting of nodes $\mathcal{V}$ and edges $\mathcal{E}$
$\mathcal{V}$	Set of nodes (vertices or agents) for graph $\mathcal{G}$
$\mathcal{E}$	Set of edges (links or relations) for graph $\mathcal{G}$
$\mathcal{N}_i$	Set of neighbors to node i
$\mathcal{U}\,[0, 1]$	Uniform distribution over the interval $[0, 1]$
$p\,(x)$	Probability density function
$\mathbb{P}\,[A]$	Probability of event A
$\mathbb{E}\,[X]$	Expectation of random variable X
$\mathrm{var}\,(X)$	Variance of random variable X
$\mathrm{cov}\,(X, Y)$	Covariance between random variables X and Y
$\widehat{x}$	Estimate of quantity x
$\Gamma\,(n)$	Gamma function of n
$\Gamma\,(k, \theta)$	Gamma distribution with shape k and scale θ
I-$\Gamma\,(k, \theta)$	Inverse Gamma distribution with shape k and scale θ

Abbreviations

a.e.	almost everywhere
CDF	cumulative distribution function
EM	expectation-maximization
i.i.d.	independent and identically distributed
MC	Monte Carlo
ML	maximum likelihood
MSE	mean squared error
P2P	peer-to-peer
PMF	probability mass function
RMSE	root-mean-square error
SCADA	supervisory control and data acquisition
TCP/IP	transmission control protocol/Internet protocol
TCP	transmission control protocol

Chapter 1

Introduction

"The important thing is not to stop questioning."

— Albert Einstein

As the society is growing and technology makes us more interconnected, we need to handle huge systems that are distributed over large areas. The transition to these interconnected systems imposes a fundamental change in our understanding of the world around us. The main developments during the last century have been in improving our understanding of isolated subsystems, but we still miss the understanding of what happens when many small and simple subsystems are connected, and how different interconnections can yield different behaviors.

The outline of this chapter is as follows: In section 1.1 we give some motivating examples that inspired the work towards this thesis. In section 1.2 we introduce some preliminary results and necessary background material. In section 1.3 we formulate the main mathematical problems which we study in the following chapters. In section 1.4 we discuss some of the related work, while section 1.5 presents the contributions of this thesis.

1.1 Motivation

There is an abundance of illustrative examples where dynamical network systems appear around us. In the following sections, we describe some of them that relate to the results in this thesis.

Computer Networks

The Internet, i.e., the network of physical connections between computers, is a large and ever-changing network. It is undeniable that the Internet has been a great success, and much of the traffic is controlled by the transmission control protocol/Internet protocol (TCP/IP), originally introduced by Cerf and Kahn (1974). One of the main goals of the transmission control protocol (TCP) is to minimize

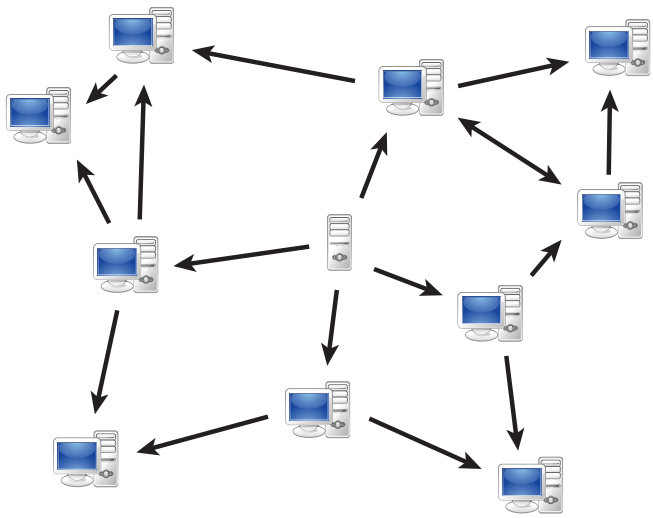


Figure 1.1: A P2P network disseminating information from a central server.

network congestion, which happens when a link or node is receiving more traffic than it can handle. The first global congestion collapse was observed in October 1986 when the backbone's throughput dropped three orders of magnitude from its normal capacity. Congestion control can be seen as a distributed optimization algorithm for solving optimal rate allocation with fair allocations.

An *overlay network* is a computer network built on top of another network, e.g., when virtual connections are built on top of the physical network, or an application network on top of the TCP network. This abstraction makes it easier to replace technological solutions, and design application protocols without full knowledge of the underlying network. However, there is still a lack of knowledge in how to design 'optimal' overlay networks.

A prominent example of an application networks is the peer-to-peer (P2P) live-streaming networks, illustrated in fig. 1.1. P2P networks are distributed network architectures, where each individual node can act as both a supplier and consumer of resources. The advantages of using P2P networks include resilience against failures, and creating scalable computer networks, where the capacity is increasing with the number of peers. The P2P networks can be divided into unstructured and structured networks, where the first class is based on nodes establishing random connections, while the structured class organizes the topology into a specific structure to ensure efficient services.

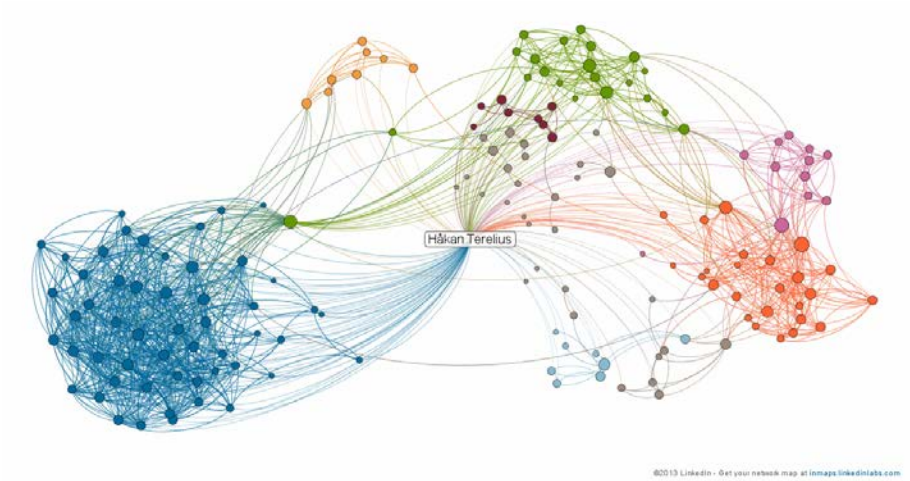


Figure 1.2: The community structure of the author's LinkedIn network. (Courtesy of LinkedIn Maps)

Social Networks

A social network is a set of people or groups of people with some relations or interactions between them. The friendships between individuals, business relationships between companies, and intermarriages between families are all examples of social networks. Social network analysis emerged as an important research topic in sociology decades ago, with the first studies focused on the adoption of medical and agricultural innovations. Most of the early work was carried out as field studies on small communities, gathering data through questionnaires, interviews, and other labor-intensive methods, e.g., the famous *small world experiment* by Travers and Milgram (1969). In recent years, social network research has adopted data collected from on-line social network platforms (e.g., Facebook, LinkedIn, mobile phone networks, etc.). It has resulted in a huge increase in the availability and in the size of social network data, and also completely redefined the types of data that can be collected and analyzed, leading to the emergence of a new computational social science. An example can be found in the paper by Lambiotte *et al.* (2008), who analyzed the network structure of mobile phone records, and could identify the French and Flemish speaking communities in Belgium. Figure 1.2 illustrates the community detection in a network.

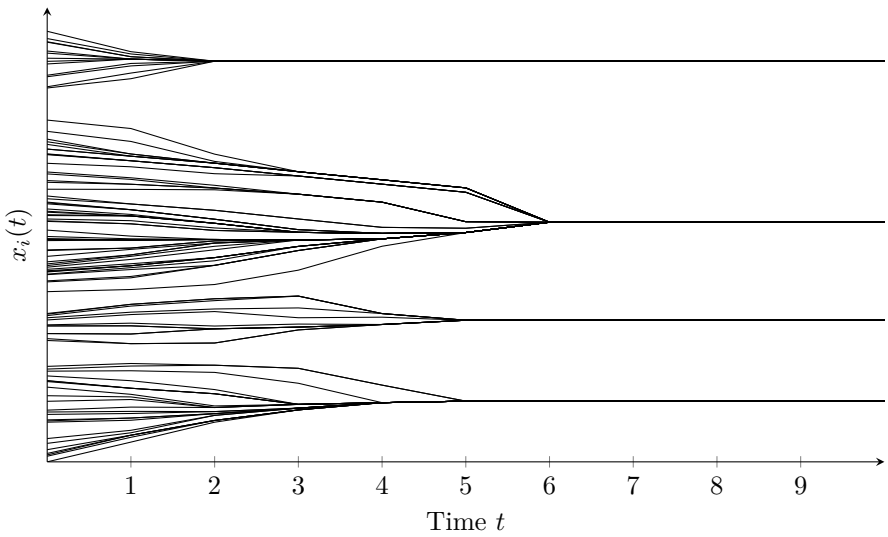


Figure 1.3: The evolution of Krause's opinion dynamics, where the initial positions where randomized with uniform probability.

There has also been considerable interest in studying opinion dynamics through social networks. One such model is the *Krause's model* by Hegselmann and Krause (2002), which is based on the two assumptions:

- people trust each other by similarity of opinions;
- people get influenced by the opinion of those they trust.

The opinion of a person is modeled as a value $x_i \in \mathbb{R}$, for example the political opinion on a left-right scale, and the options evolve according to an averaging scheme,

$$x_i(t+1) = \frac{\sum\limits_{j:[x_i(t)-x_j(t)]<1} x_j(t)}{\sum\limits_{j:[x_i(t)-x_j(t)]<1} 1}.$$

This model can explain how people form groups of similar opinions, fig. 1.3, and was studied by Blondel *et al.* (2009, 2010); Krause (2006); Nedić and Touri (2012).

However, even though online social networks can provide valuable data, there is also a great concern about preserving the users' privacy. This introduces new challenges in establishing anonymity preserving computing.

Power Networks

Power networks are large and complex dynamic networks, see fig. 1.4. The network mainly consists of power stations producing electricity, consumers of power (e.g., factories and cities), and the electric grid that transmits the electricity between the producers and consumers. The Swedish power grid can be divided into several layers, the local grid (40 kV or less), the regional grid (40-130 kV) and the national grid (220 kV and 400 kV). The national grid is also connected to other countries' power grids, adding an additional layer to the network.

The power grid is operated through supervisory control and data acquisition (SCADA) systems, a complex system where many computer controls and human operators interact. The well-known power outage *Northeast blackout 2003* in the USA affected more than 50 million people, and shows the far-reaching consequences of cascading failures in these highly interconnected systems. The primary cause of the power outage was a software bug, that prevented an alarm to reach the human operators. In light of this event, there has been a lot of interest in securing the power grid from malicious attacks, e.g., Teixeira *et al.* (2012) discussed different attack models for networked control systems. This awareness of this threat has also been fueled by the recent Stuxnet computer virus, which was targeting specific SCADA systems (Falliere *et al.*, 2011). A commonly used criteria is that the systems should be robust against any single point of failure, i.e., be able to avoid global cascading failures if any single node or link malfunctions.

Transport Networks

The transport network is another prominent example of a network system, where intersections can be modeled as nodes, and roads are the edges between the intersections. One of the main challenges in transportation networks is to accurately estimate traveling times (Jenelius and Koutsopoulos, 2013). As the transportation demand approaches the capacity of the road, traffic congestion sets in (fig. 1.5), which is often seen during rush hours. The congestion leads to both waisted commuting times and polluting effects on the environment.

Researchers cannot fully predict when traffic jams will occur, because minor incidents can have ripple effects in the entire traffic system. It is also difficult due to the hybrid nature of the traffic system, considering both continuous traffic flows and discrete vehicles.

Simply building new roads might not be the best solution. For example, Braess's paradox describes a scenario where adding extra capacity to a network can reduce the overall performance. Instead, there is a huge potential in improving the situation through better traffic management and personalized recommendations to the drivers.

This illustrate the significance of being able to estimate the current state of the network in a fast and reliable manner.

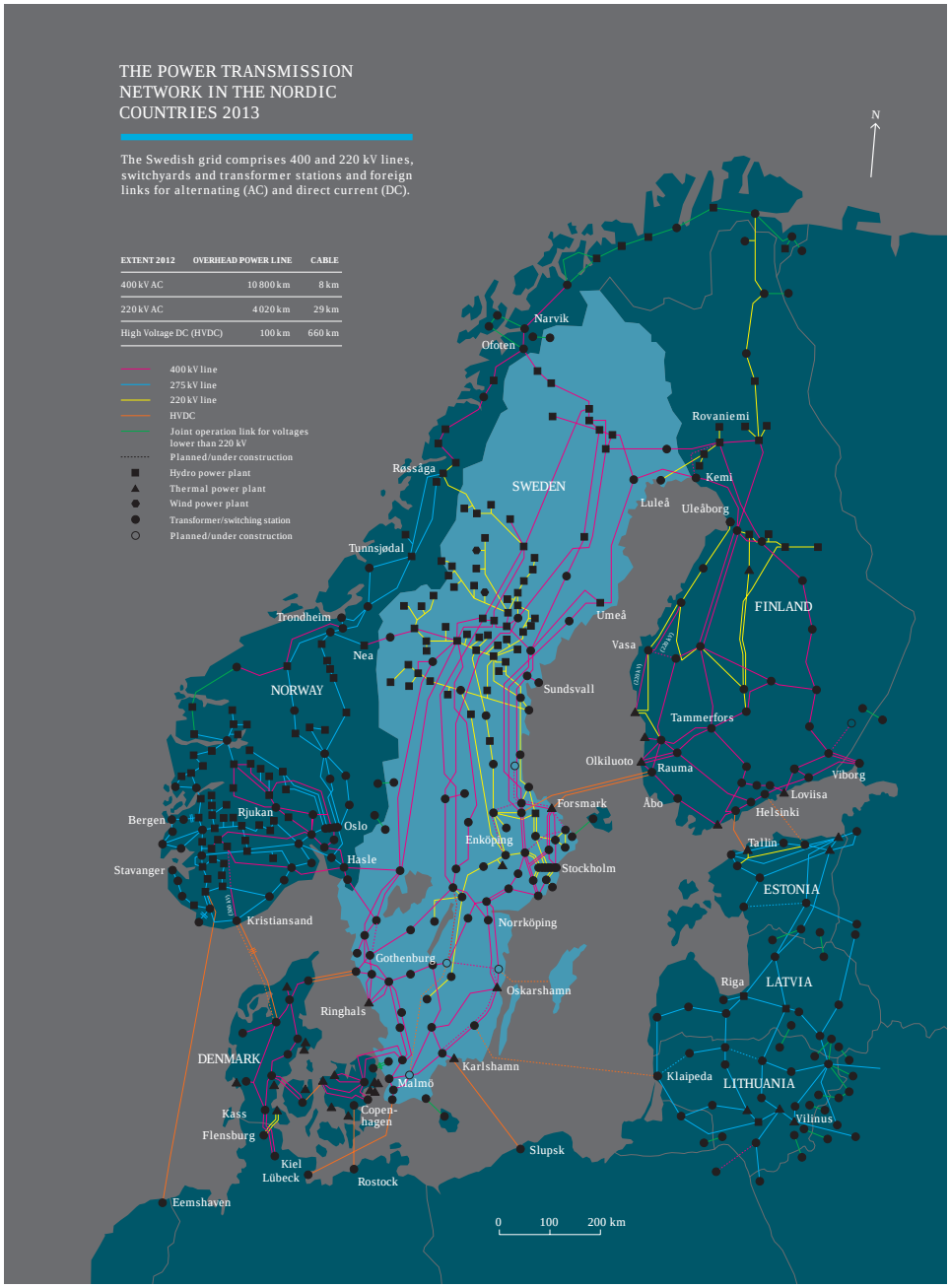


Figure 1.4: The Nordic power transmission network (Courtesy of Svenska Kraftnät)

Figure 1.5: A traffic jam in Delhi (Courtesy of Wikimedia)

1.2 Preliminaries

In this section, we introduce some mathematical preliminaries. In particular, we introduce some graph theoretical tools used to model networked systems, and also two of the basic distributed consensus algorithms.

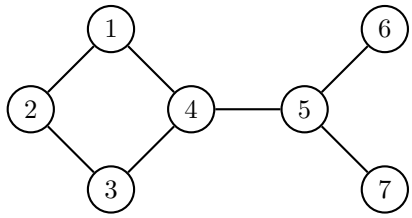
Graph Theory

Networks are commonly modeled as mathematical *graphs*, models of pairwise relations between objects. In this section, we introduce some notation and basic concepts from graph theory that will be useful for describing network systems. A more complete description of this topic can be found in the book by Diestel (2005).

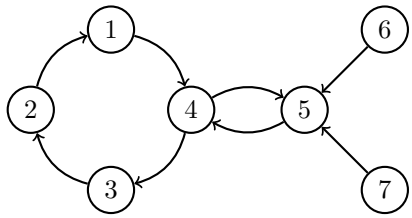
A *graph* $\mathcal{G}(\mathcal{V}, \mathcal{E})$ consists of a set of objects, called *nodes*, *vertices*, or in some of our applications *agents*, denoted by $\mathcal{V}$, and a set of pairwise relations, called *edges* or *links*, denoted by $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We usually denote the number of nodes in a graph $\mathcal{G}$ by $N_{\mathcal{G}} = |\mathcal{V}|$, or simply N if the graph can be understood from the context. Further, we usually label the nodes with numbers from 1 to N, thus $\mathcal{V} = \{1, \dots, N\}$.

We can divide the graphs into two categories, the *undirected graphs* are the graphs where the edges are unordered pairs of vertices, hence (i, j) and (j, i) are considered to be the same edge, and $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$. All other graphs are *directed graphs*, and the edge (i, j) is directed from node i to node j.

The usual way of picturing an undirected graph is by drawing a circle for each



(a) An undirected graph with 7 nodes and 7 edges.



(b) A directed graph with 7 nodes and 8 edges.

Figure 1.6: Graphs are illustrated with circles representing nodes, and lines representing edges.

node, and joining two of the circles by a line if the corresponding nodes form an edge, fig. 1.6a.

Similarly, a directed graph is pictured by drawing a circle for each node, and joining two circles with an arrow pointing from i to j if (i, j) forms an edge of the graph, fig. 1.6b.

Two nodes that have an edge between them are said to be *adjacent*. The *neighbors* of a node i are those nodes which have an incoming edge from node i. We denote the neighbors of node i by $\mathcal{N}_i$, and the *degree* of a node is the size of its neighborhood $\deg(i) = |\mathcal{N}_i|$, where

$$\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}\}.$$

A *path* in $\mathcal{G}$ is a list of distinct vertices $\{v_0, v_1, \ldots, v_n\}$ such that $(v_i, v_{i+1}) \in \mathcal{E}$, $i = 0, 1, \ldots, n-1$. The number of edges in the path is the *length* of the path. The path is said to go from v_0 to v_n.

A directed graph is *strongly connected* if there exists a path from every vertex to every other vertex in the graph, while for an undirected graph, it is simply called *connected*. The *distance* $\operatorname{dist}(v_0, v_n)$ between two nodes v_0 and v_n, is the length of the shortest path between them, or ∞ if there is no such path. Further, the *diameter* of a graph is the greatest distance between any two nodes in the graph.

Spectral Graph Theory

An important way to study graphs is through representing them with matrices and analyzing matrix properties, such as eigenvalues.

The *adjacency matrix* A of a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is an $N \times N$ matrix, whose entries a_{ij} are given by the edges as:

$$a_{ij} = \begin{cases} 1 & \text{if } i \neq j \text{ and } (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

The next important matrix is the *Laplacian matrix* L of a graph, defined as $L = D - A$, where D is the diagonal degree matrix, and A is the adjacency matrix:

$$l_{ij} = \begin{cases} \deg(i) & \text{if } i = j, \\ -1 & \text{if } i \neq j \text{ and } (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

The eigenvalues of the Laplacian matrix are important for many network applications, including robustness and convergence time. Also, the multiplicity of 0 as an eigenvalue is equal to the number of connected components.

For example, the adjacency matrix and Laplacian matrix of the undirected graph in fig. 1.6a are

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad L = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

Consensus Algorithms

The consensus problem, or agreement problem, is a distributed computing problem with the goal of reaching agreement on a final value. Typically in a setting with failing nodes, and limited computational and communication resources. Two specific algorithms are considered: to reach consensus on the average and the maximal initial value respectively.

Both of these algorithms assumes a strongly connected network $\mathcal{G}$ of nodes $\mathcal{V}$, where each node $i \in \mathcal{V}$ is given an initial real value $x_i(0) \in \mathbb{R}$. Each node then updates its own value based only on the values of its neighbors $\mathcal{N}_i$, without any influence from the rest of the network.

Average Consensus

The average consensus algorithm is based on linear iterations over the neighbors. The goal is to let every node's value converge to the average of the initial values,

$$x_i(t) \xrightarrow{t \to \infty} \frac{1}{N} \sum_{j \in \mathcal{V}} x_j(0), \quad \forall i \in \mathcal{V}. \tag{1.1}$$

Each node follows the update rule

$$x_i(t+1) = w_{ii}x_i(t) + \sum_{j\in\mathcal{N}_i} w_{ij}x_j(t). \tag{1.2}$$

If W is the weight matrix with entries w_{ij} (or 0 if w_{ij} is not present in the updates above), then, in vector form, where $x = \left[x_1, x_2, \cdots, x_N\right]^{\mathrm{T}}$, the entire network update can be condensed into

$$x(t+1) = Wx(t). \tag{1.3}$$

This system will converge to the average values, for an arbitrary initial condition, if and only if the weight matrix satisfies

$$\lim_{t\to\infty} W^t = \frac{1}{N}\mathbb{1}\mathbb{1}^{\mathrm{T}}$$

An equivalent condition is that W is a double stochastic matrix, and that $\rho(W - (1/N)\mathbb{1}\mathbb{1}^{\mathrm{T}}) < 1$ (Xiao and Boyd, 2003).

Notice that the convergence is asymptotic in the number of iterations.

Max Consensus

Here, the goal is to let every node's value converge to the maximum of the initial values,

$$x_i(t) \xrightarrow{t\to\infty} \max_{j\in\mathcal{V}} x_j(0), \quad \forall i \in \mathcal{V}. \tag{1.4}$$

The local update consists of simply taking the maximum of the neighbors' values,

$$x_i(t+1) = \max_{j\in\mathcal{N}_i} x_j(t). \tag{1.5}$$

This protocol can be implemented with either gossip or broadcast communications. In the latter case, agents sequentially broadcast their local values, and whoever receives this information updates its local value with the maximum. Under mild assumptions on the communication process, the max consensus protocols are proven to converge to the true maximum in a finite amount of time, bounded by the diameter of the network (Iutzeler *et al.*, 2012).

1.3 Problem Formulation

The aim of this thesis is to provide an analytical framework to study dynamical network systems. In particular, we focus on estimation and convergence properties for algorithms that are based on the consensus protocols.

The first problem we study is to estimate properties in a dynamical network, specifically the network size $N(t) = |\mathcal{V}(t)|$. This is an important tool for network reconfiguration and fault detection in networks. We assume that there exists a-priori knowledge on the expected evolution of the network size $N(t)$, which can be used to improve the estimate. The methods should be distributed, where all nodes execute the same algorithms in parallel, and where all agents should reach consensus on the estimate as fast as possible. We want to have a framework of anonymous agents, where the there is no guarantee that the nodes have global identities.

The second problem we consider is a continuous time consensus system,

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(t)\big(x_j(t) - x_i(t)\big),$$

but where one agent k has an error that causes it to use positive feedback instead of negative: $a_{kj} \to -a_{kj}$. The aim is to study robustness properties against this type of faults, and determine thresholds on the interaction weights such that the system still converges to consensus.

In the final problem, we consider topologies in P2P live-streaming applications. The goal is to reduce the latency and interruptions in the video stream through designing an overlay networks that aids in network reconfigurations. The gradient overlay topology should be built using a local preference function that selects peers from a random peer sampling service, and convergence properties for this system are needed.

1.4 Related Work

In this section, we summarize some of the important contributions in the field of dynamical network systems, and in particular related to the work in this thesis.

Estimation in Networks

The importance of distributed estimation is reflected by the variety of applications where agents interact and cooperate to reach a common goal. Examples of these systems include environmental monitoring (Lynch *et al.*, 2008), management of the electrical grid (Bolognani and Zampieri, 2013) and the public transportation system (Herring *et al.*, 2009).

A common approach to network size estimation is to use *random walks* (Gkantsidis *et al.*, 2006; Massoulié *et al.*, 2006; Ribeiro and Towsley, 2010), relying on a token being passed around the network to collect information each time it visits

an agent. Another strategy is to use randomly generated numbers (Kostoulas *et al.*, 2007), and then exploit classical results on order statistics to infer the number of participants (Baquero *et al.*, 2012; Cardoso *et al.*, 2009; Chassaing and Gerin, 2006; Giroire, 2009; Lumbroso, 2010; Varagnolo *et al.*, 2013). These probabilistic techniques have been statistically analyzed (Cichon *et al.*, 2012b; Clifford and Cosma, 2012), and are extensions of methodologies for estimating sums over networks (Cohen, 1997; Mosk-Aoyama and Shah, 2008). Other network size estimation schemes use the *capture-recapture* concept (Peng *et al.*, 2009; Petrovic and Brown, 2009), where seed numbers are randomly disseminated through the network, and then, by counting how many seeds are in a given subset, inferring the size of the network. Some studies (Cichon *et al.*, 2012a, 2011) exploit probabilistic counting algorithms (Flajolet *et al.*, 2007) usually implemented in non-distributed contexts. Other techniques take advantage of their specific framework and are not implementable in general settings (Ali *et al.*, 2009; Dolev *et al.*, 2006; Howlader *et al.*, 2008; Leshem and Tong, 2005).

The previous studies mainly dealt with static networks, but there are also some extensions to dynamical settings. Fusy and Giroire (2007) used order statistics, Psaltoulis *et al.* (2004) considered random walks, Chabchoub and Hébrail (2010) exploited probabilistic counting algorithms and Alouf *et al.* (2004, 2002); Shafaat *et al.* (2008) also dealt with dynamic scenarios.

When it comes to estimating probability mass functions over networks, the literature can be divided into parametric and non-parametric approaches. Parametric approaches assume the estimand to have a certain structure before obtaining observations, e.g., to be a sum of Gaussian distributions. Distributed implementations of the expectation-maximization (EM) algorithm (Forero *et al.*, 2008; Jiang and Jin, 2006; Nowak, 2003; Vlassis *et al.*, 2005) are examples of a parametric approach. Non-parametric approaches, on the other hand, do not assume a fixed structure a-priori, but rather select it from the observations. Kernel density estimation (Hu *et al.*, 2007), classification (Klusch *et al.*, 2003) and clustering approaches (Nguyen *et al.*, 2005) are all examples of non-parametric approaches.

The literature can also be characterized by how information is propagated and aggregated over the network. There are strategies based on pre-established hierarchic tree routing structures, where the nodes compute the distributions in their sub-trees and propagate the information towards the root (Greenwald and Khanna, 2004; Madden *et al.*, 2002; Motegi *et al.*, 2006; Shrivastava *et al.*, 2004). Borges *et al.* (2012); Haridasan and van Renesse (2008); Sacha *et al.* (2010) all used gossip communications, and exploit averaging techniques to explicitly compute the cumulative distribution functions, while Cheng *et al.* (2010); Massoulié *et al.* (2006) estimated how many agents are in a specific state.

Multi-Agent Control Systems

Co-operative control of multi-agent systems has been extensively investigated in the literature for consensus, formation, flocking, aggregation and coverage of a group

of autonomous agents (Cortes and Bullo, 2005; Jadbabaie *et al.*, 2003; Lin *et al.*, 2005, 2007; Olfati-Saber and Murray, 2004; Olfati-Saber and Shamma, 2005; Ren and Beard, 2008; Shi and Hong, 2009; Shi *et al.*, 2012; Tanner *et al.*, 2007; Tsitsiklis *et al.*, 1986).

In *leader-follower* protocols, some agents are designated as leaders whose task it is to guide the remaining follower agents. The controllability of leader-follower systems was introduced by Tanner (2004), in which necessary and sufficient conditions were established. Gustavi *et al.* (2010) studied conditions for maintaining connectivity in leader-follower networks, with single integrator action. The graph-theoretic characterizations were further studied by Ji *et al.* (2006); Rahmani *et al.* (2009); Rahmani and Mesbahi (2006).

Hong *et al.* (2006) studied tracking control for multi-agent consensus with one single active leader and a neighbor-based observer. Gu and Wang (2009) discussed leader-follower flocking, where only a few agents have the knowledge of a desired trajectory. The leader-to-formation stability was studied for formation control of multi-agent systems by Olfati-Saber and Murray (2004). There has also been work with multiple leaders (Couzin *et al.*, 2005; Ji *et al.*, 2008; Lin *et al.*, 2005). Cao and Ren (2009) studied distributed control protocols to drive a collection of mobile agents to stationary state, with connectivity maintenance and collision avoidance. Meng *et al.* (2012) discussed swarm tracking problems for a group of Lagrange systems. Shi and Hong (2009) studied multiple leaders aggregating the entire multi-agent system within a convex target set.

Peer-to-Peer Networks

Randomized gossiping algorithms have been recent tools for building distributed systems, in particular in the areas of overlay networks, sensor networks and cloud computing storage services (Boyd *et al.*, 2006; Kermarrec and van Steen, 2007). Convergence properties of gossip-based aggregation algorithms have been studied for both fixed topologies (Olshevsky and Tsitsiklis, 2006) and regular graphs (Jelasity *et al.*, 2007; Liu *et al.*, 2009).

Research in gossiping has also focused on using the Preferential Connectivity Model (Mihail *et al.*, 2003) to construct overlay network topologies, where nodes initially connected in a random graph use a preferential connection function to break the symmetry of the random graph, and build a topology that contains useful global information. Barabási (2002) first described how a preferential attachment function in a growing network can build a scale-free network topology from a random graph. Barabási's preferential attachment functions are based on the global state, but in overlay networks, nodes only have a relatively small partial view of the system. Thus, the preference functions can only be based on the local state and the state of the node's neighbors. Examples of existing overlay networks that construct their topologies using gossiping and preference functions include Spotify, that preferentially connects nodes with similar music play-lists (Kreitz and Niemelä, 2010), Sepidar, that preferentially connects P2P live-streaming nodes with similar upload

bandwidth capacity (Payberah *et al.*, 2010b), and T-Man, a framework that provides a generic preference function for building such overlays (Jelasity *et al.*, 2009).

1.5 Thesis Outline and Contributions

This thesis is a compilation of results presented in peer reviewed scientific venues. The remainder of this thesis is organized as follows.

Chapter 2: Estimation in Anonymous Networks

In this chapter, we consider the problem of estimating the state of a network, motivated by network maintenance. In particular, we study two problems, first estimating the size of the network, and secondly to estimate the entire empirical distributions of measurements over the network. The proposed algorithms are based on max consensus information exchange protocols, since they lead to fast convergence speeds as well as small communication burdens. What is particular in our scheme is that we assume the the agents to be anonymous, thus severely limiting the communication information.

This chapter is based on the publications

- Håkan Terelius, Damiano Varagnolo, and Karl Henrik Johansson. 2012. Distributed Size Estimation of Dynamic Anonymous Networks. In *51st IEEE Conference on Decision and Control*, pages 5221–5227, Maui, HI, USA, December 2012

- Håkan Terelius, Damiano Varagnolo, Carlos Baquero, and Karl Henrik Johansson. 2013b. Fast Distributed Estimation of Empirical Mass Functions over Anonymous Networks. In *52nd IEEE Conference on Decision and Control*, Florence, Italy. To appear

Chapter 3: Faulty Nodes in Consensus Protocols

This chapter studies consensus control for a multi-agent system with a faulty node. The node dynamics follow a normal continuous-time consensus protocol with negative feedback, where the faulty node instead uses positive feedback. In particular, we study the interaction strength parameters that can guarantee the system to reach consensus.

This chapter is based on the publication

- Håkan Terelius, Guodong Shi, and Karl Henrik Johansson. 2013a. Consensus Control for Multi-agent Systems with a Faulty Node. In *4th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 425–432, Koblenz, Germany, September 2013

Chapter 4: Convergence in Peer-to-Peer Networks

In this chapter, we investigate the topology convergence in a P2P network system. The goal of the system is to maximize live-streaming performance through establishing a gradient overlay topology. The gradient overlay network is characterized by a directed graph, where each node has a set of neighbors with the same utility value and a set of neighbors containing higher utility values, such that paths of increasing utilities emerge in the network topology. The gradient overlay network is built using gossiping and a preference function that samples from nodes using a uniform random peer sampling service. Evaluation of the gradient overlay topology in the live-streaming application *GLive* was performed by SICS.

This chapter is based on the publication

- Håkan Terelius, Guodong Shi, Jim Dowling, Amir H. Payberah, Ather Gattami, and Karl Henrik Johansson. 2011a. Converging an Overlay Network to a Gradient Topology. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 7230–7235, Orlando, FL, USA, December 2011

Chapter 5: Conclusions and Future Work

A summary of this thesis, and possible future research directions.

Other Contributions

The following publication is not covered in this thesis, but is relevant for networked systems. The paper studies a distributed multi-agent optimization problem of minimizing the sum of convex objective functions. A decentralized optimization algorithm is introduced, based on dual decomposition, together with the sub-gradient method for finding the optimal solution.

- Håkan Terelius, Ufuk Topcu, and Richard M. Murray. 2011b. Decentralized Multi-Agent Optimization via Dual Decomposition. In *18th IFAC World Congress*, pages 11245–11251, Milan, Italy, August 2011

Author's Contributions

The order of the authors reflect their contributions in the mentioned papers. The author has formulated and solved the problems, as well as written the papers. The co-authors has participated in discussions and conventional supervision.

Chapter 2

Estimation in Anonymous Networks

"Prediction is very difficult,
especially if it's about the future."
— Niels Bohr

In this chapter we consider distributed estimation of data obtained by the nodes in a network. Two specific problems are analyzed: first to estimate the size of the network $N = |\mathcal{V}|$, and later to estimate the empirical distribution of local measurements z_i generated by each node $i \in \mathcal{V}$. We explicitly target dynamical networks by utilizing a regularization term which captures a-priori assumptions on the dynamic behavior.

Our aim is to obtain distributed methods, where all nodes execute the same algorithm in parallel, and where neither leaders nor an overlay structure is present. We assume that the nodes have no knowledge of the network topology, and that they have narrowly bounded computational, memory and bandwidth resources, where especially the size of the exchanged information packets stays constant over time. Finally, the goal is that all agents should quickly reach consensus, in the sense that they should share the same estimates of the global properties for the network as fast as possible.

We further restrict our methods to anonymous networks, where the uniqueness of the nodes' IDs is not guaranteed (Yamashita and Kameda, 1988), thus avoiding the possibility of tracing or characterizing a single agent. The anonymity is motivated for maintaining users' privacy (e.g., in P2P networks where users may not want to disclose information about their identity), but is also beneficial in applications when the estimation strategies must be simple with limited resource requirements.

The outline of this chapter is as follows: In section 2.1 we introduce the network size estimation problem for dynamical networks, which we analyze in section 2.2. We continue by specifically considering quadratic regularization terms in section 2.3, and evaluate the estimator in section 2.4 with numerical experiments.

Next, we turn to the probability mass function (PMF) estimation problem for static networks in section 2.5, and introduce two different estimators for this prob-

lem in section 2.6. These estimators are analyzed in section 2.7, and evaluated with numerical simulations in section 2.8.

Finally, we combine the dynamical network size estimator from section 2.1 with the PMF estimator from section 2.6 to estimate PMFs in dynamical networks in section 2.9. In section 2.10 we summarize and conclude this chapter.

2.1 Size Estimation Problem

The first problem we consider in this anonymous network framework is to estimate the network size $N(t) = |\mathcal{V}(t)|$ of a time-dependent network $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t))$. In many distributed network applications, the network will need to redirect resources or take other restorative actions if the network topology changes. Thus, being able to estimate the size and, in particular, changes in the network size is indispensable for automatic network reconfiguration and fault detection.

Compared to the previous literature, we derive a distributed estimator that extends techniques based on order statistics with a regularization approach (Vapnik, 1998; Wahba, 1990). We introduce a regularization term that allows the designer to combine the empirical evidence from the data with a-priori believes on the expected behavior of the network size to be estimated, and then provide an analysis of the quadratic regularization functions.

Let $N(t)$ represent the true number of agents in the network at time $t \in \mathbb{N}$, while $\widehat{N}(t)$ denotes the estimated value of this quantity. In the maximum likelihood (ML) estimator, we denote a generic hypothesis by $\overline{N}(t)$ for the estimated value of $N(t)$. The estimator will simultaneously estimate the network size for a time window of length $\tau+1$, and we can utilize previous estimates up until time $t-\eta$, where $\eta \geq \tau$, in the regularization term. We therefore introduce the following vectorized versions of the previous quantities, where the bold italics indicate vectors:

$$\boldsymbol{N}(t) \doteq [N(t), \ldots, N(t-\tau)]^T \tag{2.1}$$

$$\overline{\boldsymbol{N}}(t) \doteq \left[\overline{N}(t), \ldots, \overline{N}(t-\tau)\right]^T \tag{2.2}$$

$$\widehat{\boldsymbol{N}}(t) \doteq [\widehat{N}(t), \ldots, \widehat{N}(t-\tau)]^T \tag{2.3}$$

$$\widehat{\boldsymbol{N}}_\tau^\eta(t) \doteq [\widehat{N}(t-\tau-1), \ldots, \widehat{N}(t-\eta)]^T. \tag{2.4}$$

Thus, $\boldsymbol{N}(t)$ refers to the true values over a time window of length $\tau+1$, $\overline{\boldsymbol{N}}(t)$ refers to a generic hypothesis on the true value of $\boldsymbol{N}(t)$, and $\widehat{\boldsymbol{N}}(t)$ refers to the estimate of the true values. $\widehat{\boldsymbol{N}}_\tau^\eta(t)$ contains an additional memory of previous estimates that is used to improve the regularization process of the estimate. Notice that $\tau, \eta \in \mathbb{N}$ are fixed design parameters of the algorithm.

The considered network model $\mathcal{G}(t)$ of interconnected agents $\mathcal{V}(t) = \{1, \ldots, N(t)\}$ is based on agents which can join or leave at any time. The goal is to distributively track the network size, i.e., each agent should create an estimate of the number of

agents as the network size is evolving. At the same time, each agent can only communicate with its direct neighbors, and the anonymity assumption further implies that no global unique identifiers can be exchanged for the estimation purposes.

At the communication protocol level, we consider a simplified framework, where effects of clock synchronization, packet loss and quantization issues can be neglected.

2.2 Network Size Estimation Algorithm

The basic idea behind the network size estimation scheme, introduced by Varagnolo *et al.* (2010), is that each agent $i \in \mathcal{V}(t)$ generates a uniform, random sample $\mathcal{U}[0,1]$, and then the max consensus protocol, described in section 1.2, is used to compute the maximum of these samples. This yields a sample of a random variable whose distribution is the maximum of $N(t)$ independent and identically distributed (i.i.d.) random variables, which depends upon $N(t)$. Hence, this sample can be used to compute a ML estimate of $N(t)$.

The specific extension of this idea to dynamic network size estimation includes a repetitive generation of new random samples and computation of the maximal value. The samples from the max distribution is kept for a time window of length $\tau+1$, to simultaneously estimate $\boldsymbol{N}(t)$ with the a-priori assumptions on the evolution. Our algorithm also include an additional memory of the previous estimates $\widehat{\boldsymbol{N}}_\tau^\eta(t)$ for an extended time window of length $\eta-\tau$, which are considered as fixed parameters in the estimation scheme. All this data is then used to compute a penalized ML estimate $\widehat{\boldsymbol{N}}(t)$, as described in Algorithm 2.1 and eq. (2.6).

Remark. The time index t does not need to denote physical quantities (such as seconds), but rather *epochs*, under which each iteration of algorithm 2.1 is completed. We thus implicitly assume that the agents always reach the max consensus on the locally generated samples.

The penalized log-likelihood function J in (2.5) is defined as:

$$J\Big(\overline{\boldsymbol{N}}\ ;\ \boldsymbol{f}(t),\dots,\boldsymbol{f}(t-\tau),\widehat{\boldsymbol{N}}_\tau^\eta(t)\Big) \doteq \\ -\log\left(p\left(\boldsymbol{f}(t),\dots,\boldsymbol{f}(t-\tau)\ ;\ \overline{\boldsymbol{N}}\right)\right)+\gamma\mathcal{R}\left(\overline{\boldsymbol{N}},\widehat{\boldsymbol{N}}_\tau^\eta(t)\right)\ . \quad (2.6)$$

This allows us to estimate the network size $\boldsymbol{N}(t)$ while penalizing hypotheses $\overline{\boldsymbol{N}}$ that deviate from expected behaviors by means of the regularization term $\mathcal{R}$: $\mathbb{R}^{\tau+1}\times\mathbb{R}^{\eta-\tau}\to\mathbb{R}_+$. Thus, given a hypothesis $\overline{\boldsymbol{N}}$, eq. (2.6) evaluates both its plausibility in the regularization term and its empirical evidence in the log-likelihood function (Schölkopf and Smola, 2002, Chap. 4). The parameter γ in (2.6) is called the *regularization parameter*, and can be tuned to capture the trade-off between the empirical evidence of $\overline{\boldsymbol{N}}$ and its plausibility.

Notice that the hypothesis $\overline{\boldsymbol{N}}$ correspond to a time-window of length $\tau+1$, while the regularization term $\mathcal{R}$ explicitly depends on the memory of the past estimates

Algorithm 2.1 Dynamic Network Size Estimation Algorithm

1: **for** every $t = 1, 2, \dots$ **do**

2: (Generation step) Each agent $i = 1, \dots, N(t)$ generates M i.i.d. random values

$$y_{i,m}(t) \sim \mathcal{U}[0,1], \quad m = 1, \dots, M$$

3: (Communication step) Agents compute, through max consensus strategies, the M-dimensional max vector

$$\boldsymbol{f}(t) \doteq [f_1(t), \dots, f_M(t)]^T,$$

where

$$f_m(t) = \max_{i=1,\dots,N(t)} y_{i,m}(t).$$

4: (Computation step) Each agent estimates the total number of agents in the network through the penalized ML scheme as

$$\widehat{\boldsymbol{N}}(t) = \underset{\overline{\boldsymbol{N}} \in \mathbb{R}^{\tau+1}}{\arg\min}\, J\Big(\overline{\boldsymbol{N}}\ ;\ \boldsymbol{f}(t), \dots, \boldsymbol{f}(t-\tau), \widehat{\boldsymbol{N}}_{\tau}^{\eta}(t)\Big) \tag{2.5}$$

5: **end for**

$\widehat{\boldsymbol{N}}_{\tau}^{\eta}(t)$ up to time $t-\eta$ ($\eta \geq \tau$), defined in (2.4). The past estimates $\widehat{\boldsymbol{N}}_{\tau}^{\eta}(t)$ are not changed by the estimator, and are used as fixed extra parameters. An illustrative description of how these time windows shift in time is given in fig. 2.1.

Remark. If the regularization term is removed, $\mathcal{R} = 0$, then Algorithm 2.1 is reduced to sequentially computing the estimates as

$$\begin{aligned} \widehat{N}(t) &\doteq \underset{\overline{N} \in \mathbb{R}}{\arg\min} \Big(-\log\big(p\left(\boldsymbol{f}(t)\ ;\ \overline{N}\right)\big) \Big) \\ &= -\left(\frac{1}{M} \sum_{i=1}^{M} \log\left(f_i\right) \right)^{-1} . \end{aligned} \tag{2.7}$$

In this case, the various $\widehat{N}(t)$'s are estimated independently, and this corresponds to the ML approach used in static anonymous network frameworks (Varagnolo *et al.*, 2010). The accuracy is clearly improved by increasing the number of random samples M, as can be seen from the statistical properties of $\widehat{N}$ in this case (assuming $M > 2$):

$$\mathbb{E}\left[\frac{\widehat{N}(t)}{N(t)}\ ;\ M \right] = \frac{M}{M-1}\ , \tag{2.8}$$

$$\mathbb{E}\left[\left(\frac{N(t) - \widehat{N}(t)}{N(t)} \right)^2\ ;\ M \right] = \frac{M^2 + M - 2}{(M-1)^2(M-2)}\ . \tag{2.9}$$

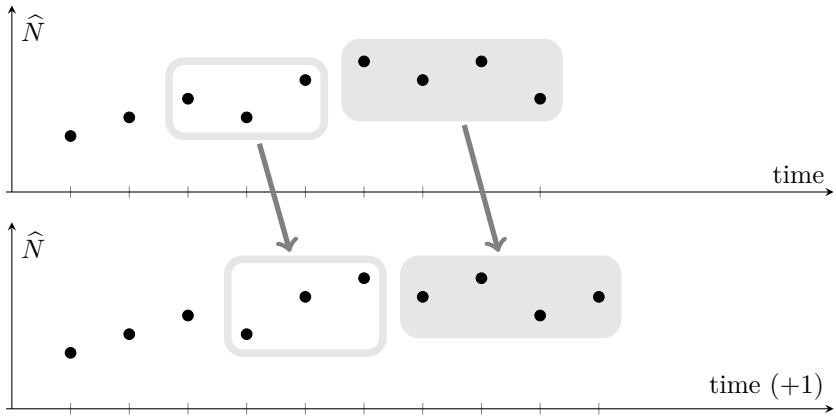


Figure 2.1: Example of the time behavior of the estimation scheme of eq. (2.6). The white rectangle indicates the extra parameters $\widehat{\boldsymbol{N}}_\tau^\eta(t)$, while the gray rectangle indicates the time-window where the optimization problem of eq. (2.5) acts to obtain new estimates.

Parameter Design Constraints

Also for the dynamical network size estimation, the estimation accuracy is intuitively non-decreasing in M, τ and η. However, M is bounded by transmission costs (in the max consensus step), τ is bounded by computational constraints (in the size of the optimization problem, eq. (2.5)), while η is bounded by memory limitations.

Notice that memory can be saved by compressing the vectors $\boldsymbol{f}(t), \boldsymbol{f}(t-1), \ldots$ into scalars without loss of information, as the following proposition states.

Proposition 2.1. *Let* $s(\tau) \doteq -\sum_{m=1}^{M} \log\left(f_m(\tau)\right)$. *Then* $s(\tau)$ *is a complete and minimal sufficient statistic for* $N(\tau)$.

By introducing $\boldsymbol{s}(t) \doteq [s(t), \ldots, s(t-\tau)]^T$, the penalized likelihood (2.6) can be rewritten as

$$J\Big(\overline{\boldsymbol{N}}\ ;\ \boldsymbol{s}(t), \widehat{\boldsymbol{N}}_\tau^\eta(t)\Big) = -\log\left(p\left(\boldsymbol{s}(t)\ ;\ \overline{\boldsymbol{N}}\right)\right) + \gamma \mathcal{R}\left(\overline{\boldsymbol{N}}, \widehat{\boldsymbol{N}}_\tau^\eta(t)\right)$$

Proof. Because the samples $y_{i,m}(\tau)$ are i.i.d., it follows that $p\left(\boldsymbol{f}(t), \ldots, \boldsymbol{f}(1)\ ;\ \overline{\boldsymbol{N}}\right) = \prod_{\tau=1}^{t} p\left(\boldsymbol{f}(\tau)\ ;\ \overline{N}(\tau)\right)$. To prove the proposition it is then sufficient to show that $s(\tau)$ is a complete and minimal sufficient statistic for $N(\tau)$.

$$p\left(\boldsymbol{f}(\tau)\ ;\ \overline{N}(\tau)\right) = \prod_{m=1}^{M} \overline{N}(\tau) \cdot f_m(\tau)^{\overline{N}(\tau)-1} = \overline{N}(\tau)^M e^{-(\overline{N}(\tau)-1)s(\tau)},$$

for all τ, thus, $s(\tau)$ is a sufficient statistic for $N(\tau)$ because of the Fisher-Neyman factorization theorem. It is also clearly minimal since it is a scalar.

To show the completeness of $s(\tau)$, we must show that if $g(s(\tau))$ is a generic measurable function such that $\mathbb{E}\left[g(s(\tau)) \mid N\right] = 0$ independently of N, then it must be $g(\cdot) \equiv 0$ almost everywhere (a.e.). Consider now that $-\log\left(f_i(\tau)\right)$ is an exponential random variable with rate N. Thus, $s(\tau)$ is the sum of i.i.d. exponential random variables, i.e., $s(\tau) \sim \Gamma\left(M, \frac{1}{N}\right)$. $\mathbb{E}\left[g(s(\tau)) \mid N\right] = 0$ can then be rewritten as

$$\Gamma\left(M\right)^{-1} N^M \int_0^{+\infty} g(s) s^{M-1} \exp\left(-sN\right) ds \equiv 0 \, .$$

This is equivalent to the fact that the Laplace transform of $g(s)s^{M-1}$ has to be zero a.e., and this happens if and only if $g(s)$ is zero a.e.. □

This compression of variables actually results in a memory saving of $M \cdot \tau$ scalars, and only a single vector of M scalars is needed for computing the current $\boldsymbol{f}(t)$ in the max consensus step.

Quadratic Regularization

Adding a regularization term $\mathcal{R}$ in empirical risk minimization problems, as we did in eq. (2.6), generally improves their conditioning properties (Schölkopf and Smola, 2002, Chap. 4). The usage of these terms can also be motivated by Bayesian perspectives, where the penalty $\mathcal{R}$ reflects a-priori beliefs on a typical behavior.

Here we explicitly consider quadratic regularization terms, i.e.,

$$\mathcal{R}\left(\overline{\boldsymbol{N}}, \widehat{\boldsymbol{N}}_\tau^\eta\right) = \begin{bmatrix} \overline{\boldsymbol{N}} - \boldsymbol{\mu}_1 \\ \widehat{\boldsymbol{N}}_\tau^\eta - \boldsymbol{\mu}_2 \end{bmatrix}^T \underbrace{\begin{bmatrix} \mathcal{Q}_{11} & \mathcal{Q}_{12} \\ \mathcal{Q}_{12}^T & \mathcal{Q}_{22} \end{bmatrix}}_{Q^{-1}} \begin{bmatrix} \overline{\boldsymbol{N}} - \boldsymbol{\mu}_1 \\ \widehat{\boldsymbol{N}}_\tau^\eta - \boldsymbol{\mu}_2 \end{bmatrix} \tag{2.10}$$

where $\boldsymbol{\mu}$ is a nominal behavior of $\boldsymbol{N}$, and Q^{-1} is a symmetric positive definite matrix.

Proposition 2.2. *Given a quadratic regularization term as in eq.* (2.10)*, the optimal estimator* $\widehat{\boldsymbol{N}}(t)$ *for eq.* (2.5) *satisfies the quadratic equation system*

$$\operatorname{diag}\left(\widehat{\boldsymbol{N}}(t)\right) \cdot \left(\boldsymbol{s}(t) + 2\gamma\mathcal{Q}_{11}\big(\widehat{\boldsymbol{N}}(t) - \boldsymbol{\mu}_1\big) + 2\gamma\mathcal{Q}_{12}\big(\widehat{\boldsymbol{N}}_\tau^\eta(t) - \boldsymbol{\mu}_2\big)\right) - M \cdot \mathbb{1} = \boldsymbol{0} \, . \tag{2.11}$$

Proof. Since $s(t), \ldots, s(t-\tau)$ are independent, and their probability distribution is

$$p\left(s(\tau)\, ;\, \overline{N}(\tau)\right) = \overline{N}(\tau)^M e^{-(\overline{N}(\tau)-1)s(\tau)},$$

it follows that

$$-\log\left(p\left(s(t),\ldots,s(t-\tau)\,;\,\overline{\boldsymbol{N}}\right)\right)=\sum_{i=t-\tau}^{t}\left((\overline{N}(i)-1)s(i)-M\log\left(\overline{N}(i)\right)\right).$$

We can thus rewrite the estimator (2.5) as

$$\begin{aligned}\arg\min_{\overline{\boldsymbol{N}}}\sum_{i=t-\tau}^{t}&\left((\overline{N}(i)-1)s(i)-M\log\left(\overline{N}(i)\right)\right)\\&+\gamma\left(\overline{\boldsymbol{N}}-\boldsymbol{\mu}_1\right)^T\mathcal{Q}_{11}\left(\overline{\boldsymbol{N}}-\boldsymbol{\mu}_1\right)\\&+2\gamma\left(\overline{\boldsymbol{N}}-\boldsymbol{\mu}_1\right)^T\mathcal{Q}_{12}\left(\widehat{\boldsymbol{N}}_\tau^\eta-\boldsymbol{\mu}_2\right)\\&+\gamma\left(\widehat{\boldsymbol{N}}_\tau^\eta-\boldsymbol{\mu}_2\right)^T\mathcal{Q}_{22}\left(\widehat{\boldsymbol{N}}_\tau^\eta-\boldsymbol{\mu}_2\right).\end{aligned}$$

Setting the gradient with respect to $\overline{\boldsymbol{N}}$ equal to zero yields, for each $i=t-\tau,\ldots,t$,

$$s(i)-\frac{M}{\overline{N}(i)}+2\gamma\left(\mathcal{Q}_{11}^{(i)}\left(\overline{\boldsymbol{N}}-\boldsymbol{\mu}_1\right)+\mathcal{Q}_{12}^{(i)}\left(\widehat{\boldsymbol{N}}_\tau^\eta-\boldsymbol{\mu}_2\right)\right)=0,$$

where $\mathcal{Q}_{11}^{(i)}$ is the i-th row of $\mathcal{Q}_{11}$ (similarly for $\mathcal{Q}_{12}^{(i)}$). Multiplying by $\overline{N}(i)$ and vectorizing the previous equation into a matrix equality leads to eq. (2.11). □

Quadratic regularization terms, as in eq. (2.10), especially capture the design strategies where $\mathcal{R}$ penalizes just the changes between consecutive estimates $N(t),\ldots,N(t-\eta)$. In fact, by defining $\Omega_{ij}\doteq(\boldsymbol{e}_i-\boldsymbol{e}_j)(\boldsymbol{e}_i-\boldsymbol{e}_j)^T$, where $\{\boldsymbol{e}_i\}$ is the standard basis of $\mathbb{R}^n$, $\boldsymbol{x}=[x_1,\ldots,x_n]^T$, and letting $Q^{-1}=\sum_{i,j}q_{ij}\Omega_{ij}$ with $q_{ij}>0$ then $\|\boldsymbol{x}-\mu\mathbb{1}\|_Q^2=\sum_{i,j}q_{ij}(x_i-x_j)^2$. In this case, choices for η larger than $\eta=\tau+1$ are meaningless, since a larger value would just add a constant to the regularization term.

2.3 Properties under a Markovian P2P Model

We now derive the quadratic regularization term as an approximation of the probabilistic model for a simple but practical network example. This example also illustrates an important extension of the algorithm; that it can be used to count the number of nodes that satisfies any property, as long as the nodes can determine this property themselves.

Consider an anonymous peer-to-peer file sharing network, where a certain file is only available at a subset of the peers, and the goal is to estimate how many peers that have this file. At any time, a user who does not have the file can choose to download it, and a user who does have the file can choose to delete it. All peers in the network will participate in the estimation algorithm, but only the peers who

have the file perform the first step of generating new random values. Those peers who do not have the file would instead initialize their state with zeros in order to not affect the max consensus protocol. We further assume that:

- there exists a boundary on the total number of peers[1], say $N_{\max}$;
- downloading and deleting files happen independently among the peers;
- the stochastic process that peer i downloads or deletes the file is a Markov process with (known) probabilities:

$$\begin{aligned} p &\doteq \mathbb{P}\left[x_i(t) = 1 \mid x_i(t-1) = 0\right] \\ q &\doteq \mathbb{P}\left[x_i(t) = 0 \mid x_i(t-1) = 1\right] \end{aligned} \tag{2.12}$$

  where $x_i(t) = 1$ corresponds to peer i having the file at time t, while $x_i(t) = 0$ corresponds to that peer i does not have the file at time t.

Given these assumptions, we derive an estimator for the current time step ($\tau = 0$), but with two-steps of regularization memory ($\eta = 1$).

Derivation of the Regularization Term

Let us consider the Bayesian interpretation of the quadratic regularization term as a log-Gaussian prior on $[N(t), N(t-1)]$. Given the independence assumptions stated above, we need to compute the nominal behavior $\mu \doteq \mathbb{E}\left[N(t)\right]$ and variance

$$Q \doteq \mathbb{E}\left[\begin{bmatrix} N(t) - \mu \\ N(t-1) - \mu \end{bmatrix} \begin{bmatrix} N(t) - \mu \\ N(t-1) - \mu \end{bmatrix}^T\right] .$$

Lemma 2.3. *Let $\alpha \doteq \frac{p}{q}$ be the radio between the two transition probabilities in the Markov chain. Then,*

$$\mu \doteq \mathbb{E}\left[N(t)\right] = \frac{\alpha}{1+\alpha} N_{\max} \tag{2.13}$$

$$\operatorname{var}\left(N(t)\right) = \frac{\alpha}{(1+\alpha)^2} N_{\max} \tag{2.14}$$

$$\operatorname{cov}\left(N(t), N(t-1)\right) = (1-p-q)\frac{\alpha}{(1+\alpha)^2} N_{\max} \tag{2.15}$$

Proof. Notice that $N(t) = \sum_{a=1}^{N_{\max}} x_a(t)$, where the processes x_a are i.i.d.. Thus, let us first compute the expected value, variance and covariance for a single agent.

The Markov process in (2.12) is described by the transition matrix P given by

$$P = \begin{bmatrix} 1-p & p \\ q & 1-q \end{bmatrix} .$$

[1] As stated later in this section, this assumption is not strictly required and could be removed.

The equilibrium distribution, $\boldsymbol{\pi} = \boldsymbol{\pi} P$, for the Markov process is $\boldsymbol{\pi} = \frac{1}{1+\alpha}\begin{bmatrix}1 & \alpha\end{bmatrix}$, thus the expected value is

$$\mathbb{E}\left[x_a(t)\right] = \frac{\alpha}{1+\alpha}.$$

Further, the variance is

$$\operatorname{var}\left(x_a(t)\right) = \mathbb{E}\left[x_a(t)^2\right] - \mathbb{E}\left[x_a(t)\right]^2 = \frac{\alpha}{1+\alpha} - \left(\frac{\alpha}{1+\alpha}\right)^2 = \frac{\alpha}{(1+\alpha)^2}.$$

Finally, for a single agent we have the covariance

$$\begin{aligned}\operatorname{cov}\left(x_a(t), x_a(t-1)\right) &= \mathbb{E}\left[x_a(t)x_a(t-1)\right] - \mathbb{E}\left[x_a(t)\right]\mathbb{E}\left[x_a(t-1)\right] = \\ &= \frac{\alpha}{1+\alpha}(1-q) - \left(\frac{\alpha}{1+\alpha}\right)^2 = (1-p-q)\frac{\alpha}{(1+\alpha)^2}.\end{aligned}$$

For the entire system $N(t) = \sum_{a=1}^{N_{\max}} x_a(t)$ we can simply multiply the results for a single agent by $N_{\max}$, because the different agents are i.i.d., and because of the linearity of the expected value, variance and covariance. □

Thus we have the quadratic regularization

$$Q = N_{\max}\frac{\alpha}{(1+\alpha)^2}\begin{bmatrix}1 & 1-p-q\\ 1-p-q & 1\end{bmatrix}. \tag{2.16}$$

Derivation of the Estimator

We had $\tau = 0$ and $\eta = 1$, thus our variables corresponds to $\widehat{\boldsymbol{N}}(t) = \widehat{N}(t)$, $\widehat{\boldsymbol{N}}_\tau^\eta(t) = \widehat{N}(t-1)$,

$$\begin{aligned}\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \mu &= \frac{\alpha}{1+\alpha}N_{\max},\\ \mathcal{Q}_{11} = \mathcal{Q}_{22} &= \frac{1}{\mu q\big(2-q(1+\alpha)\big)},\\ \mathcal{Q}_{12} = \mathcal{Q}_{21} &= \frac{q(1+\alpha)-1}{\mu q\big(2-q(1+\alpha)\big)}.\end{aligned}$$

In this case, the condition on the optimal estimator (2.11) simplifies into the quadratic equation

$$a\widehat{N}^2(t) + \Big(b\widehat{N}(t-1) + c\Big)\widehat{N}(t) - M = 0 \tag{2.17}$$

where

$$\begin{aligned}a &\doteq 2\gamma\mathcal{Q}_{11},\\ b &\doteq 2\gamma\mathcal{Q}_{12},\\ c &\doteq s(t) - 2\gamma\left(\mathcal{Q}_{11}+\mathcal{Q}_{12}\right)\mu.\end{aligned}$$

The unique admissible solution for $\widehat{N}(t)$ is given by

$$\widehat{N}(t) = \sqrt{\left(\frac{b\widehat{N}(t-1)+c}{2a}\right)^2 + \frac{M}{a}} - \left(\frac{b\widehat{N}(t-1)+c}{2a}\right). \tag{2.18}$$

Remarkably, our penalized ML approach leads to a recursive estimator that is nonlinear but still easy to be implemented in devices with small computational capabilities. The reason that the obtained smoother is nonlinear is because even though we derived the regularization term using Gaussian assumptions on $[N(t), N(t-1)]$, the likelihood term in eq. (2.6) is non-Gaussian. If the likelihood had been Gaussian, then the estimator would have been a linear smoother, leading to a Kalman filter.

Notice that the derivation of Q using Gaussian assumptions is formally incorrect, since it implies that $N(t), N(t-1)$ could take negative values. A formally correct probabilistic interpretation would require the regularization term $\mathcal{R}$ to be derived from the actual prior distribution, but this would lead to a non-quadratic $\mathcal{R}$, and non-closed-form solutions of (2.5). Despite this formal error, the effects of this approximation vanish as $N_{\max}$ increase since $N(t) = \sum_{a=1}^{N_{\max}} x_a(t)$ is approximatively Gaussian due to central limit effects.

The Role of the Regularization Parameter γ

In eq. (2.6), the log-likelihood function $-\log\left(p\left(\boldsymbol{s}(t)\,;\,\overline{\boldsymbol{N}}\right)\right)$ takes into account the experimental evidence, while the regularization term $\mathcal{R}$ reflects the a-priori information about the regularity of the solution. The regularization parameter γ then captures the trade-off between these two components, and represents how much one trusts the regularity assumptions. Notice that the γ maximizing the predictive capabilities of the filter strongly depends on M, i.e., on the accuracy of the experimental evidence.

If $N_{\max}$ is not known a-priori, or if its value is uncertain, then γ can also be tuned on-line, e.g., with cross-validation methods (Hastie *et al.*, 2009, Chap. 7.10). In this case, tuning γ, assuming that the probabilities q and p are known, corresponds to estimating $N_{\max}$ given q, p and M.

2.4 Evaluation of the Size Estimation Algorithm

Let us evaluate the regularization based dynamic network size estimator on the Markovian network model introduced in section 2.3. We generate networks with $N_{\max} = 1000$ total peers, and let the transition probabilities be $p = q = 0.01$. Each active peer is generating $M = 200$ uniformly random samples at each time step, and the regularization parameter is chosen as $\gamma = 0.001$.

We start by noticing the beneficial effects of our regularization approach of eq. (2.18) in fig. 2.2, where we compare the outcomes of our estimator with a point-wise estimation (corresponding to $\gamma = 0$).

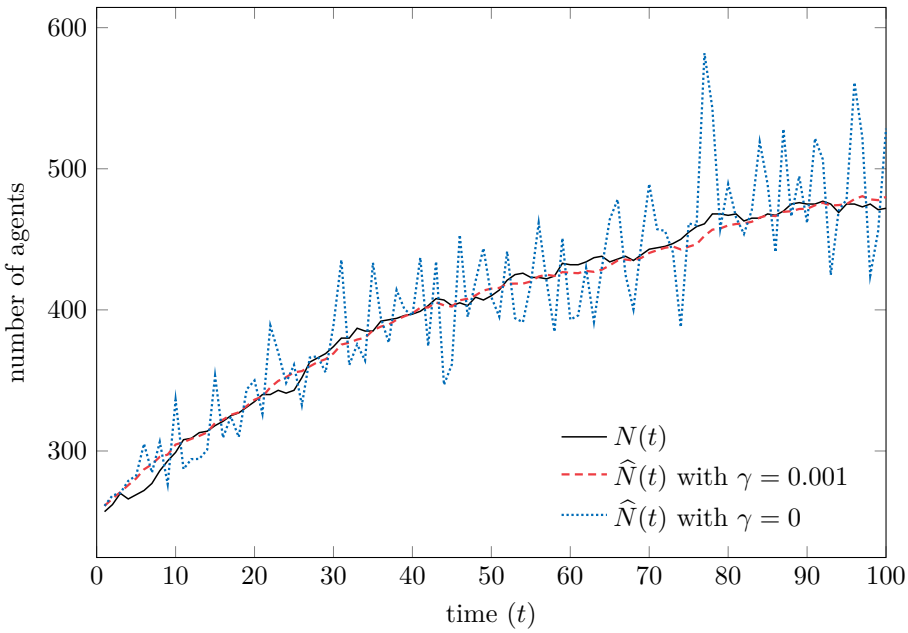


Figure 2.2: Comparison of the results for a regularization based estimator and a point-wise estimator, for the same set of $s(\tau)$. The network consists of $N_{\max} = 1000$ nodes, with transition probabilities $p = q = 0.01$.

Next we examine the effects of the parameters $p, q, \gamma, N_{\max}$ and M on the estimation performance by considering 4 different scenarios: $p = q = 0.1$ or 0.01; $N_{\max} = 1000$ or 2000. For each of these scenarios we evaluate the root-mean-square error (RMSE) by generating 1000 independent network models $N_j(t)$, $t = 1, \ldots, 100$, $j = 1, \ldots, 1000$ from the Markovian model in section 2.3. For each trajectory $N_j(t)$ we compute the estimator of eq. (2.18) using different M's in $[10, 200]$ and different γ's in $[10^{-6}, 10^{-2}]$. In fig. 2.3 each of the 4 subplots then illustrates the dependency on M and γ of the RMSE, defined as:

$$\mathrm{RMSE}(M, \gamma) \doteq \sqrt{\frac{1}{10^5} \sum_{j=1}^{1000} \sum_{t=1}^{100} \Big(N_j(t) - \widehat{N}_j(t\,;\, M, \gamma) \Big)^2}, \tag{2.19}$$

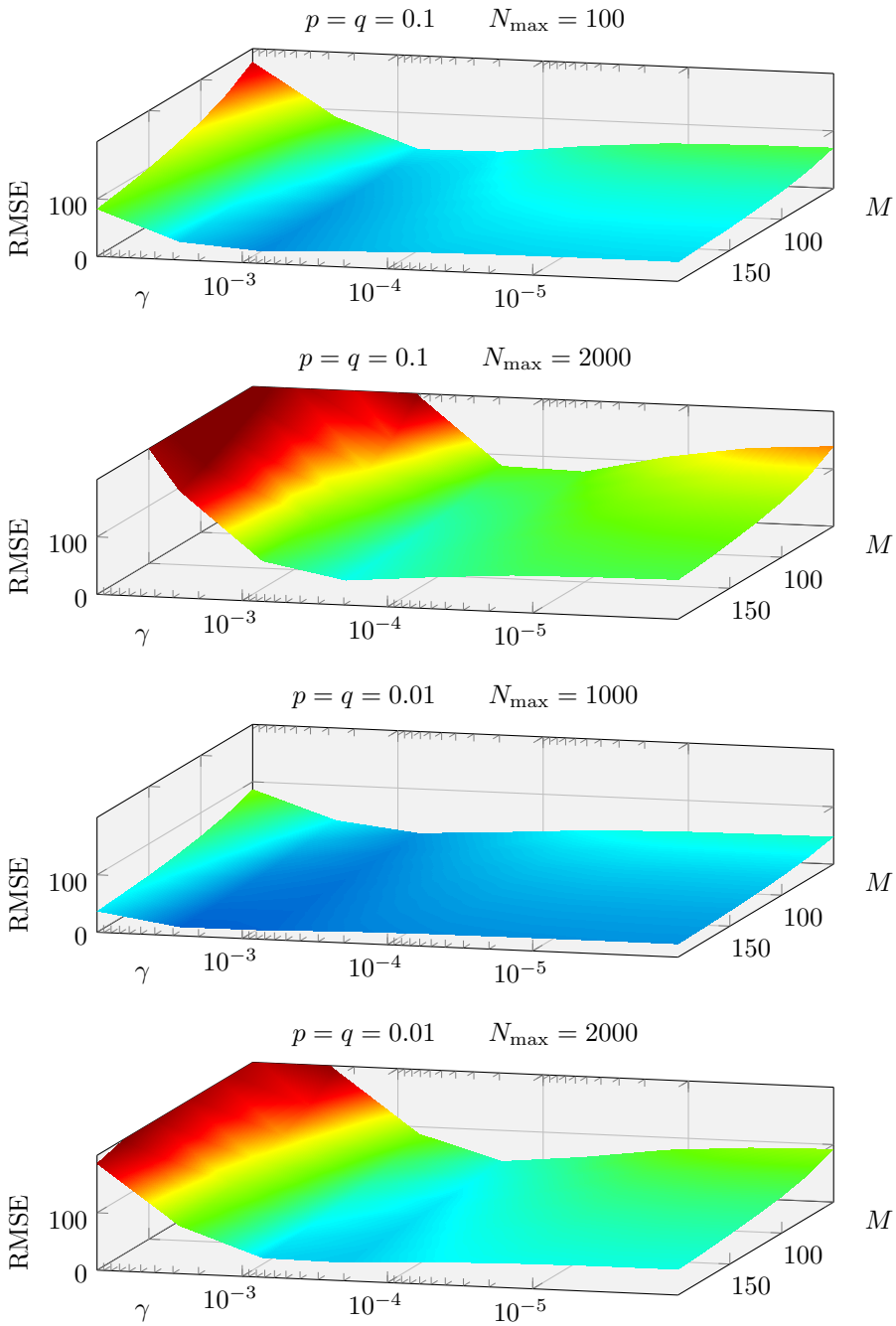


Figure 2.3: Dependency of the average root-mean-square error on the parameters M and γ for various values of p, q and $N_{\max}$.

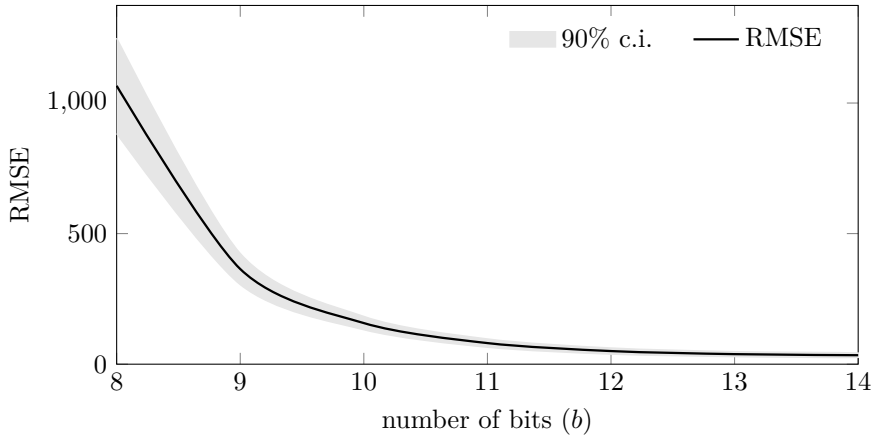


Figure 2.4: Dependency of the RMSE (2.19) on the number of bits used to represent the samples $y_{i,m}(t)$, assuming (2.18) and (2.19) to be computed using 64-bits precisions. $N_{\max} = 1000$, $p = q = 0.01$, $M = 200$ and $\gamma = 0.001$.

Assuming that p, q, M and $N_{\max}$ are fixed, then there exists an optimal regularization parameter γ^* minimizing the RMSE. The behaviors of the four surfaces supports the following intuitive rules-of-thumb for selecting the estimator's parameters.

- if p, q and $N_{\max}$ are fixed, then increasing M leads to a smaller optimal regularization parameter;
- if M and $N_{\max}$ are fixed, then increasing p and q leads to a smaller optimal regularization parameter;
- if p, q and M are fixed, then increasing $N_{\max}$ leads to a smaller optimal regularization parameter.

We finally evaluate how finite representations using only b-bits of the random samples $y_{i,m}(t)$ in algorithm 2.1 can affect the estimation performances, i.e.,

$$y_{i,m}(t) \in \{0, \alpha, 2\alpha, \ldots, 1\} \quad \text{with} \quad \alpha = \frac{1}{2^b - 1} \,. \tag{2.20}$$

Considering again 1000 trajectories $N_j(t)$, $t = 1, \ldots, 100$, $j = 1, \ldots, 1000$ from the network model in section 2.3, with $N_{\max} = 1000$, $p = q = 0.01$, $M = 200$ and $\gamma = 0.001$, as in fig. 2.2. During the communication step the peers only use b-bits precision, but later in the local computation of the estimate (2.18), they use full 64-bits precision. The average RMSE performance index shows (fig. 2.4) that for small networks it is sufficient to represent the random samples $y_{i,m}(t)$ with 12 bits.

Remark. Experiments in figs. 2.2 and 2.3 have been computed with the discretization scheme in fig. 2.4 using 12 bits. With $M = 200$ random samples, and ignoring the communication protocol overheads, this leads to data packets consisting of 300 bytes.

2.5 Estimation of Probability Mass Functions Problem

As we have seen so far, aggregating and estimating data over networks is essential for many distributed systems. However, simple aggregations, such as computing averages, maxima, or sums of a distributed data set, lose a lot of the information contained in the original data. We will now continue and propose algorithms that estimate the entire empirical PMF over anonymous networks. Specifically, we consider protocols that aim to estimate the empirical distributions in the shortest possible time.

Our proposed strategy is based on the max consensus estimator in section 2.2. As the max consensus protocol convergence is bounded by the number of steps needed to transmit information between arbitrary nodes in the network, this is one of the fastest possible aggregation mechanism. The emphasis on *fast* convergence is because the time aspect is a crucial factor in many practical situations, for example in control applications.

From an algorithmic point of view our strategy departs from Borges *et al.* (2012); Jesus (2012); Sacha *et al.* (2010) by substituting the average consensus schemes with max consensus. This apparently minor modification actually makes the two estimators completely different, and opens up for a variety of novel problems. In fact, while the average consensus scheme requires exchanging very few scalars per iteration, and where the agents compute the *exact* PMF asymptotically in time, the max consensus scheme converges much faster than the average consensus scheme, but not to the exact value. Again, the statistical performance depends on how many scalars are exchanged per iteration. We specifically compare the temporal behavior of these two strategies.

Statement of the PMF Estimation Problem

Consider a strongly connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of $N = |\mathcal{V}|$ agents communicating through the links $\mathcal{E}$. In this section we assume the network to be static, and return to the problem of estimating PMFs in time dependent networks in section 2.9. Let $\mathcal{N}_i$ denote the set of neighbors of agent i, and $\mathcal{N}_i^{(t)}$ the set of the t-steps neighbors of agent i, i.e., all agents that can be reached by paths of length at most t from agent i. Recall that $\mathcal{N}_i^{(t)}$ can be defined for $t = 0$ as $\mathcal{N}_i^{(0)} = \{i\}$ and, for $t \geq 1$, through the recursion

$$\mathcal{N}_i^{(t)} \doteq \bigcup_{j\,:\,(i,j)\in\mathcal{E}} \mathcal{N}_j^{(t-1)} \,. \tag{2.21}$$

Let every agent $i \in \mathcal{V}$ belong to a discrete state $z_i \in \mathbb{N}_B \doteq \{0, \dots, B-1\}$, e.g., given by sensor measurements, where $\mathbb{N}_B$ is the set of plausible states. We are then interested in distributively estimating the relative frequencies of the local states $z_1, \dots, z_N$, i.e., if $n_b \doteq \left|\{i \ : \ z_i = b\}\right|$ is the number of agents in state b, then we aim to estimate the PMF

$$p_b \doteq \frac{n_b}{N}, \qquad b \in \mathbb{N}_B, \tag{2.22}$$

given that the network size N is unknown, while the plausible states $\mathbb{N}_B$ are known.

We focus on distributed algorithms where each agent $i \in \mathcal{V}$ has a local variable $x_i(t)$ that can be modified at time $t+1$ by accessing the states $x_j(t)$'s of the neighboring nodes, and performing the aggregation operation

$$x_i(t+1) = f\left(x_i(t), x_{j_1}(t), x_{j_2}(t), \dots\right), \quad j_1, j_2, \dots \in \mathcal{V}_i$$

that preserves the dimension of $x_i(t)$. At every time t, each agent also computes a local estimate of the PMF function from the local variable $x_i(t)$,

$$\widehat{p}_b^{(i)}(t) = g\left(x_i(t)\right)$$

for an appropriate estimation function $g(\cdot)$.

The estimation strategy is thus defined by the initial variables $x_i(0)$, the update function f and the estimation function g. In order to compare different estimation strategies we consider the mean squared error (MSE) as a performance index, i.e.,

$$\mathrm{MSE}\big(\widehat{p}_1, \dots, \widehat{p}_B\big) \doteq \mathbb{E}\left[\frac{1}{N \cdot B} \sum_{b \in \mathbb{N}_B, i \in \mathcal{V}} \left(p_b - \widehat{p}_b^{(i)}\right)^2\right] \tag{2.23}$$

where the expectation is taken over all initial conditions.

2.6 PMF Estimators Based on Consensus Protocols

We consider two particular estimators, one based on the average consensus strategy (see also Borges *et al.* (2012); Jesus (2012); Sacha *et al.* (2010)), and a strategy based on max consensus in section 2.2.

In the following, we abstract away the message transmission and consider a distributed system where agents communicate by synchronous rounds. At each round, and over each edge, only a constant size message is transmitted, and no messages are lost.

Remark. For notational simplicity we consider synchronous communication steps. Nonetheless this could be relaxed for both estimators, since they can be adapted to operate with gossip asynchronous transmissions.

Estimator Based on Average Consensus

In the average consensus protocol, the local variable is a B-dimensional real vector $x_i(t) \in \mathbb{R}^B$ containing the estimate of the PMF. At initialization, each node sets its local variable based on its own state,

$$x_i^{(b)}(0) = \begin{cases} 1, & \text{if } z_i = b, \\ 0, & \text{otherwise.} \end{cases}$$

Let $\boldsymbol{x}^{(b)}$ denote the vector of all agents' states $x_i^{(b)}$. It is known that if at each time the local variables are updated with an average consensus update like

$$\boldsymbol{x}^{(b)}(t+1) = W\boldsymbol{x}^{(b)}(t), \quad b \in \mathbb{N}_B \tag{2.24}$$

where W is a doubly-stochastic weight matrix (for example chosen as the Metropolis weights), then, assuming perfect computations[2], every $x_i^{(b)}(t)$ converges to the average of the initial values (Fagnani and Zampieri, 2008). Thus

$$x_i^{(b)}(t) \xrightarrow{t\to\infty} \frac{1}{N}\sum_{j\in\mathcal{V}} x_i^{(b)}(0) = \frac{n_b}{N} = p_b.$$

The PMF estimate is simply the local state,

$$\widehat{p}_b^{(i)}(t) = x_i^{(b)}(t). \tag{2.25}$$

To describe the convergence properties of eq. (2.24), recall that the estimation error can be bounded by an exponential function (Xiao and Boyd, 2003), i.e., by

$$\left|\left|p_b - \widehat{p}_b(t)\right|\right|_2 \leq ce^{-\alpha t}$$

where c and α depend on the initial condition, the network topology and the choice of the weights.

Remark. We do not consider more advanced protocols, such as accelerated average consensus (Aysal *et al.*, 2009), or finite-time average consensus (Yuan *et al.*, 2011). The reason is that we want to characterize the simplest averaging algorithm, with the smallest demands from both communication and computational points of view.

Estimator Based on Max Consensus

In the max consensus based estimator, the local variable is instead a $B \times M$-dimensional real matrix $x_i(t) \in \mathbb{R}^{B\times M}$ whose elements are initially generated from a uniform random distribution based on the local state as

$$x_i^{(b,m)}(0) \sim \begin{cases} \mathcal{U}[0,1], & \text{if } z_i = b \\ 0, & \text{otherwise} \end{cases} \tag{2.26}$$

[2]For simplicity we do not consider the quantization effects (Carli *et al.*, 2010).

where $\mathcal{U}[0,1]$ is the uniform distribution between 0 and 1. Then at each time t, the local variables are updated with the max consensus update

$$x_i^{(b,m)}(t) = \max_{j \in \mathcal{V}_i} \left\{ x_j^{(b,m)}(t-1) \right\}, \quad b \in \mathbb{N}_B, m = 1, \ldots, M. \tag{2.27}$$

Notice that the definition of t-steps neighborhood $\mathcal{N}_i^{(t)}$ precisely captures the agents that contributed to the generation of $x_i^{(b,m)}(t)$, i.e.,

$$x_i^{(b,m)}(t) = \max_{j \in \mathcal{N}_i^{(t)}} \left\{ x_j^{(b,m)}(0) \right\}. \tag{2.28}$$

Let $N_i^{(t)} \doteq \left| \mathcal{N}_i^{(t)} \right|$ be the number of t-step neighbors, then

$$p_b^{(i)}(t) \doteq \frac{\left| \{ i \in \mathcal{N}_i^{(t)} \ : \ z_i = b \} \right|}{N_i^{(t)}}, \tag{2.29}$$

and $n_b^{(i)}(t) \doteq p_b^{(i)}(t) N_i^{(t)}$. As was shown in eq. (2.7), the ML estimator for $n_b^{(i)}(t)$ given the $x_i^{(b,m)}(t)$'s was

$$\widehat{n}_b^{(i)} = -\left(\frac{1}{M} \sum_{m=1}^{M} \log \left(x_i^{(b,m)} \right) \right)^{-1}. \tag{2.30}$$

Now, since

$$p_b^{(i)}(t) = \frac{p_b^{(i)}(t)}{\sum_{\beta \in \mathbb{N}_B} p_\beta^{(i)}(t)} = \frac{n_b^{(i)}(t)}{\sum_{\beta \in \mathbb{N}_B} n_\beta^{(i)}(t)}$$

because of the functional invariance property of ML estimators (Casella and Berger, 2002, Thm. 7.2.10, p. 320), the ML estimate of $p_b^{(i)}(t)$ given the $x_i^{(b,m)}(t)$'s is

$$\widehat{p}_b^{(i)}(t) = \frac{\widehat{n}_b^{(i)}(t)}{\sum_{\beta \in \mathbb{N}_B} \widehat{n}_\beta^{(i)}(t)}. \tag{2.31}$$

For $t \geq d$ (d being the network diameter) the max consensus strategy converges globally, and $n_b^{(i)}(t) = n_b$, thus the PMF estimated $p_1^{(i)}(t), \ldots, p_B^{(i)}(t)$ converges to an estimate of the global PMF $p_1, \ldots, p_B$.

Remarkably, this estimator provides additional estimates of the distributions of the states in every t-steps neighborhood. Considering a certain agent i, the set of $p_b^{(i)}(0), p_b^{(i)}(1), \ldots$ correspond to local views of the neighborhood's empirical distribution that can be used by i to rapidly infer if close neighbors tend to have similar states.

Notice that the statistical properties of the estimator in eq. (2.31) are essentially different from the previous network size estimation scheme, since the vectors

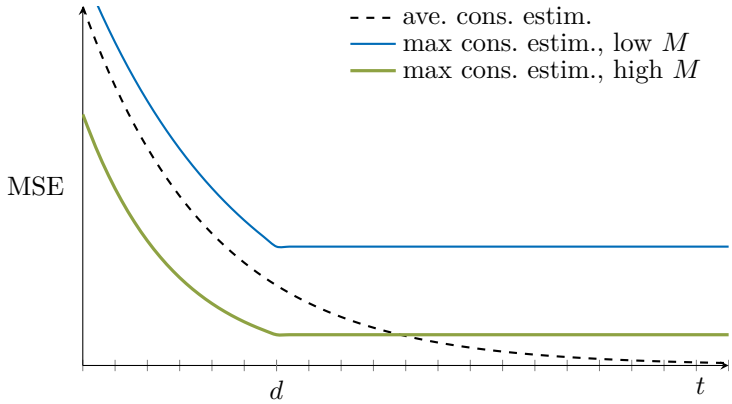


Figure 2.5: Graphical representation of the properties from the estimators. By increasing M it is possible to let the max consensus estimator (2.31) perform better than the average consensus scheme (2.25) for $t \leq d$.

$\left[\widehat{p}_1^{(i)}(t), \ldots, \widehat{p}_B^{(i)}(t)\right]$ have correlated components. We also notice that appropriate termination rules can be based on estimates of the diameter d of the network, again obtained by exploiting max consensus approaches as done by Cardoso *et al.* (2009); Garin *et al.* (2012).

Finally, notice that under continuity assumptions, the choice of stochastic generation mechanism proposed in (2.26) is general, since as soon as we neglect quantization effects, substituting $\mathcal{U}[0,1]$ with another continuous probability distribution leads to estimators with identical statistical performance (Varagnolo *et al.*, 2010).

Summary of the Differences Between the Two Estimators

The max consensus scheme (2.31) converges in finite d steps to an estimate of the true PMF. Given a fixed M, its MSE (2.23) will vary up to time $t = d$ and then remain constant. By increasing the sample size M, the MSE curves are also expected to get closer and closer to zero, due to the consistency property of ML estimators.

The average consensus scheme (2.25) requires nodes to exchange less information, and is in general converging asymptotically for $t \to +\infty$. These comments are graphically represented in fig. 2.5.

The aim is to find conditions on M and on the network for which it is possible to state which algorithm is preferred for $t \leq d$, i.e., when time is a concern. To this end, we first need to describe the statistical properties of the max consensus estimator.

2.7 Statistical Characterization of the Max Consensus PMF Estimator

As stated before, interrupting the max consensus protocol before it has reached consensus is equivalent to estimating the PMF for a t-step neighborhood. Thus, for notational simplicity we consider the stationary state where the max consensus has already been reached, i.e., where $x_i^{(b,m)}(t) = x^{(b,m)} \doteq \max_{i\in\mathcal{V}} \left\{ x_i^{(b,m)}(0) \right\}$. With this assumption the joint PMF $p\left(\widehat{n}_b \,;\, n_1, \ldots, n_B, M\right)$ is equal to the local $p\left(\widehat{n}_b^{(i)}(t) \,;\, n_1^{(i)}(t), \ldots, n_B^{(i)}(t), M\right)$. To derive this distribution, consider that $x^{(b,m)}$ is statistically independent of the parameter n_β if $b \neq \beta$. Thus, from simple order-statistics arguments (David and Nagaraja, 2004),

$$p\left(x^{(b,m)} \,;\, n_1, \ldots, n_B\right) = p\left(x^{(b,m)} \,;\, n_b\right) = n_b \left(x^{(b,m)}\right)^{n_b - 1}$$

for all m (omitting the dependency on the parameter M for notational simplicity). Since the states $x^{(b,m)}$'s are i.i.d. we have

$$p\left(x^{(b,1)}, \ldots x^{(b,M)} \,;\, n_b\right) = \prod_{m=1}^{M} p\left(x^{(b,m)} \,;\, n_b\right) = n_b^M \prod_{m=1}^{M} \left(x^{(b,m)}\right)^{n_b - 1} \tag{2.32}$$

To derive the probability density $p\left(\widehat{n}_b \,;\, n_b\right)$ consider that $y \doteq -\log\left(\left(x^{(b,m)}\right)\right)$ is an exponential random variable with rate n_b, i.e.,

$$p\left(y \,;\, n_b\right) = \begin{cases} n_b e^{-n_b y} & \text{if } y \geq 0, \\ 0 & \text{otherwise .} \end{cases} \tag{2.33}$$

From eq. (2.30), $M\widehat{n}_b^{-1}$ is the sum of M i.i.d. exponential random variables with rate n_b, i.e., $M\widehat{n}_b^{-1}$ is a Γ variate with shape M and scale $\dfrac{1}{n_b}$. Thus $M^{-1}\widehat{n}_b \sim \text{I-}\Gamma\left(M, n_b\right)$, where $\text{I-}\Gamma\left(\cdot,\cdot\right)$ is the inverse Gamma distribution with shape M and scale Mn_b, i.e.,

$$\begin{aligned} p\left(\widehat{n}_b \,;\, n_b, M\right) &= \text{I-}\Gamma\left(M, Mn_b\right) \\ &= \Gamma\left(M\right)^{-1} \frac{1}{\widehat{n}_b} \left(\frac{Mn_b}{\widehat{n}_b}\right)^M \exp\left(-\frac{Mn_b}{\widehat{n}_b}\right). \end{aligned}$$

For the estimate in eq. (2.31), $\widehat{p}_b$ is the ratio of correlated sums of inverse-Gamma variates, each with its own scale.

Unfortunately, to the best of our knowledge there exists no current available literature describing a distribution of this kind. The closest manuscripts characterize ratios of the form $\frac{x}{x+y}$, where x and y are independent inverse Gamma variates (Ali *et al.*, 2007). Moreover both the Gamma and inverse Gamma distributions are not closed, i.e., linear combinations of independent copies of these

kinds of variates do not have the same original distribution up to location and scale parameters (Witkovský, 2001). This means that there is, in general, no possibility to reduce the fraction (2.31) to the case described by Ali *et al.* (2007), and the characterization of the statistical properties of $\widehat{p}_b$ must rely on Monte Carlo (MC) integration methods.

Case $\mathbb{N}_B = \{0, 1\}$

Consider the restricted case when the distribution only consists of two different states, $\mathbb{N}_B = \{0, 1\}$. Then $\widehat{p}_b^{(i)}(t)$ becomes a special ratio that is described by Ali *et al.* (2007), with probability density

$$p_{\widehat{p_0}}(x\ ;\ n_0, n_1, M) = \frac{\Big(x(1-x)\Big)^{M-1}}{\left(\dfrac{n_0}{n_1}\right)^{M} B(M, M)} \left(1 + \frac{n_1 - n_0}{n_0} x\right)^{-2M}, \tag{2.34}$$

where $B(\cdot, \cdot)$ is the Beta function, and $x \in [0, 1]$. Its cumulative distribution is given by eq. (2.35)

$$F_{\widehat{p_0}}(x\ ;\ n_0, n_1, M) = \frac{\left(1 + \frac{n_1}{n_0}\frac{1-x}{x}\right)^{-M}}{M \cdot B(M, M)} \cdot {}_2F_1\left(M, 1-M; M+1; \left(1 + \frac{n_1}{n_0}\frac{1-x}{x}\right)^{-1}\right) \tag{2.35}$$

where

$${}_2F_1(a, b; c; x) \doteq \sum_{i=0}^{+\infty} \frac{(a)_i (b)_i}{(c)_i \cdot i!} x^i \tag{2.36}$$

is the Gauss hypergeometric function and

$$(x)_i \doteq x(x+1)\cdots(x+i-1) \tag{2.37}$$

is the so called Pochhammer symbol (with the convention that $(x)_0 = 1$). From this, it is possible to compute the moments of $\widehat{p}_0$, and in particular of $\widehat{p}_0 - \mathbb{E}[\widehat{p}_0])$, using the relation

$$\mathbb{E}\left[(\widehat{p}_0)^k\right] = \begin{cases} \dfrac{B(M+k, M)}{B(M, M)} \mathcal{F}(k, M, n_0, n_1) & \text{if } n_0 > n_1, \\ \left(\dfrac{n_0}{n_1}\right)^k \dfrac{B(M+k, M)}{B(M, M)} \mathcal{F}(k, M, n_1, n_0) & \text{otherwise.} \end{cases} \tag{2.38}$$

where

$$\mathcal{F}(k, M, a, b) \doteq {}_2F_1\left(k, M; 2M+k; \frac{a-b}{a}\right)$$

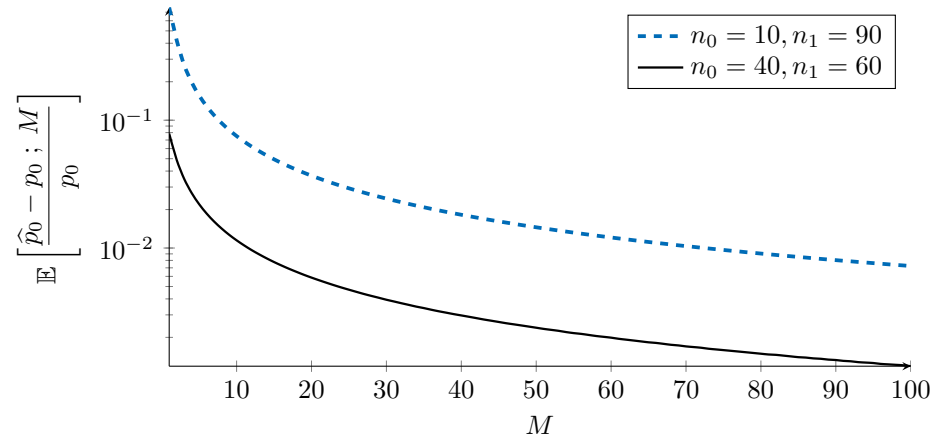


Figure 2.6: The relative bias $\mathbb{E}\left[\frac{\widehat{p_0}-p_0}{p_0} ; M\right]$ dependency on M for different values of n_0 and n_1. The estimator is unbiased if $n_0 = n_1$.

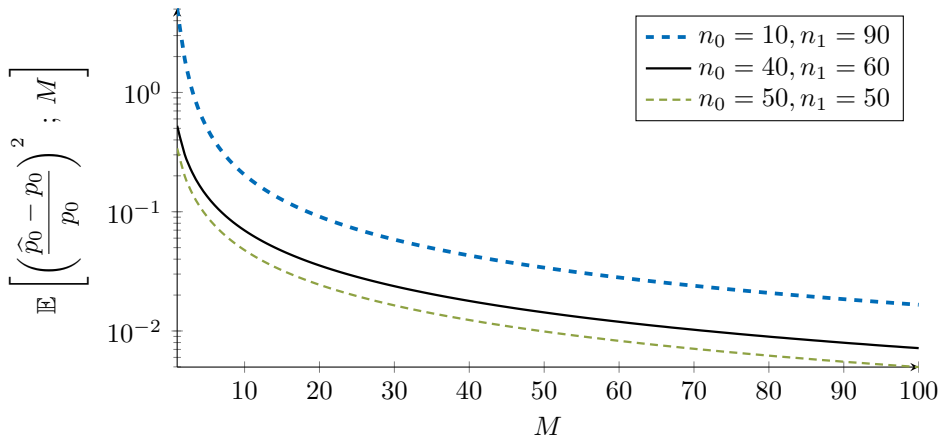


Figure 2.7: The relative MSE $\mathbb{E}\left[\left(\frac{\widehat{p_0}-p_0}{p_0}\right)^2 ; M\right]$ dependency on M for different values of n_0 and n_1.

(notice that n_0 and n_1 appear in inverted positions in the two cases in eq. (2.38)).

When $n_0 = n_1$ the estimators are unbiased for every M, otherwise, as expected, they are only asymptotically unbiased (for $M \to +\infty$).

In fig. 2.6 we evaluate the relative bias, and in fig. 2.7 the relative MSE of the estimator, based on the design parameter M and on the distribution of the states. Notice that the MSE performance follows the typical $\frac{1}{M}$ behavior for this kind of

estimators.

Remark. The performance indicators summarized in figs. 2.6 and 2.7 are valid for general $\widehat{p}_b^{(i)}(t)$'s when associated with the local $n_b^{(i)}(t)$'s. The derivations of this section also characterize the behavior of the estimators during the transient.

2.8 Evaluation of PMF Estimators

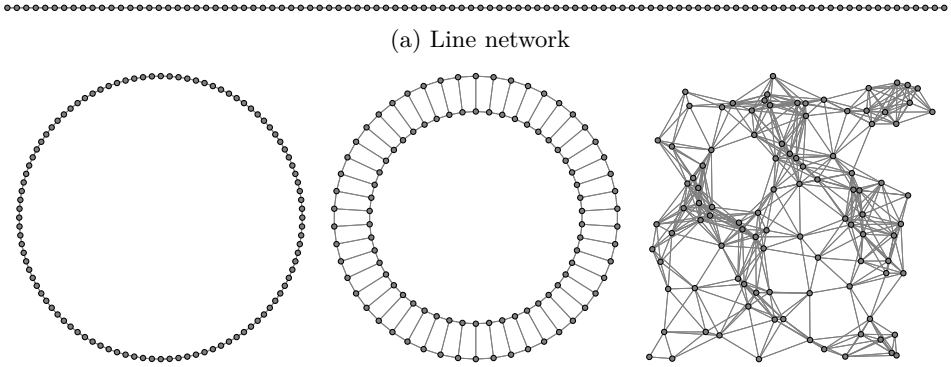
Here we evaluate the performance of the average consensus based estimator of eq. (2.25) and the max consensus based estimator of eq. (2.31) during their transient phases. The primary goal is to determine when to use each algorithm, and how to tune the parameter M for the max consensus estimator.

We consider four different network topologies, the line topology (fig. 2.8a), the cyclic topology (fig. 2.8b), the cyclic grid topology (fig. 2.8c), and a geometric random topology (fig. 2.8d), each network consisting of 100 agents.

We evaluate the algorithms through Monte Carlo (MC) simulations, using the MSE from eq. (2.23) as the performance index, where the mean is taken over both all agents and all MC simulations. On each network the communication protocol proceeds in synchronous steps, where nodes cyclically repeat the algorithms described in eq. (2.24) and eq. (2.27).

In the first experiment, fig. 2.9, the initial state is selected randomly for each MC simulation, where each agent is placed in either state $z_i = 0$ or $z_i = 1$ with equal probability. The figure shows the 95% confidence intervals for both the average consensus based estimator as well as for the max consensus based estimator with $M = 10$, $M = 100$ and $M = 1000$.

As expected, the average consensus based estimator converges *asymptotically* to the true value, while the max consensus based estimator converges *in finite time*



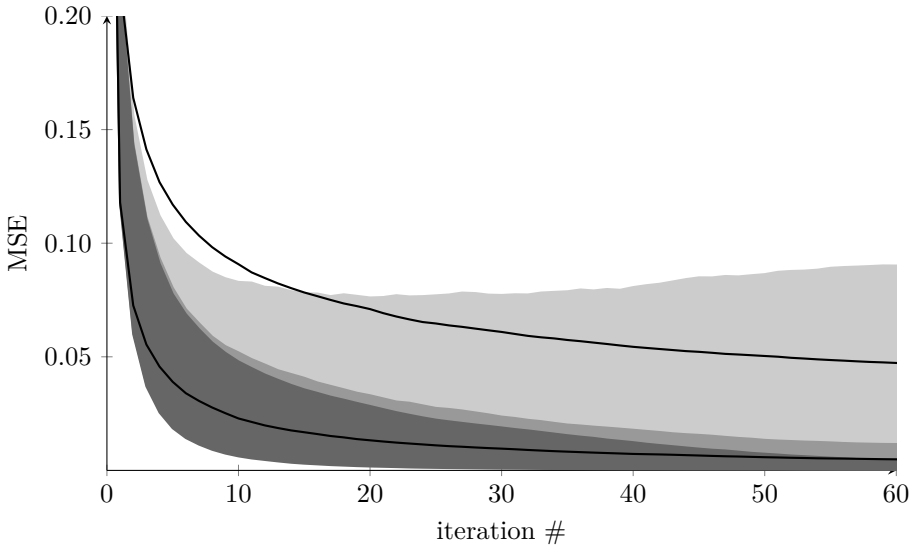
(a) Line network

(b) Cyclic network

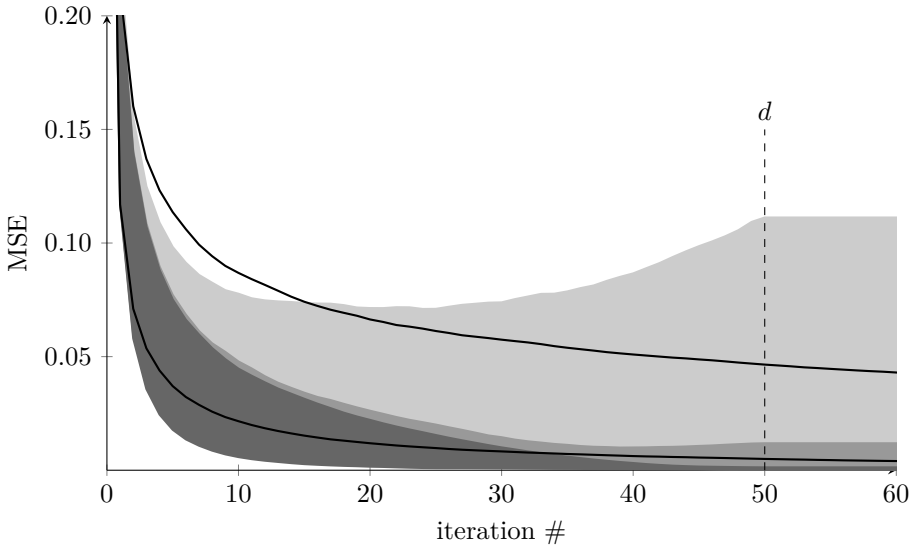
(c) Cyclic grid network with (2×50) nodes

(d) Geometric random network

Figure 2.8: Network topologies, with 100 nodes.

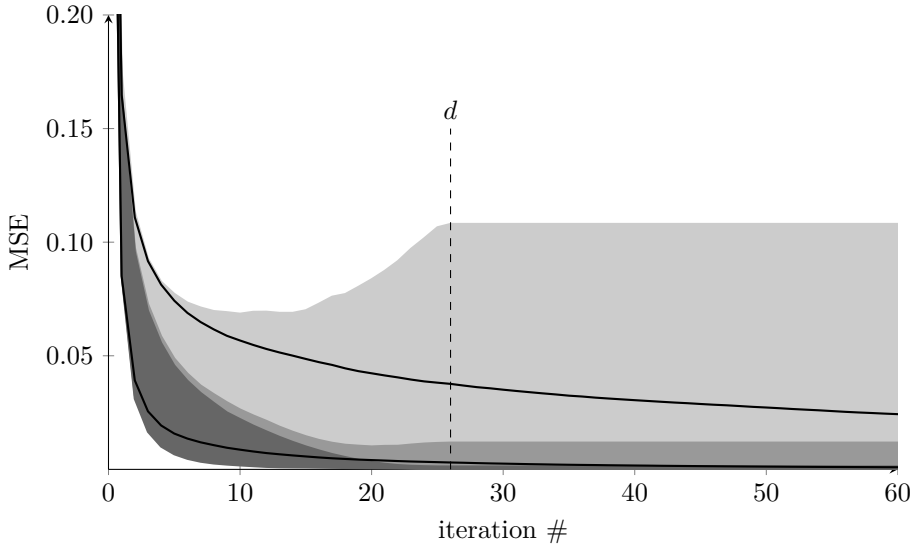


(a) Line network (Diameter $d = 99$)

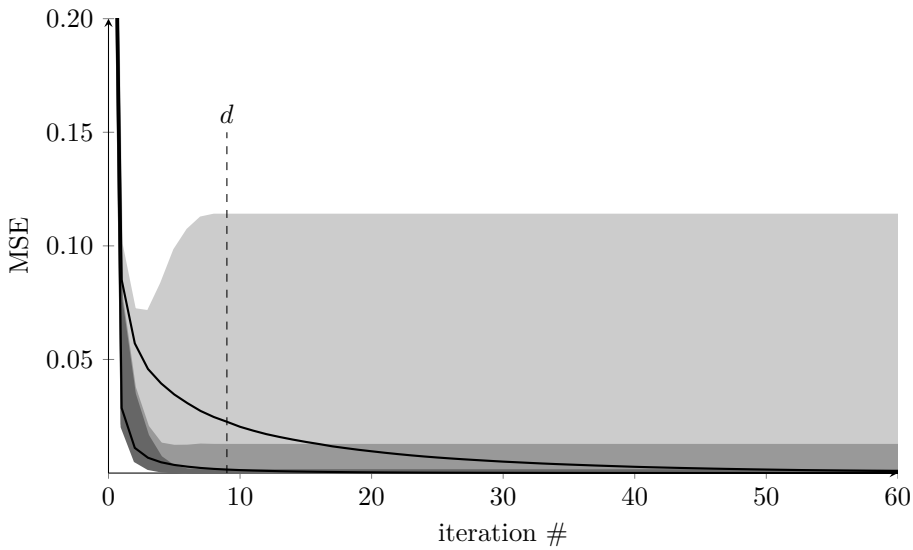


(b) Cyclic network

$M = 10$ $M = 100$ $M = 1000$ Average



(c) Cyclic grid network (2×50 nodes)



(d) Geometric network

Figure 2.9: Comparison of max consensus based estimator against the average consensus based estimator. Each network consists of 100 nodes, and the network diameter d is marked in the figures. The shaded regions mark the 95% confidence interval for the max consensus estimator, while the two solid lines mark the upper and lower bound of the 95% confidence interval for the average consensus estimator.

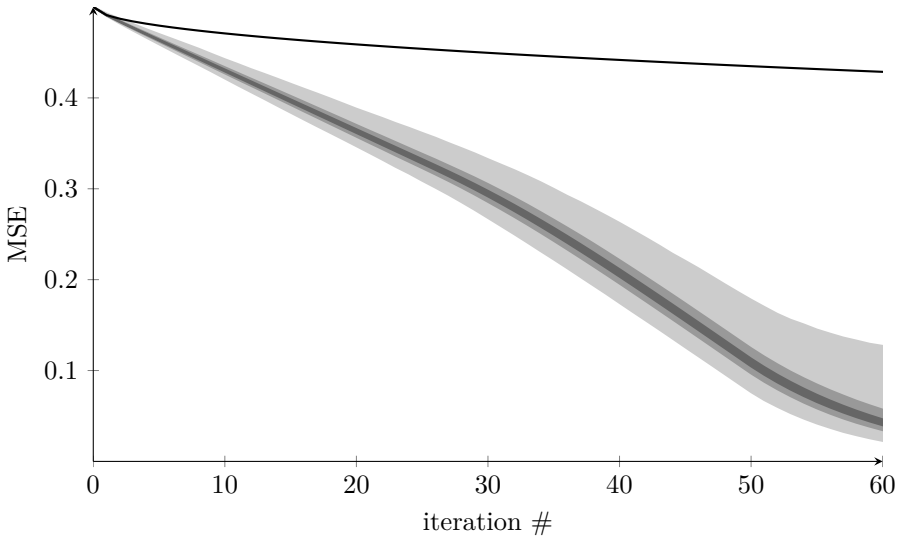
(after d steps, where d is the diameter of the network). However, the max consensus based estimate does not converge to the true value, but instead its MSE decreases with M. In this scenario the choices of $M = 100$ and $M = 1000$ yield similar precisions that outperform the average consensus in most reasonable time scales.

We observe that for the max consensus-based scheme a remarkable phenomenon may appear, especially when M is small (e.g., $M = 10$ in fig. 2.9), the MSE actually increases with the number of iterations. This phenomenon appears because the MSE index sums the agents' local MSEs, and small M's induce estimates with high statistical variance, i.e., increase the chances that at least one agent will have some $\widehat{p}_b^{(i)}(t)$ noticeably overestimated. At time $t = 1$ this overestimation has not yet influenced a majority of the agents and the overall MSE, since it only affected the erroneous agent, but as time passes, the max consensus protocol spreads this overestimation through the network of agents, which is seen in the MSE.

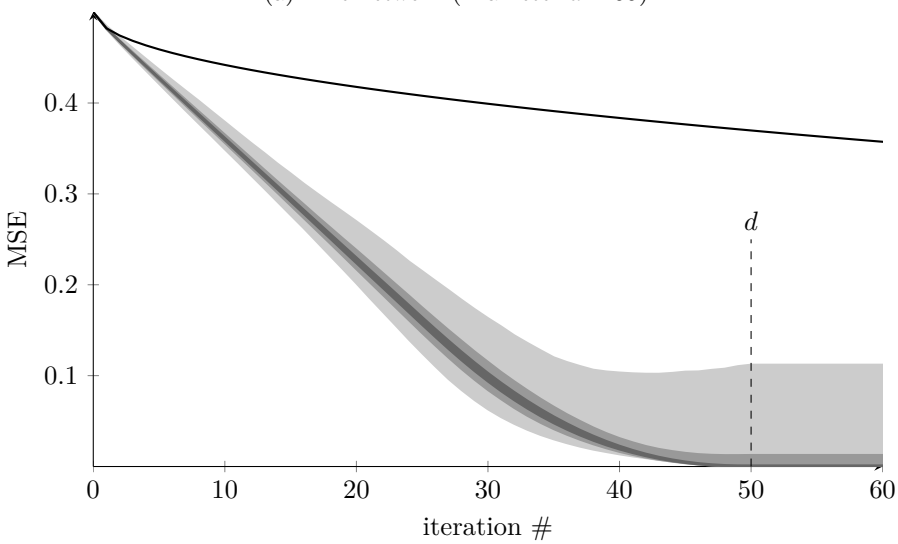
In the second experiment, fig. 2.10, we switch the initial condition to a single worst-case distribution of the states z_i, where the leftmost half of the agents in fig. 2.8 are in state 0 and the rightmost half are in state 1. Notice that this corresponds exactly to a spatial correlation between the agents positions and their measurements, which is actually a reasonable assumption for estimation applications in wireless sensor networks.

Since there is only one fixed initial state, the average consensus based estimator is now deterministic and unique. The figure thus compares the confidence intervals of the max consensus estimators (depending upon the realization of the initial random sample $x_i^{(b,m)}$'s) against the performance of the deterministic average consensus estimate.

The first thing we notice is that the convergence speed is slower than for the randomized initial distribution, because in that case each t-step neighborhood was a reasonable sample of the complete distribution, while in this case a t-step neighborhood ($t \ll d$) is not a good representation of the complete distribution. Still, this example shows even more clearly that a max consensus based estimator can be much faster and accurate than an average consensus based counterpart, even for very small M's (even though a larger M improves the accuracy). The motivation is that the max consensus protocol has a much faster mixing time, and if the distribution of states in the network topology is not homogeneous then the max consensus is much more efficient at propagating information about certain states through the network.

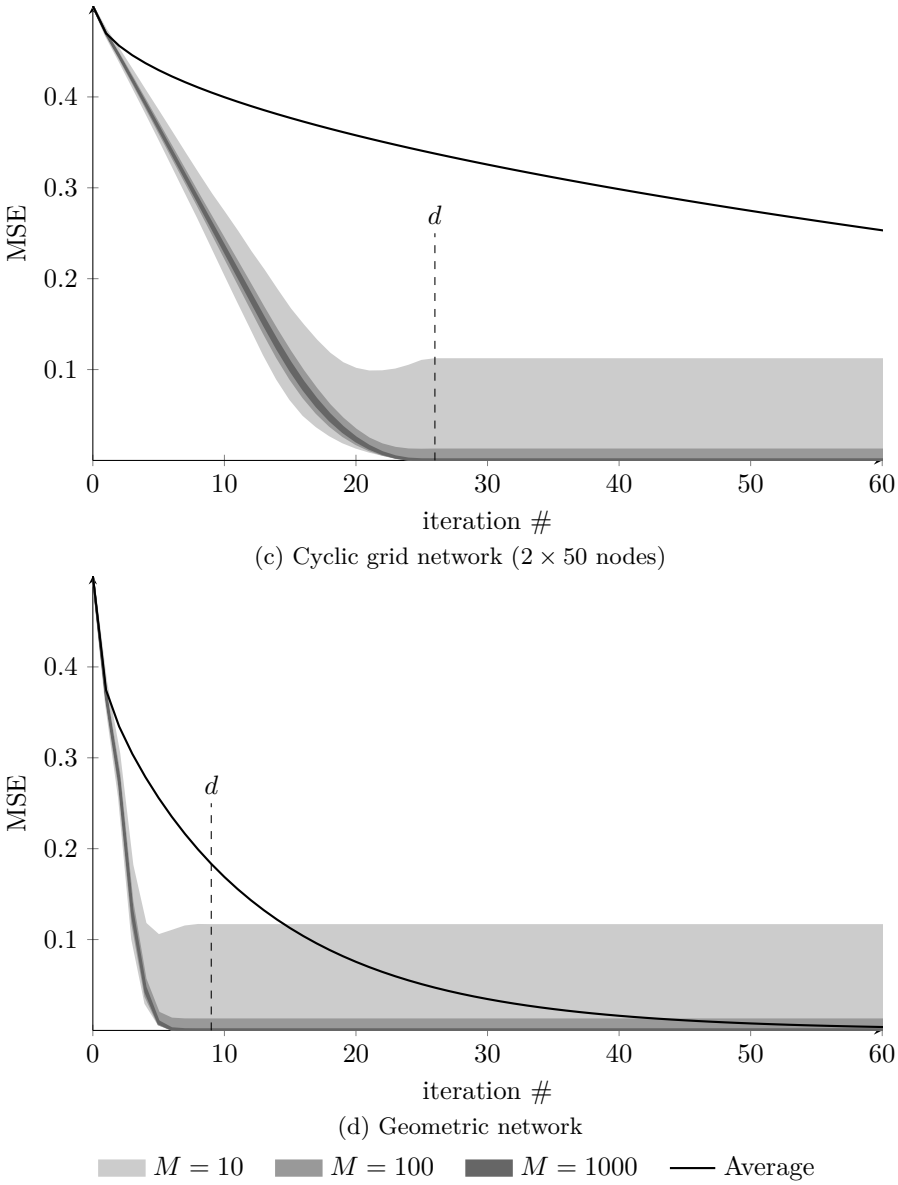


(a) Line network (Diameter $d = 99$)



(b) Cyclic network

$M = 10$ $M = 100$ $M = 1000$ Average



(c) Cyclic grid network (2×50 nodes)

(d) Geometric network

Figure 2.10: Comparison of max consensus based estimator against the average consensus based estimator for a single worst-case initial condition. Each network consists of 100 nodes, and the initial state is determined by the agents spatial configuration. The shaded regions mark the 95% confidence interval for the max consensus estimator, while the solid line marks the deterministic estimation for the average consensus estimator.

2.9 Estimating Time Dependent PMFs

The PMF estimator we proposed in section 2.6 is using the max consensus based network size estimation scheme that we also developed in section 2.2 to handle time dependent networks. It is therefore straightforward to handle time dependent PMF estimation by substituting the regularization based estimator in eq. (2.6) into the PMF estimator in eq. (2.30).

This can be demonstrated by considering a Markov chain driven state, similar to section 2.3. We create a network with $N = 1000$ nodes, each node initially belonging to one out of four states $z_i \in \mathbb{N}_4 = \{0, 1, 2, 3\}$. The nodes change their state according to a Markov process, i.e., there exists a transition matrix $P \in \mathbb{R}^{4\times 4}$ with entries p_{ij} giving the probabilities

$$p_{ij} \doteq \mathbb{P}\left[z_i(t+1) = j \mid z_i(t) = i\right].$$

Thus, after updating their state, they generate $B \times M$ random values (all except M equal to zero), according to eq. (2.26). After the max consensus step, each node can estimate $\widehat{n}_b$, $b \in \mathbb{N}_4$, where we replace eq. (2.30) with the regularization based estimator of eq. (2.6), and finally compute the dynamic PMF estimate through eq. (2.31).

In fig. 2.11 we evaluate the regularization based dynamic PMF estimator ($\gamma = 0.005$) against the point-wise PMF estimator ($\gamma = 0$), both using the same realization of random samples, and a small $M = 10$. In the regularization based estimator we only use one step memory, $\tau = 0$, $\eta = 1$.

Since the number of samples are relatively small (only $M = 10$, compared to figs. 2.9 and 2.10), the variance in the estimates are quite significant, but the regularization based estimator does successfully reduce the variance and yields an improved estimate of the true distribution.

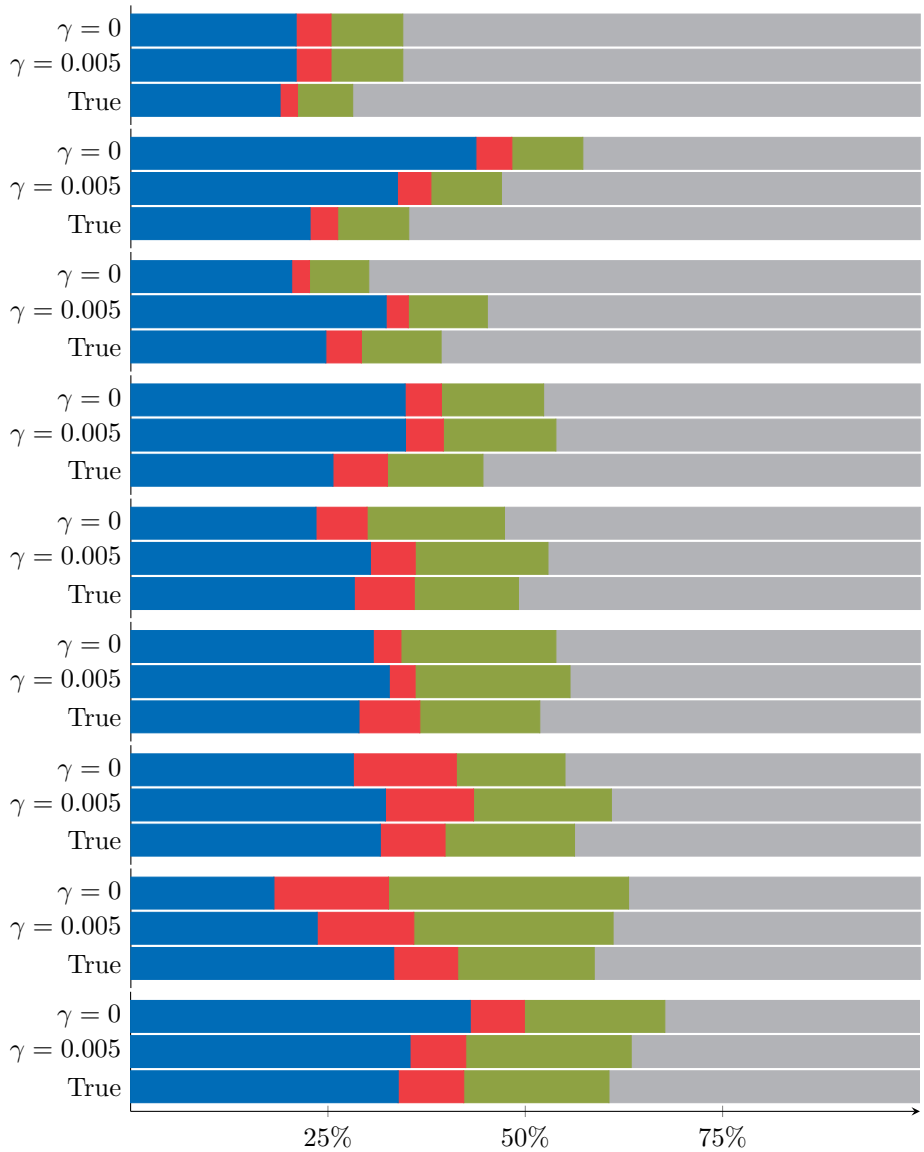


Figure 2.11: Comparing the regularization based ($\gamma = 0.005$) and the point-wise ($\gamma = 0$) PMF estimator against the true dynamic distribution, of four states. In the first step (top figure), both estimators yield the same result since the regularization based estimator has not yet initialized its memory, but in the following steps it reduces the variance compared to the point-wise estimator.

2.10 Conclusions

In this chapter, the max consensus protocol was used to derive two specific estimators, one which estimates the network size in time dependent networks, and one which estimates probability mass functions. Finally, we showed how these estimators could be combined to estimate time dependent PMFs.

One of the main advantages with the max consensus protocol compared to other strategies is that it has the fastest possible convergence time, and it is also very easy to implement, even in networks with unreliable communication through pairwise gossiping communication. It does neither require any global information nor global identities among the nodes, so it is a suitable choice for anonymous networks. The main disadvantage of the max consensus strategy is that it does not converge to the exact value, but is instead based on probabilistic estimation schemes. However, its accuracy can be improved through increasing the packet sizes.

The size estimation of anonymous dynamical networks extended the static network estimation strategy through a regularization term, which penalizes hypotheses conflicting with a-priori assumptions on the network's behavior. We explicitly considered and characterized the class of quadratic regularization terms, which resulted in a closed form estimator that corresponds to a nonlinear smoother.

Two distributed estimators of empirical PMFs were considered, one based on max consensus and one based on average consensus. The main differences are that the average consensus based estimator converges asymptotically in time, while the max consensus based estimator converges in finite time, but not to the exact value. Also, the accuracy of the max consensus based estimator can be improved through increasing the packet size, which means that it will typically use larger packets than the average consensus based estimator. However, our experiments show that if estimation speed is important, and in particular if there is also a spatial correlation among the nodes' initial values, then the max consensus based estimator outperforms the average consensus based estimator.

The algorithms has indeed been derived using some simplifying assumptions, in particular reliable communications and infinite numerical precision. We have however shown with numerical experiments that quantization effects seem to play a minor role, and that representing the numbers with just a few bits is enough even for networks of hundreds of agents. Furthermore, we remark that if the max consensus protocol does not reach consensus, it would imply that the network is, at least temporarily, not strongly connected. The estimation process would still succeed in the estimation of the reachable subset of nodes, which is an interesting extension.

Chapter 3

Faulty Nodes in Consensus Protocols

"Don't find fault, find a remedy."
— Henry Ford

In this chapter we consider a leader-follower multi-agent system. We present a framework where the follower agents are tracking an antagonistic leader rather than a cooperative leader. Related are the classical results of game theory for the so-called pursuit-evasion game (Başar and Olsder, 1999; Ho *et al.*, 1965), where the considered game consists of a pursuer who aims to capture the evader while the evader tries to prevent being captured.

A motivating example is resilience of a multi-agent system against faults, for example considered in power networks as robustness against any single point of failure. There has been much work on trying to detect failures (Guo *et al.*, 2012; Shames *et al.*, 2012; Teixeira *et al.*, 2012), but here we focus on the effects of the fault, when the leader is using positive feedback, instead of negative feedback.

The outline of this chapter is as follows: In section 3.1 we introduce the tracking problem of the faulty agent. In section 3.2 we present and analyze the convergence theorems for this multi-agent system, which we further illustrate with numerical simulations in section 3.3. Section 3.4 concludes this chapter.

3.1 Problem Statement

We consider a multi-agent system, consisting of one faulty agent (*leader*) and n follower agents ($n+1$ agents in total). The set of agents is denoted by $\mathcal{V} = \{0, 1, \ldots, n\}$, where 0 is the faulty agent, and the remaining follower agents are denoted by $\mathcal{V}^F \doteq \{1, \ldots, n\}$.

The interaction topology of the multi-agent network is modeled as a switching topology, which means that the undirected graph $\mathcal{G}_{\sigma(t)}(\mathcal{V}, \mathcal{E}_{\sigma(t)})$ is time dependent, and switches topology between a limited number of graphs. Here, $\sigma : [0, +\infty) \to \mathcal{Q}$ is a piecewise constant function, where $\mathcal{Q}$ is the finite set indicating the possible undirected graphs, see Royle (2001). $\mathcal{G}^F_{\sigma(t)}(\mathcal{V}^F, \mathcal{E}^F_{\sigma(t)})$ denotes the induced commu-

nication graph among the follower agents, i.e., the graph obtained by removing the faulty agent. Let $\mathcal{N}_i(\sigma(t))$ represent the neighbors of agent i in $\mathcal{G}_{\sigma(t)}$, for $i \in \mathcal{V}$ at time t, and $\mathcal{N}_i^F(\sigma(t))$ the set of neighbors in $\mathcal{G}_{\sigma(t)}^F$, for $i \in \mathcal{V}^F$.

Each follower $i \in \mathcal{V}^F$ has an associated state $x_i \in \mathbb{R}$, and the state of the faulty agent is denoted by $y \in \mathbb{R}$. The goal of each follower agent $i \in \mathcal{V}^F$ is to reach consensus with every other agent in $\mathcal{V}$. The evolution of their state $x_i(t)$ is given by

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i^F(\sigma(t))} a_{ij}(x,t)\big(x_j(t) - x_i(t)\big) + b_i(t)\big(y(t) - x_i(t)\big)$$

where the function

$$b_i(t) = \begin{cases} b_*, & \text{if } i \text{ is connected to the faulty agent} \\ 0, & \text{otherwise.} \end{cases}$$

marks whether agent $i \in \mathcal{V}^F$ is connected to the faulty agent, with a given weight $b_* > 0$ and a piecewise continuous function $a_{ij}(x,t) > 0$, $i,j \in \mathcal{V}^F$ describing the weight of the edge between agent i and j.

The faulty agent follows the same algorithm, but has distorted weights, which has resulted in a changed sign. Without loss of generality, we have scaled this weight to be -1. The evolution of the faulty agent's state is given by

$$\dot{y}(t) = \sum_{j \in \mathcal{N}_0(\sigma(t))} -\Big(x_j(t) - y(t)\Big).$$

The overall dynamics for the considered multi-agent systems can then be summarized as:

$$\begin{cases} \dot{y}(t) = \displaystyle\sum_{j \in \mathcal{N}_0(\sigma(t))} -\big(x_j(t) - y(t)\big), \\ \dot{x}_i(t) = \displaystyle\sum_{j \in \mathcal{N}_i^F(\sigma(t))} a_{ij}(x,t)\big(x_j(t) - x_i(t)\big) + b_i(t)\big(y(t) - x_i(t)\big), \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } i = 1, \ldots, n \end{cases} \tag{3.1}$$

Note that, compared to most of the existing leader-follower models (Hong *et al.*, 2006; Shi and Hong, 2009; Tanner *et al.*, 2004), the faulty agent is observing the follower's states and then takes opposed actions in order to escape from being tracked. Let $(x(t), y(t)) = (x_1(t), \ldots, x_n(t), y(t))^T \in \mathbb{R}^{n+1}$ denote the solution to eq. (3.1) when starting from initial value $x(0) \in \mathbb{R}^n$ and $y(0) \in \mathbb{R}$. The interesting question is whether the faulty agent can be tracked, or if it will escape from the followers. To this end, define the tracking measurement

$$\Upsilon(t) = \max_{i=1,\ldots,n} \big|x_i(t) - y(t)\big|$$

We introduce the following notations.

Definition 3.1.

(i) System (3.1) is *trackable* for initial value $x(0) \in \mathbb{R}^n$ and $y(0) \in \mathbb{R}$ if

$$\lim_{t\to\infty} \Upsilon(t) = 0.$$

(ii) System (3.1) is *globally trackable* if it is trackable for all initial values.

(iii) System (3.1) is *escapable* for initial value $x(0) \in \mathbb{R}^n$ and $y(0) \in \mathbb{R}$ if

$$\lim_{t\to\infty} \Upsilon(t) = \infty.$$

3.2 Convergence Results

Here we analyze the convergence properties of the consensus system, first for fixed communication graphs, and then for time dependent communication graphs.

Fixed Graphs

First we focus on time-invariant graphs, with the following assumption.

Assumption 3.1 (Fixed Topology)**.** The communication graph $\mathcal{G}_{\sigma(t)}$ and the weight functions $a_{ij}(t)$, $b_i(t)$ are time-independent.

Hence, in the remainder of this section we will drop the time parameter t from these symbols. Now, introduce the state difference with the faulty agent as $\xi_i(t) = x_i(t) - y(t)$. System (3.1) can then be written as:

$$\dot{\xi}_i = \sum_{j\in\mathcal{N}_i^F} a_{ij}(\xi_j - \xi_i) - b_i\xi_i + \sum_{j\in\mathcal{N}_0} \xi_j \tag{3.2}$$

Let $L_F = D_F - A_F$ be the Laplacian matrix of the follower graph, given by the adjacency matrix $A_F = [a_{ij}] \in \mathbb{R}^{n\times n}$ and degree matrix $D_F = \operatorname{diag}(d_1 \dots d_n)$, where $d_i = \sum_{j=1, j\neq i}^{n} a_{ij}$ is the node degree. Let $B = \operatorname{diag}(b_1, \dots, b_n)$ denote the connection weights from the followers to the faulty agent, and $E = \mathbb{1}e^T$ denote the connections from the faulty agent to the followers, with $e^T = (e_1 \dots e_n) \in \mathbb{R}^n$, where $e_i = 1$ if $i \in \mathcal{N}_0$ and $e_i = 0$ otherwise. By denoting $\xi = (\xi_1, \dots, \xi_n)^T$, we can simplify the system equations (3.2) to give the compact matrix form

$$\dot{\xi} = -G\xi, \tag{3.3}$$

where $G = L_F + B - E$.

Notice that global tracking for System (3.1) is equivalent with stability of the matrix system (3.3). The following theorem follows directly.

Theorem 3.1. *Suppose assumption 3.1 holds, then*

(i) System (3.1) *is globally trackable if and only if* $-G$ *is a Hurwitz matrix.*

(ii) There exists initial values for which System (3.1) *is escapable if and only if* $-G$ *has at least one eigenvalue with strictly positive real part.*

In fact, if $-G$ has an eigenvalue λ with strictly positive real part and the corresponding eigenvector β_λ, then, for every initial value $(x(0), y(0))$ with $x(0) - y(0)\mathbb{1}$ not orthogonal to β_λ, System (3.1) is escapable.

Remark. The boundary case when the system is neither globally trackable nor escapable is if the eigenvalue with largest real part is zero. We will not specifically consider this case.

Although Theorem 3.1 gives a clear description of the trackability of System (3.1), we still need simple conditions which only rely on the structure of the communication graph. The following lemma can be found in Hong *et al.* (2006).

Lemma 3.2. *Suppose assumption 3.1 holds, and* $\mathcal{G}$ *is connected. Then* $L_F + B$ *is a positive definite matrix.*

According to lemma 3.2, we can denote the eigenvalues of $-(L_F + B)$ as $\lambda_n^* \leq \cdots \leq \lambda_1^* < 0$, and then the following theorem holds.

Theorem 3.3. *Suppose assumption 3.1 holds and* $\mathcal{G}$ *is connected. System* (3.1) *is globally trackable if* $\lambda_1^* < -\sqrt{n|\mathcal{N}_0|}$*, where* $|\mathcal{N}_0|$ *represents the number of neighbors of the faulty agent.*

The proof of Theorem 3.3 relies on the following lemma on the perturbation of eigenvalues (Quarteroni *et al.*, 2007):

Lemma 3.4. *Given a matrix* $C \in \mathbb{R}^{n\times n}$*, let* λ *be an eigenvalue to* C*, and* Λ *the matrix of eigenvectors. If* μ *is an eigenvalue of the perturbed matrix* $C + P \in \mathbb{R}^{n\times n}$*, then*

$$\min_{\lambda\in\lambda(C)} \left|\lambda - \mu\right| \leq \|\Lambda\|_2 \cdot \|\Lambda^{-1}\|_2 \cdot \|P\|_2 \tag{3.4}$$

where $\lambda(C)$ *denotes the spectrum of* C*.*

We are now ready to present the proof of Theorem 3.3.

Proof of Theorem 3.3. Let μ be any eigenvalue of $-G = -L_F - B + E$, and apply lemma 3.4 on the matrix $(-L_F - B) + E$. We get

$$\min_{\substack{\lambda_i^* \\ i=1,\ldots,n}} \left|\lambda_i^* - \mu\right| \leq \|E\|_2 \tag{3.5}$$

since we can select eigenvectors of $-L_F - B$ which forms an orthogonal matrix. Moreover, noticing that

$$\|E\|_2 = \sqrt{n|\mathcal{N}_0|},$$

we obtain

$$\mathrm{Re}\,(\mu) \leq \lambda_1^* + \|E\|_2 \leq \lambda_1^* + \sqrt{n|\mathcal{N}_0|} < 0$$

for any eigenvalue μ of $-G$ when $\lambda_1^* < -\sqrt{n|\mathcal{N}_0|}$. □

Theorem 3.3 gives us a sufficient condition for global trackability, and in the next theorem we give a necessary condition for global trackability.

Theorem 3.5. *Suppose assumption 3.1 holds and $\mathcal{G}$ is connected. If System* (3.1) *is globally trackable, then the fault threshold $b_* \geq |\mathcal{N}_0|$ is satisfied.*

Proof of Theorem 3.5. Theorem 3.1 implies that, for all eigenvalues,

$$\mathrm{Re}\,(\lambda(L_F + B - E)) \geq 0$$

if the system is globally trackable. But if the system is trackable, then so is also the system where $b_i = b_*$, $\forall i \in \mathcal{V}^F$, hence

$$\mathrm{Re}\,(\lambda(L_F - E)) \geq -b_*.$$

Notice that $\mathbb{1}$ is an eigenvector to E with eigenvalue $|\mathcal{N}_0|$, but also an eigenvector of L_F with eigenvalue 0. Thus, we have in particular for this eigenvector that

$$0 - |\mathcal{N}_0| \geq -b_* \quad \Rightarrow \quad b_* \geq |\mathcal{N}_0|$$

□

Time Dependent Graphs

Now we turn our attention to dynamical graphs, but follow the tradition of switching dynamics (Jadbabaie *et al.*, 2003; Lin *et al.*, 2007; Shi and Hong, 2009). The following assumption gives a bound on the switching signal $\sigma(t) : [0, +\infty) \to \mathcal{Q}$ between graph topologies.

Assumption 3.2 (Dwell Time)**.** There exists a lower bound $\tau_D > 0$ between two switching instances of $\sigma(t)$.

We also impose bounds on the weight functions, $a_{ij}(x, t)$:

Assumption 3.3 (Weights Rule)**.** There exists a lower bound $a_* > 0$ and an upper bound $a^* > 0$ such that

$$a_* \leq a_{ij}(x, t) \leq a^*, \quad t \in \mathbb{R}_+,\ x \in \mathbb{R}^n.$$

The joint graph of $\mathcal{G}_{\sigma(t)}$ over a time interval $[t_1, t_2)$ with $t_1 < t_2 \leq +\infty$ is denoted as $\mathcal{G}([t_1, t_2)) = \cup_{t\in[t_1,t_2)}\mathcal{G}(t) = (\mathcal{V}, \cup_{t\in[t_1,t_2)}\mathcal{E}_{\sigma(t)})$. The joint follower graph is similarly defined as $\mathcal{G}^F([t_1, t_2))$. Another assumption is given on the connectivity of the joint communication graphs:

Assumption 3.4 (Joint Connectivity)**.** There exists $T > 0$ such that both graphs $\mathcal{G}([t, t+T))$ and $\mathcal{G}^F([t, t+T))$ are connected for all t.

For time dependent communication graphs, we have the following main results.

Theorem 3.6. *Suppose assumptions 3.2 to 3.4 hold, then there exists initial values for which System* (3.1) *is escapable if* $b_* < 1$.

Theorem 3.7. *Suppose assumptions 3.2 to 3.4 hold. System* (3.1) *is globally trackable if the system parameters* b_*, a_*, a^*, T, τ_D *satisfy*

$$0 < \left(e^{n(n+1)T_0} - w_*\left(\varrho_0 e^{-(n^2-1)a^*T_0}\right)^n\right) < 1 \tag{3.6}$$

where $T_0 = T + \tau_D$ *and*

$$w_* = \frac{b_* + (e^{\tau_D} - 1)\left(e^{nT_0} - 1\right)\left(e^{-b_*T_0} - 1\right)}{(n-1)a^* + b_*} \cdot e^{-(n-1)a^*(n+1)T_0};$$
$$\varrho_0 = \frac{\left(1 - e^{-((n-2)a^* + a_*)\tau_D}\right)a_*}{(n-2)a^* + a_*}.$$

Notice that parameters meeting the requirement of Theorem 3.7 can always be found as long as we choose T_0^{-1} and a^* sufficiently large. In the rest of this subsection, we first establish several lemmas which are useful for the convergence analysis, and then prove Theorems 3.6 and 3.7.

Key Lemmas

Since we are analyzing piecewise continuous functions, we recall the Dini derivatives. Let a and b $(> a)$ be two real numbers, and consider a function $h : (a, b) \to \mathbb{R}$ together with a point $t \in (a, b)$. The upper Dini derivative of h at t is defined as

$$D^+h(t) = \limsup_{s\to 0^+} \frac{h(t+s) - h(t)}{s}.$$

It is well known that when h is continuous on (a, b), then h is non-increasing on (a, b) if and only if $D^+h(t) \leq 0$ for every $t \in (a, b)$ (Clarke *et al.*, 1998). The next result is given for the calculation of the Dini derivative (Danskin, 1966; Lin *et al.*, 2007).

Lemma 3.8. *Let $V_i(t,x) : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ $(i = 1,\dots,n)$ be C^1 and $V(t,x) = \max_{i=1,\dots,n} V_i(t,x)$. If $\mathcal{I}(t) = \{i \in \{1,2,\dots,n\} : V(t,x(t)) = V_i(t,x(t))\}$ is the set of indices where the maximum is attained at time t, then*

$$D^+V(t,x(t)) = \max_{i\in\mathcal{I}(t)} \dot{V}_i(t,x(t)).$$

Introduce the minimum and maximum state value among the followers as

$$m(t) = \min_{i\in\mathcal{V}^F} x_i(t);$$
$$M(t) = \max_{i\in\mathcal{V}^F} x_i(t).$$

The following lemma holds for $m(t)$ and $M(t)$.

Lemma 3.9. *Suppose $y(t) \in [m(t), M(t)]$ for $t \geq 0$. Then $D^+m(t) \geq 0$ and $D^+M(t) \leq 0$ for all $t \geq 0$.*

Proof. We only prove $D^+m(t) \geq 0$, the other case follows by a symmetric argument. Denoting $\mathcal{I}_0(t)$ as the index set consisting of all the follower agents which reaches the minimal value at time t. Let $i \in \mathcal{I}_0(t)$, then we have

$$\dot{x}_i(t) = \sum_{j\in\mathcal{N}_i^F(\sigma(t))} a_{ij}(x,t)\big(x_j(t) - x_i(t)\big) + b_i(t)\big(y(t) - x_i(t)\big) \geq 0$$

because $x_j(t) \geq x_i(t) = m(t)$ for all $j \in \mathcal{N}_i^F(\sigma(t))$ and $y(t) \geq x_i(t) = m(t)$. Therefore, according to lemma 3.8,

$$D^+m(t) = \max_{i\in\mathcal{I}_0(t)} \dot{x}_i(t) \geq 0.$$

□

The following lemma indicates that System (3.1) is trackable when $y(t) \in [m(t), M(t)]$ for $t \geq 0$.

Lemma 3.10. *Suppose assumptions 3.2 to 3.4 hold, and $y(t) \in [m(t), M(t)]$ for $t \geq 0$. Then System (3.1) is trackable.*

Proof. The idea behind the proof is to consider an agent attaining the minimal value $m(t)$, and then, since the network is jointly connected over a period of time T, we can find a path of influence in the joint network which will bound the state even for the agents attaining $M(t)$.

To this end, take any $t_0 \geq 0$. Suppose agent $i_0 \in \mathcal{V}^F$ attains the minimal value at time t_0, i.e., $x_{i_0}(t_0) = m(t_0)$. Based on lemma 3.9, we have

$$m(t) \geq m(t_0);$$
$$M(t) \leq M(t_0)$$

for all $t \geq t_0$. As a result, with assumption 3.3, if $y(t) \in [m(t), M(t)]$ for $t \geq 0$, we obtain

$$\begin{aligned}
\dot{x}_{i_0}(t) &= \sum_{j \in \mathcal{N}_{i_0}^F(\sigma(t))} a_{i_0 j}(x,t)\big(x_j(t) - x_{i_0}(t)\big) + b_i(t)\big(y(t) - x_{i_0}(t)\big) \\
&\leq \left(\sum_{j \in \mathcal{N}_{i_0}^F(\sigma(t))} a_{i_0 j}(x,t) \right) \Big(M(t) - x_{i_0}(t) \Big) + b_i(t) \Big(M(t) - x_{i_0}(t) \Big) \\
&\leq (n-1)a^* \big(M(t_0) - x_{i_0}(t) \big) + b_* \big(M(t_0) - x_{i_0}(t) \big) \\
&= -\big((n-1)a^* + b_* \big) \big(x_{i_0}(t) - M(t_0) \big), \quad t \geq t_0.
\end{aligned} \tag{3.7}$$

Thus, by Grönwall's inequality, we further conclude that

$$\begin{aligned}
x_{i_0}(t) &\leq e^{-\left((n-1)a^* + b_*\right)(t-t_0)} x_{i_0}(t_0) + \left(1 - e^{-\left((n-1)a^* + b_*\right)(t-t_0)} \right) M(t_0) \\
&= e^{-\left((n-1)a^* + b_*\right)(t-t_0)} m(t_0) + \left(1 - e^{-\left((n-1)a^* + b_*\right)(t-t_0)} \right) M(t_0),
\end{aligned}$$

for $t \geq t_0$, which implies

$$\begin{aligned}
x_{i_0}(t) &\leq e^{-\left((n-1)a^* + b_*\right)(n-1)T_0} m(t_0) + \left(1 - e^{-\left((n-1)a^* + b_*\right)(n-1)T_0} \right) M(t_0) \\
&= d_0 m(t_0) + (1 - d_0) M(t_0) \\
&\doteq \phi_0
\end{aligned} \tag{3.8}$$

for all $t \in [t_0, t_0 + (n-1)T_0]$, where $d_0 = e^{-\left((n-1)a^* + b_*\right)(n-1)T_0}$ and $T_0 = T + \tau_D$.

Now we have bounded the first agent's state in the network, so we will continue with the next one. According to the joint connectivity assumption 3.4, there exists at least one agent i_1 such that i_1 is connected to i_0 in the graph $\mathcal{G}_{\sigma(\hat{t}_1)}$ for some $\hat{t}_1 \in [t_0, t_0 + T)$.

First, if there exists some $s \in [\hat{t}_1, \hat{t}_1 + \tau_D]$ such that

$$x_{i_1}(s) \leq \phi_0 = d_0 m(t_0) + (1 - d_0) M(t_0).$$

then we have immediately bounded x_{i_1} also. Thus, we assume that for all $t \in$

$[\hat{t}_1, \hat{t}_1 + \tau_D]$, it holds that $x_{i_1}(t) \geq \phi_0$. Then we see from eq. (3.8) that

$$\begin{aligned}
\dot{x}_{i_1}(t) &= \sum_{j \in \mathcal{N}_{i_1}^F(\sigma(t))} a_{i_1 j}(x,t)\big(x_j(t) - x_{i_1}(t)\big) + b_i(t)\big(y(t) - x_{i_1}(t)\big) \\
&\leq a_{i_1 i_0}(t)\big(x_{i_0}(t) - x_{i_1}(t)\big) + b_i(t)\Big(M(t) - x_{i_1}(t)\Big) \\
&\quad + \Big(M(t) - x_{i_1}(t)\Big) \sum_{j \in \mathcal{N}_{i_1}^F(\sigma(t)) \setminus \{i_0\}} a_{ij}(x,t) \\
&\leq a_*\big(\phi_0 - x_{i_1}(t)\big) + b_*\big(M(t_0) - x_{i_1}(t)\big) + (n-2)a^*\big(M(t_0) - x_{i_1}(t)\big) \\
&= -\big((n-2)a^* + b_* + a_*\big)\left(x_{i_1}(t) - \frac{M(t_0)\big((n-2)a^* + b_*\big) + a_*\phi_0}{(n-2)a^* + b_* + a_*}\right),
\end{aligned}$$

for $t \in [\hat{t}_1, \hat{t}_1 + \tau_D]$. This implies

$$\begin{aligned}
x_{i_1}(\hat{t}_1 + \tau_D) &\leq \delta_0 x_{i_1}(\hat{t}_1) + \big(1 - \delta_0\big)\left(\frac{M(t_0)\big((n-2)a^* + b_*\big) + a_*\phi_0}{(n-2)a^* + b_* + a_*}\right) \\
&\leq \delta_0 M(t_0) + \big(1 - \delta_0\big)\left(\frac{M(t_0)\big((n-2)a^* + b_*\big) + a_*\phi_0}{(n-2)a^* + b_* + a_*}\right) \\
&= \frac{a_*(1-\delta_0)d_0}{(n-2)a^* + b_* + a_*} m(t_0) + \left(1 - \frac{a_*(1-\delta_0)d_0}{(n-2)a^* + b_* + a_*}\right) M(t_0) \\
&\doteq (1-\delta_0)d_0\lambda_0 m(t_0) + \Big(1 - (1-\delta_0)d_0\lambda_0\Big)M(t_0),
\end{aligned}$$

where

$$\begin{aligned}
\delta_0 &\doteq e^{-\left((n-2)a^* + b_* + a_*\right)\tau_D}; \\
\lambda_0 &\doteq \frac{a_*}{(n-2)a^* + b_* + a_*}.
\end{aligned}$$

Consequently, in either case, there exists a $\tilde{t}_1 \in [t_0, t_0 + T_0]$ such that we can bound x_{i_1} with

$$x_{i_1}(\tilde{t}_1) \leq (1-\delta_0)d_0\lambda_0 m(t_0) + \Big(1 - (1-\delta_0)d_0\lambda_0\Big)M(t_0).$$

Noticing that inequality (3.7) also holds for i_1, we can similarly obtain

$$x_{i_1}(t) \leq (1-\delta_0)d_0^2\lambda_0 m(t_0) + \Big(1 - (1-\delta_0)d_0^2\lambda_0\Big)M(t_0)$$

for all $t \in [t_0 + T_0, t_0 + (n-1)T_0]$.

We can now proceed with the analysis at the time intervals $[t_0 + T_0), \ldots, [t_0 + (n-2)T_0, t_0 + (n-1)T_0)$, and by the joint connectivity assumption 3.4, we can find agents $i_2, i_3, \ldots, i_{n-1} \in \mathcal{V}^F$ such that

$$x_{i_s}(t) \leq \big[(1-\delta_0)d_0\lambda_0\big]^s d_0 m(t_0) + \Big(1 - \big[(1-\delta_0)d_0\lambda_0\big]^s d_0\Big)M(t_0)$$

for all $t \in [t_0 + sT_0, t_0 + (n-1)T_0]$. Notice that at the latest when $s = n-1$ we have a bound that is also valid for $M(t)$,

$$\begin{aligned} M\big(t_0 + (n-1)T_0\big) \leq & \big[(1-\delta_0)d_0\lambda_0\big]^{n-1} d_0 m(t_0) \\ & + \Big(1 - \big[(1-\delta_0)d_0\lambda_0\big]^{n-1} d_0\Big) M(t_0). \end{aligned}$$

Thus, according to lemma 3.9, we eventually obtain

$$\begin{aligned} & M\big(t_0 + (n-1)T_0\big) - m\big(t_0 + (n-1)T_0\big) \\ & \qquad \leq \Big(1 - \big[(1-\delta_0)d_0\lambda_0\big]^{n-1} d_0\Big)\Big(M(t_0) - m(t_0)\Big). \end{aligned} \tag{3.9}$$

Since t_0 is chosen arbitrarily, eq. (3.9) implies

$$\lim_{t\to\infty} \Big|M\big(t\big) - m\big(t\big)\Big| = 0,$$

and thus

$$\lim_{t\to\infty} \xi(t) = 0$$

as long as $y(t) \in [m(t), M(t)]$ for $t \geq 0$. This completes the proof. □

Let us now proceed to the general case, when there exists some time $t_* \geq 0$ such that $y(t_*) \notin [m(t_*), M(t_*)]$.

Lemma 3.11.

(i) If there exists some $t_* \geq 0$ such that $y(t_*) > M(t_*)$, then $y(t) > M(t)$ for all $t \geq t_*$.

(ii) If there exists some $t_* \geq 0$ such that $y(t_*) < m(t_*)$, then $y(t) < m(t)$ for all $t \geq t_*$.

Proof. We focus on proving (i), since (ii) holds from a symmetric argument.

The differential equation (3.1) is piecewise continuous, and since $y(t_*) > M(t_*)$, there exists an $\varepsilon > 0$ such that $y(t) > M(t)$ for all $t \in [t_*, t_* + \varepsilon)$. Consequently, by a similar analysis as in lemma 3.9, we have

$$\begin{aligned} D^+M(t) &\leq b_*\big(y(t) - M(t)\big); \\ D^+y(t) &\geq 0 \end{aligned}$$

for $t \in [t_*, t_* + \varepsilon]$. This leads to

$$y(t_* + \varepsilon) - M(t_* + \varepsilon) > e^{-b_*\varepsilon}\big(y(t_*) - M(t_*)\big) > 0. \tag{3.10}$$

Then eq. (3.10) implies that $y(t) > M(t)$ for all $t \geq t_*$. □

We are now ready to prove the main theorems for time dependent networks.

Proof of Theorem 3.6. The intuitive explanation of the proof is that if the attractive force from the followers (b_*) is lower than the repulsive force (-1) from the faulty agent, then the system will be escapable. Consider initial values $(x(0), y(0))$ such that $y(0) > M(0)$. Then, lemma 3.11 implies that $y(t) > M(t)$ for all $t > 0$. Therefore, when there are no follower agents connected to the faulty agent, we have

$$\begin{aligned} D^+M(t) &\leq 0; \\ D^+y(t) &= 0, \end{aligned}$$

and when at least one follower agent is connected to the faulty agent

$$\begin{aligned} D^+M(t) &\leq b_*\big(y(t) - M(t)\big); \\ D^+y(t) &\geq y(t) - M(t). \end{aligned}$$

This leads to

$$D^+\big[y(t) - M(t)\big] \geq \begin{cases} 0, & \text{if no follower is connected to the faulty} \\ & \text{agent at time } t \\ (1 - b_*)\big[y(t) - M(t)\big], & \text{otherwise.} \end{cases}$$

It is straightforward to see that $\lim_{t\to\infty}\big[y(t) - M(t)\big] = \infty$ since the assumptions guarantee that there will be some follower agent connected to the faulty agent sufficiently often. □

Proof of Theorem 3.7. Based on lemma 3.11, we just need to prove Theorem 3.7 for the cases when $\exists\, t_* \geq 0$ such that $y(t_*) > M(t_*)$ or $y(t_*) < m(t_*)$. We focus on the first case, since the proof for the second case can be obtained by a symmetric argument.

Suppose $y(t_*) > M(t_*)$ for some $t_* > 0$. Then, lemma 3.11 suggests that $y(t) > M(t)$ for all $t \geq t_*$. The proof follows the same ideas as in the proof of lemma 3.10, to create a chain of influence between $y(t)$ and $m(t)$.

Choose $t_0 \geq t_*$, in our first step we bound $y(t)$. Similar to lemma 3.9, since $y(t) > M(t)$, we have $D^+m(t) \geq 0$ for all $t \geq t_*$. Noticing that

$$\begin{aligned} \dot{y}(t) &= \sum_{j\in\mathcal{N}_0(\sigma(t))} \big(y(t) - x_j(t)\big) \\ &\leq n\big(y(t) - m(t)\big) \\ &\leq n\big(y(t) - m(t_0)\big) \end{aligned}$$

for all $t \geq t_0$, we obtain

$$y(t) \leq e^{n(t-t_0)}y(t_0) + \big(1 - e^{n(t-t_0)}\big)m(t_0), \quad t \geq t_0. \tag{3.11}$$

This implies

$$y(t) \le e^{n(n+1)T_0} y(t_0) + \big(1 - e^{n(n+1)T_0}\big) m(t_0), \quad t \in [t_0, t_0 + (n+1)T_0],$$

where $T_0 = T + \tau_D$.

On the other hand, (3.11) implies

$$\begin{aligned} D^+ M(t) &\le b_* \big(y(t) - M(t)\big) \\ &\le -b_* \Big(M(t) - e^{nT_0} y(t_0) - \big(1 - e^{nT_0}\big) m(t_0)\Big), \end{aligned}$$

for $t \in [t_0, t_0 + T_0]$, which yields

$$M(t) \le e^{-b_* T_0} M(t_0) + \big(1 - e^{-b_* T_0}\big) \cdot \Big(e^{nT_0} y(t_0) + \big(1 - e^{nT_0}\big) m(t_0)\Big), \quad t \in [t_0, t_0 + T_0]. \tag{3.12}$$

Since $\mathcal{G}\big([t_0, t_0 + T)\big)$ is connected, there exists $\hat{t}_1 \in [t_0, t_0 + T)$ such that the faulty agent is connected to some follower agent at time $\hat{t}_1$. As a result, (3.12) leads to

$$\begin{aligned} \dot{y}(t) &= \sum_{j \in \mathcal{N}_0(\sigma(t))} \big(y(t) - x_j(t)\big) \\ &\ge y(t) - M(t) \\ &\ge y(t) - e^{-b_* T_0} M(t_0) - \big(1 - e^{-b_* T_0}\big) \\ &\quad \cdot \Big(e^{nT_0} y(t_0) + \big(1 - e^{nT_0}\big) m(t_0)\Big) \end{aligned} \tag{3.13}$$

for $t \in [\hat{t}_1, \hat{t}_1 + \tau_D]$ with $\hat{t}_1 + \tau_D \le T_0$, which implies

$$\begin{aligned} y(\hat{t}_1 + \tau_D) \ge\ & e^{\tau_D} y(t_0) + (1 - e^{\tau_D}) \cdot \\ & \Big[e^{-b_* T_0} M(t_0) + \big(1 - e^{-b_* T_0}\big)\Big(e^{nT_0} y(t_0) + \big(1 - e^{nT_0}\big) m(t_0)\Big)\Big] \end{aligned} \tag{3.14}$$

Let $0 < \chi \le 1$ be a constant satisfying $y(t_0) - M(t_0) = \chi\big[y(t_0) - m(t_0)\big]$. Noticing that $y(t)$ is strictly increasing for $t > t_*$, we see from (3.14) that

$$\begin{aligned} y(t) &\ge y(\hat{t}_1 + \tau_D) \\ &\ge y(t_0) + \Big(e^{\tau_D} - 1\Big) \cdot \Big(y(t_0) - m(t_0)\Big) \\ &\quad \cdot \Big(\big(e^{nT_0} - 1\big)\big(e^{-b_* T_0} - 1\big) + \chi e^{-b_* T_0}\Big) \\ &= y(t_0) + p_0 \Big(y(t_0) - m(t_0)\Big), \end{aligned} \tag{3.15}$$

for all $t \ge t_0 + T_0$ after some simple algebra, where

$$p_0 \doteq \big(e^{\tau_D} - 1\big)\big(\big(e^{nT_0} - 1\big)\big(e^{-b_* T_0} - 1\big) + \chi e^{-b_* T_0}\big). \tag{3.16}$$

Next, we give a lower bound for $m\big(t_0+(n+1)T_0\big)$. Since $\mathcal{G}\big([t_0+T_0,t_0+2T)\big)$ is connected, there exists at least one follower agent $i_0\in\mathcal{V}^F$ and $\hat{t}_2\in[t_0+T_0,t_0+2T)$ such that i_0 is connected to the faulty agent at time $\hat{t}_2$. Therefore, with (3.15), we have

$$\begin{aligned}\dot{x}_{i_0}(t)&=\sum_{j\in\mathcal{N}_{i_0}^F(\sigma(t))}a_{i_0j}(x,t)\big(x_j(t)-x_{i_0}(t)\big)+b_*\big(y(t)-x_{i_0}(t)\big)\\&\geq(n-1)a^*\big(m(t_0)-x_{i_0}(t)\big)\\&\quad+b_*\Big(y(t_0)+p_0\big(y(t_0)-m(t_0)\big)-x_{i_0}(t)\Big),\end{aligned}$$

for $t\in[\hat{t}_2,\hat{t}_2+\tau_D]$, which implies

$$\begin{aligned}x_{i_0}(\hat{t}_2+\tau_D)&\geq e^{-((n-1)a^*+b_*)\tau_D}x_{i_0}(\hat{t}_2)+\big(1-e^{-((n-1)a^*+b_*)\tau_D}\big)\\&\quad\cdot\frac{(n-1)a^*m(t_0)+b_*\big[y(t_0)+p_0\big(y(t_0)-m(t_0)\big)\big]}{(n-1)a^*+b_*}\\&\geq e^{-((n-1)a^*+b_*)\tau_D}m(t_0)+\big(1-e^{-((n-1)a^*+b_*)\tau_D}\big)\\&\quad\cdot\frac{(n-1)a^*m(t_0)+b_*\big[y(t_0)+p_0\big(y(t_0)-m(t_0)\big)\big]}{(n-1)a^*+b_*}\\&=\frac{b_*+p_0}{(n-1)a^*+b_*}y(t_0)+\Big(1-\frac{b_*+p_0}{(n-1)a^*+b_*}\Big)m(t_0).\end{aligned}$$

Next, for $t\in[\hat{t}_2+\tau_D,t_0+(n+1)T_0]$, we have

$$\begin{aligned}\dot{x}_{i_0}(t)&=\sum_{j\in\mathcal{N}_{i_0}^F(\sigma(t))}a_{i_0j}(x,t)\big(x_j(t)-x_{i_0}(t)\big)+b_i(t)\big(y(t)-x_{i_0}(t)\big)\\&\geq(n-1)a^*\big(m(t_0)-x_{i_0}(t)\big),\end{aligned}$$

and thus,

$$\begin{aligned}x_{i_0}(t)&\geq e^{-(n-1)a^*(n+1)T_0}x_{i_0}(\hat{t}_2+\tau_D)\\&\quad+\big(1-e^{-(n-1)a^*(n+1)T_0}\big)m(t_0)\\&\geq\frac{b_*+p_0}{(n-1)a^*+b_*}e^{-(n-1)a^*(n+1)T_0}y(t_0)\\&\quad+\Big(1-\frac{(b_*+p_0)e^{-(n-1)a^*(n+1)T_0}}{(n-1)a^*+b_*}\Big)m(t_0)\\&\doteq w_0y(t_0)+(1-w_0)m(t_0)\end{aligned}$$

for all $t\in[t_0+2T_0,t_0+(n+1)T_0]$, where $w_0=\frac{b_*+p_0}{(n-1)a^*+b_*}e^{-(n-1)a^*(n+1)T_0}$.

Since $\mathcal{G}\big([t_0+2T_0,t_0+2T_0+T)\big)$ is connected, there exists at least one follower agent $i_1\in\mathcal{V}^F$ and time $\hat{t}_3\in[t_0+2T_0,t_0+2T_0+T)$ such that i_1 is connected to

the faulty agent, or to the follower agent i_0, at time $\hat{t}_3$.

$$\begin{aligned}\dot{x}_{i_1}(t) &= \sum_{j\in\mathcal{N}_{i_1}^F(\sigma(t))} a_{i_1 j}(x,t)\big(x_j(t)-x_{i_1}(t)\big) + b_i(t)\big(y(t)-x_{i_1}(t)\big)\\ &\geq a_{i_1 i_0}(t)\big(x_{i_0}(t)-x_{i_1}(t)\big) + \Big(M(t)-x_{i_1}(t)\Big)\sum_{j\in\mathcal{N}_{i_1}^F(\sigma(t))\setminus\{i_0\}} a_{ij}(x,t)\\ &\geq a_*\big(w_0 y(t_0)+(1-w_0)m(t_0)-x_{i_1}(t)\big) + (n-2)a^*\big(m(t_0)-x_{i_1}(t)\big),\end{aligned}$$

for $t\in[\hat{t}_3,\hat{t}_3+\tau_D]$, where we assume $x_{i_1}(t)\leq w_0y(t_0)+(1-w_0)m(t_0), t\in[\hat{t}_3,\hat{t}_3+\tau_D]$, without loss of generality. As a result, we have

$$\begin{aligned}x_{i_1}(\hat{t}_3+\tau_D) &\geq e^{-((n-2)a^*+a_*)\tau_D}m(t_0) + \big(1-e^{-((n-2)a^*+a_*)\tau_D}\big)\\ &\quad\cdot\frac{a_*\big(w_0y(t_0)+(1-w_0)m(t_0)\big)+(n-2)a^*m(t_0)}{(n-2)a^*+a_*}\\ &= \frac{\big(1-e^{-((n-2)a^*+a_*)\tau_D}\big)a_*w_0}{(n-2)a^*+a_*}y(t_0)\\ &\quad+\Big(1-\frac{\big(1-e^{-((n-2)a^*+a_*)\tau_D}\big)a_*w_0}{(n-2)a^*+a_*}\Big)m(t_0)\\ &\doteq \varrho_0w_0y(t_0)+(1-\varrho_0w_0)m(t_0),\end{aligned}$$

where $\varrho_0=\frac{\big(1-e^{-((n-2)a^*+a_*)\tau_D}\big)a_*}{(n-2)a^*+a_*}$. This immediately implies

$$x_{i_1}(t)\geq\varrho_0w_0e^{-(n-1)a^*(n+1)T_0}y(t_0)+\Big(1-\varrho_0w_0e^{-(n-1)a^*(n+1)T_0}\Big)m(t_0)$$

for all $t\in[t_0+3T_0,t_0+(n+1)T_0]$.

Continuing the analysis for the remaining follower agents $i_2,\ldots,i_n$ eventually results in the bound

$$\begin{aligned}x_i\big(t_0+(n+1)T_0\big) &\geq w_0\big(\varrho_0e^{-(n-1)a^*(n+1)T_0}\big)^n y(t_0)\\ &\quad+\Big(1-w_0\big(\varrho_0e^{-(n-1)a^*(n+1)T_0}\big)^n\Big)m(t_0)\end{aligned}$$

for all $i=1\ldots,n$, and thus

$$\begin{aligned}m\big(t_0+(n+1)T_0\big) &\geq w_0\big(\varrho_0e^{-(n-1)a^*(n+1)T_0}\big)^n y(t_0)\\ &\quad+\Big(1-w_0\big(\varrho_0e^{-(n-1)a^*(n+1)T_0}\big)^n\Big)m(t_0).\end{aligned}\tag{3.17}$$

As a result of eqs. (3.11) and (3.17) we get

$$\begin{aligned}\Big[y\big(t_0+(n+1)T_0\big)-m\big(t_0+(n+1)T_0\big)\Big]&\\ \leq e^{n(n+1)T_0}y(t_0)+\big(1-e^{n(n+1)T_0}\big)m(t_0)&\\ -w_0\big(\varrho_0 e^{-(n-1)a^*(n+1)T_0}\big)^n y(t_0)&\\ -\Big(1-w_0\big(\varrho_0 e^{-(n-1)a^*(n+1)T_0}\big)^n\Big)m(t_0)&\\ =\Big(e^{n(n+1)T_0}-w_0\big(\varrho_0 e^{-(n^2-1)a^*T_0}\big)^n\Big)\cdot\Big[y\big(t_0\big)-m\big(t_0\big)\Big].&\end{aligned} \tag{3.18}$$

By denoting $\Psi(t)=y\big(t\big)-m\big(t\big)$, eq. (3.18) can be written as

$$\Psi\big(t_0+(n+1)T_0\big)\leq\Big(e^{n(n+1)T_0}-w_0\big(\varrho_0 e^{-(n^2-1)a^*T_0}\big)^n\Big)\Psi(t_0) \tag{3.19}$$

for all $t_0\geq t_*$. According to the definition of p_0 in (3.16), w_0 increases as long as χ increases, and then we see from eq. (3.19) that

$$\Psi\big(t_0+(n+1)T_0\big)\leq\Big(e^{n(n+1)T_0}-w_*\big(\varrho_0 e^{-(n^2-1)a^*T_0}\big)^n\Big)\Psi(t_0)$$

where

$$w_*=\frac{b_*+\big(e^{\tau_D}-1\big)\big(e^{nT_0}-1\big)\big(e^{-b_*T_0}-1\big)}{(n-1)a^*+b_*}\cdot e^{-(n-1)a^*(n+1)T_0}.$$

It is now clear that when the parameter condition holds,

$$0<\Big(e^{n(n+1)T_0}-w_*\big(\varrho_0 e^{-(n^2-1)a^*T_0}\big)^n\Big)<1,$$

the system will be globally trackable. □

Remark. Systems with multiple faulty agents will never be globally trackable in our framework. A trivial example can be constructed by placing all follower agents symmetrically distributed between two faulty agents, where the two faulty agents will then diverge.

3.3 Illustration of Convergence Results

In this section we will explore the system's convergence properties through numerical simulations. First we evaluate the states of the agents in a system consisting of 5 agents (the faulty agent and four follower agents), when the communication topology is a fixed line graph, see fig. 3.1. In figs. 3.2 to 3.4 we simulate the system with different b_* selected as 3, 4 and 5 respectively. In the first fig. 3.2, where $b_*<n$, the state errors are diverging, and we thus have an escapable system. In

Figure 3.1: System configuration with a line graph topology. The red node denotes the faulty agent, while the blue nodes denote the followers.

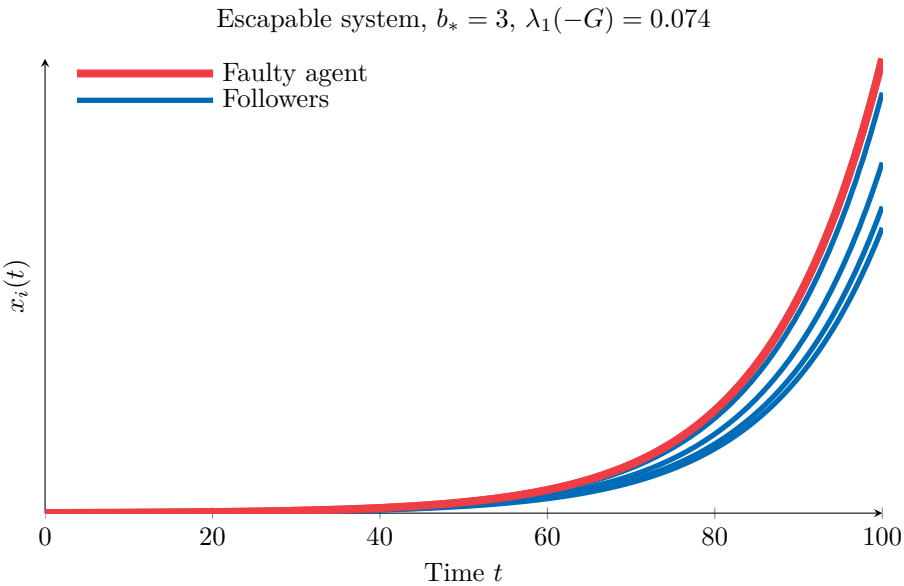


Figure 3.2: Simulation of a multi agent system with a line topology consisting of 5 agents, $n = 4$ and $b_* = 3$.

fig. 3.3, where b_* is increased to $b_* = n$, the state error remains constant, which is the boundary case when the largest eigenvalue is zero, and the system is neither escapable nor trackable. Finally, in fig. 3.4, where $b_* > n$, the state errors are diminishing to zero, and the eigenvalues are all smaller than zero, so this system is globally trackable.

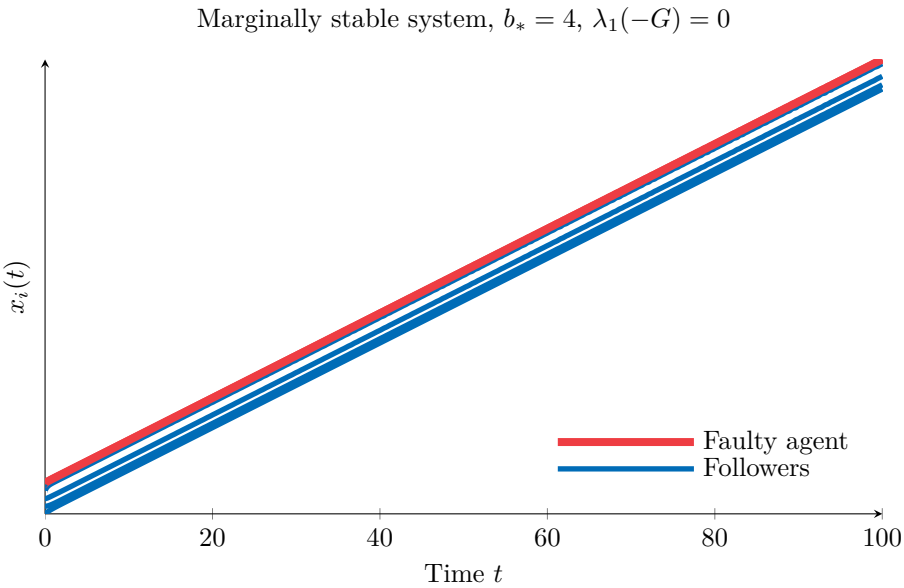


Figure 3.3: Simulation of a multi agent system with a line topology consisting of 5 agents, $n = 4$ and $b_* = 4$.

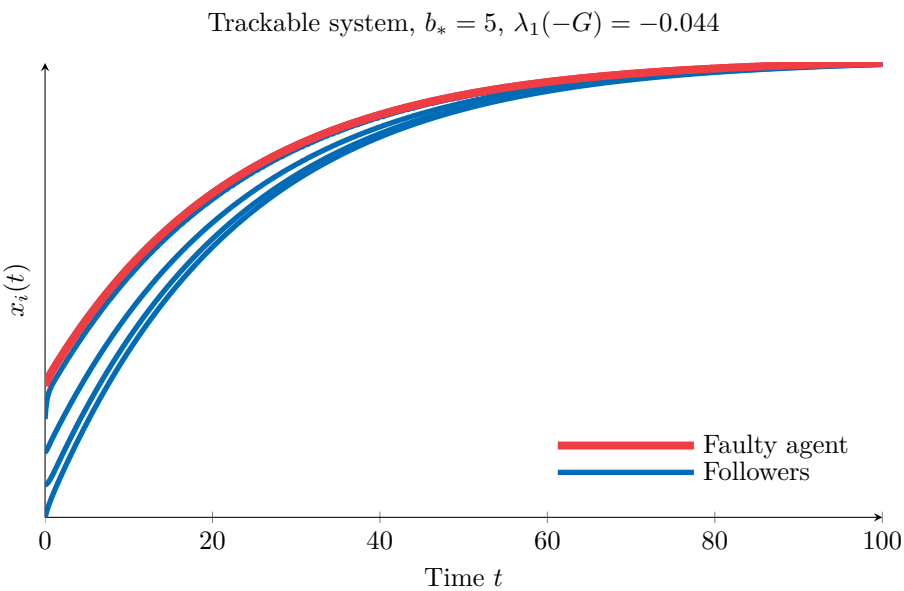


Figure 3.4: Simulation of a multi agent system with a line topology consisting of 5 agents, $n = 4$ and $b_* = 5$.

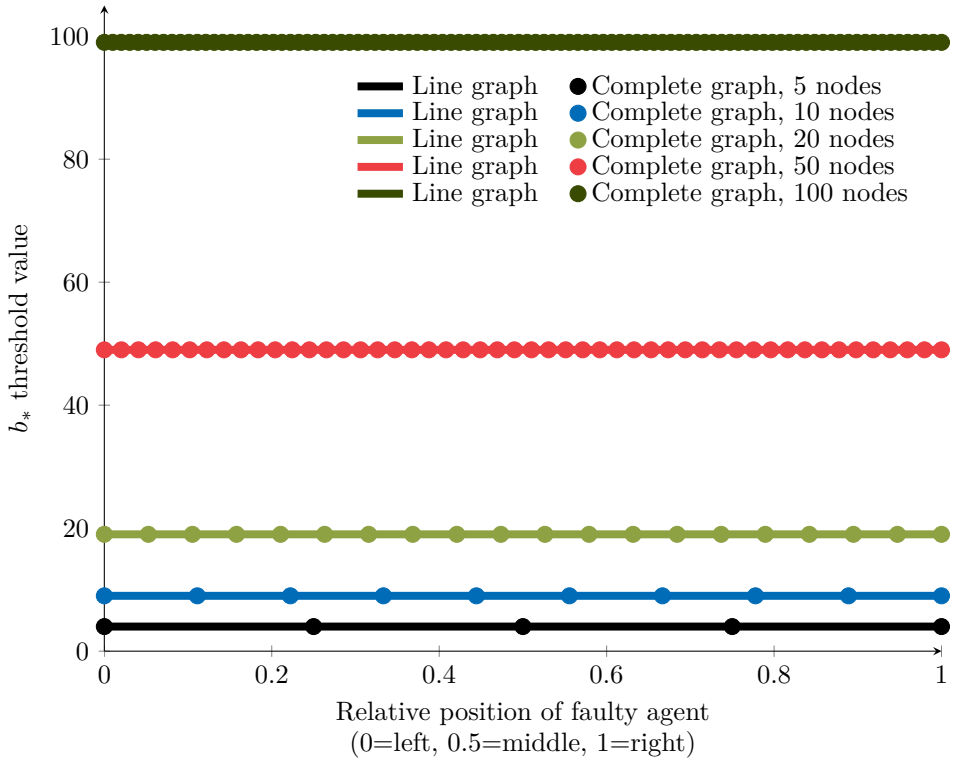


Figure 3.5: Threshold values for b_* for the line and complete graphs. The threshold value $b_* \geq n$ seems to be a tight condition, and independent of the agents position and the topology.

Next, we examine the threshold value of b_* for the system to be either escapable or trackable. We consider both the line graph and complete graph as communication topologies, and also vary the size of the network from 5 to 100 agents. For the line graph, it is interesting to see if the faulty agents position on the line would affect the threshold value b_*. The results are shown in fig. 3.5. According to Theorem 3.5, the threshold must be at least $b_* \geq |\mathcal{N}_0|$, which seems to be a tight condition for the complete graph where $|\mathcal{N}_0| = n$. The simulations does however indicate that the same value of $b_* \geq n$ is true even for the line graph, and we therefore hypothesize that our result could be strengthened to be $b_* \geq n$ as a both necessary and sufficient condition for trackability, i.e., that the condition is independent of the topology, and only depends on the number of follower agents.

3.4 Conclusions

This chapter presented a framework for a multi-agent system tracking a faulty agent through consensus updates. In contrast to most published results, the faulty agent is measuring the position of its follower agents and acting against the follower agents using positive feedback from their relative state. Sufficient conditions and necessary conditions were established for the multi-agent system to converge to consensus under both fixed and time dependent communication topologies, and these conditions could also be related to the topology. Numerical simulations validate our results, and also indicate that a stronger condition might be possible.

Chapter 4

Convergence in Peer-to-Peer Networks

"Never trust a computer you can't throw out a window."
— Steve Wozniak

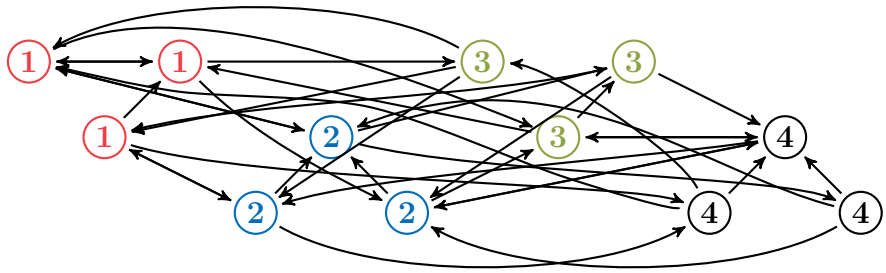
In this chapter, we investigate a peer-to-peer (P2P) network for efficient live-streaming television, inspired by gradientTv and Sepidar (Payberah *et al.*, 2010a,b). The gradient topologies are fundamental in self-organizing systems, and are a generalization of rooted trees. The goal of this application is to distribute a data stream from a small set of seed nodes to every other node in the network, and the problem is to design distributed algorithms for creating an efficient overlay network topology. In particular, this chapter deals with the convergence problem, in which the network graph converges to a complete *gradient overlay network*. The contribution of this chapter is the convergence analysis of the given algorithm.

The outline of this chapter is as follows: In section 4.1 we introduce the network model and topology convergence problem, which we analyze in section 4.2. In section 4.3 we simulate the construction of a gradient topology using the model in section 4.1, and in section 4.4 we evaluate the live-streaming performance in a real P2P application using the gradient overlay topology. Finally, section 4.5 concludes this chapter.

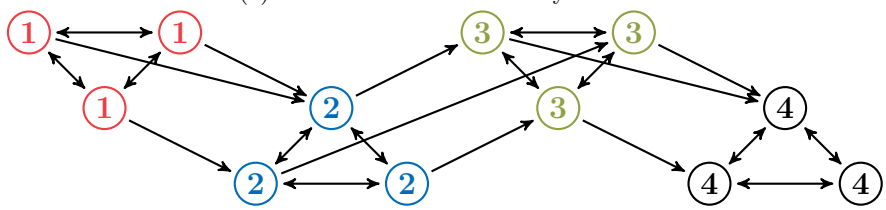
4.1 The Gradient Topology Problem

The gradient topology belongs to the class of gossip-generated overlay networks that are built from a random overlay network through symmetry breaking using a preference function. Thus, we are given a node set $\mathcal{V} = \{1, \ldots, N\}$, and need to select edges $\mathcal{E}$ to construct our network $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

In the live streaming application, the idea is to utilize the nodes in the P2P network to aid in the content distribution, but since the peers are heterogeneous, not all peers will be equally useful. Thus, we classify each node $i \in \mathcal{V}$ with its *utility value* $u_i \in \mathbb{R}$, which captures, for example, the node's upload capacity, latency and reliability for the P2P network.



(a) The initial random overlay network



(b) The network after converging to a complete gradient topology

Figure 4.1: The gradient network is described as a directed graph. The nodes are labeled with their respective utility value, and the edges from the similar neighbor sets are shown. In the gradient topology, paths of increasing utilities emerge.

Formally, a gradient topology is defined as an overlay network satisfying, for any two nodes v_1 and v_2 with utility values u_{v_1} and u_{v_2}, if $u_{v_1} \geq u_{v_2}$ then $\text{dist}\,(v_1, v_\star) \leq \text{dist}\,(v_2, v_\star)$, where $v_\star$ is the node with highest utility in the system and $\text{dist}\,(\cdot,\cdot)$ is the length of the shortest path between the nodes in the network (Sacha, 2009). In other words, nodes with a higher utility value should be closer to the seed nodes compared to nodes with a lower utility value, so that gradient paths of increasing utilities emerge in the system, see fig. 4.1.

In constructing the gradient overlay, the nodes $i \in \mathcal{V}$ build two sets of neighbors: a *similar view* $\mathcal{N}_i^s(t)$ and a *random view* $\mathcal{N}_i^r(t)$. For the similar view, nodes prefer neighbors with close but slightly higher utility values, while the random view is used to sample new nodes with uniform probability for possible inclusion in the similar view. Thus, the node i's total neighbors are $\mathcal{N}_i(t) = \mathcal{N}_i^s(t) \cup \mathcal{N}_i^r(t)$.

Each node i defines a *preference function* $>_i$ over its neighbors, where node i is said to prefer node a over node b (denoted by $a >_i b$) if

$$\begin{aligned} u_a \geq u_i > u_b \qquad & \text{or if} \\ |u_a - u_i| < |u_b - u_i| \qquad & \text{when } u_a, u_b > u_i \text{ or } u_a, u_b < u_i. \end{aligned}$$

Further, let $\min \mathcal{N}_i^s$ denote node i's least preferred neighbor in its similar neighbor set.

For any given initial overlay network, the topology is evolving through each node i at each time t updating its own neighbor set $\mathcal{N}_i(t)$ independently of the other nodes according to Algorithm 4.1. The algorithm can be summarized as sampling random nodes from the network, and if the random node is preferred over the least preferred node in the similar set, then those two neighbors are exchanged.

Algorithm 4.1 Topology Dynamics

for every $t = 1, 2, 3, \ldots$ **do**
  for each node $i \in \mathcal{V}$ **do**
    Let $\mathcal{N}_i^r(t) = \{j\}$, where $j \in \mathcal{V}$ is a randomly selected node with uniform probability p_t, $0 < Np_t < 1$. Otherwise $\mathcal{N}_i^r(t) = \emptyset$.
    if $\mathcal{N}_i^r(t) \neq \emptyset$ **then**
      if $j \notin \mathcal{N}_i^s(t-1)$ and $j >_i \min \mathcal{N}_i^s(t-1)$ **then**
        $\mathcal{N}_i^s(t) = \mathcal{N}_i^s(t-1) \cup \{j\} \setminus \{\min \mathcal{N}_i^s(t-1)\}$
      else
        $\mathcal{N}_i^s(t) = \mathcal{N}_i^s(t-1)$
      end if
    end if
  end for
end for

It is assumed that the node degree $d_i(t) = |\mathcal{N}_i^s(t)| = d_i$ stays constant throughout the algorithm. Notice that the sampling probabilities p_t can be time dependent, and that the random neighbor set $\mathcal{N}_i^r(t)$ is empty with probability $1 - Np_t$. The reason is that a node typically samples more frequently just after joining the network to improve its neighbors, and then stabilizing at a lower sampling rate.

The preference function $>_i$ induced a partial order on the nodes $\mathcal{V}$, thus there could be multiple optimal neighbor sets. In order to study the system topology convergence to a gradient structure with the proposed algorithm, we let Λ_i denote the set of optimal similar neighbor sets for node i, i.e., $\forall \widehat{N} \in \Lambda_i$ there are no $j \in \widehat{N}$ and $k \in \mathcal{V} \backslash \widehat{N}$ such that $k >_i j$.

For every node $i \in \mathcal{V}$, we define $X_i(t)$ that counts the number non-optimal neighbors in i's similar neighbor set,

$$X_i(t) \doteq d_i - \max_{\widehat{N} \in \Lambda_i} \left| \mathcal{N}_i^s(t) \cap \widehat{N} \right|.$$

Notice that $X_i(t)$ is monotonically decreasing under Algorithm 4.1 since an optimal neighbor will never be removed from the similar neighbor set $\mathcal{N}_i^s(t)$.

Let $\mathcal{G}(t)$ be the graphs generated by Algorithm 4.1. Then we give the definition of gradient convergence as follows (see also fig. 4.1).

Definition 4.1. $\mathcal{G}(t)$ is said to converge to a gradient topology if

$$\lim_{t\to\infty} X_i(t) = 0$$

for all $i \in \mathcal{V}$.

4.2 Convergence Analysis

Since each node updates its neighbor set independent of the other nodes, the analysis can be carried out separately on each $X_i(t)$. We therefore simplify the notations in the following discussion, and let $X(t)$ represents $X_i(t)$ for an arbitrary node $i \in \mathcal{V}$.

Denote the maximum degree $D = \max_i\{d_i\}$, then it is not hard to see that $X(0) = D$ is the worst initial condition. Furthermore, $X(t)$ decreases precisely when the randomly sampled node is a new optimal neighbor, and the probability of this event occurring is minimal when the optimal solution is unique, and then equal to

$$\mathbb{P}\left[X(t+1) = k-1 \mid X(t) = k\right] = kp_t, \quad k = 1, \ldots, D, \tag{4.1}$$

where k is the number of non-optimal neighbors.

In the following theorem we propose a both necessary and sufficient condition on the probabilities p_t for almost sure convergence of Algorithm 4.1.

Theorem 4.1. *The graph generated by Algorithm 4.1 converges to a gradient topology ($X(t) = 0$) with probability 1 if and only if*

$$\lim_{T\to\infty} \prod_{t=0}^{T} (1 - p_t) = 0. \tag{4.2}$$

Before proving Theorem 4.1, let us take a closer look at Algorithm 4.1, and notice especially that the stochastic process in eq. (4.1) for $X(t)$ has the Markov property, hence we can describe it as a simple Markov chain, fig. 4.2.

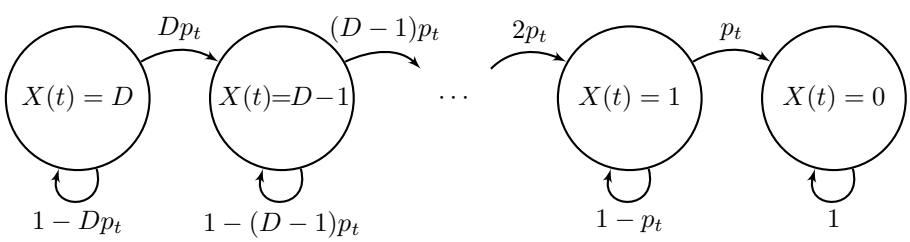


Figure 4.2: Markov chain illustration of the stochastic process.

Let $\boldsymbol{\pi}(t)$ denote the row vector of probabilities for the states $X(t)$, i.e.,

$$\pi_i(t) = \mathbb{P}\left[X(t) = D - i\right]. \tag{4.3}$$

The evolution of $\boldsymbol{\pi}(t)$ can be written in matrix form as

$$\boldsymbol{\pi}(t+1) = \boldsymbol{\pi}(t)P_t, \tag{4.4}$$

where P_t is the transition matrix at time t,

$$P_t = \begin{bmatrix} 1 - Dp_t & Dp_t & 0 & \cdots & 0 & 0 \\ 0 & 1-(D-1)p_t & (D-1)p_t & \cdots & 0 & 0 \\ 0 & 0 & 1-(D-2)p_t & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1-p_t & p_t \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

Since P_t is a triangular matrix, the eigenvalues are given by the diagonal elements, i.e., the eigenvalues of P_t are $\lambda_i(t) = 1 - (D-i)p_t$, $i = 0, \ldots, D$. Notice that there is a single unit eigenvalue $\lambda_D(t) = 1$, and all other eigenvalues are strictly less than one. Furthermore, all eigenvalues are distinct, hence the eigenvectors form a basis for $\mathbb{R}^D$. In the following lemma, we characterize the eigenvectors.

Lemma 4.2. *The eigenvector $\xi^i(t)$ corresponding to eigenvalue $\lambda_i(t)$ is independent of $p_t \neq 0$, for $i = 0, \ldots, D$.*

Proof. The (left-)eigenvectors of P_t satisfy $\lambda_i(t)\xi^i(t) = \xi^i(t)P_t$. Let $\xi^i_j(t)$ denote the j:th component of $\xi^i(t)$, then, by inspection of the matrix P_t, we have the system of equations

$$\begin{aligned} (1-(D-i)p_t)\,\xi^i_0(t) &= (1-Dp_t)\,\xi^i_0(t) \\ (1-(D-i)p_t)\,\xi^i_j(t) &= (1-(D-j)p_t)\,\xi^i_j(t) \\ &\quad + (D-j+1)p_t\xi^i_{j-1}(t) \qquad\qquad j = 1, \ldots, D \end{aligned}$$

which is equivalent to

$$\begin{aligned} i\xi^i_0(t) &= 0 \\ (i-j)\xi^i_j(t) &= (D-j+1)\xi^i_{j-1}(t) \qquad\qquad j = 1, \ldots, D \end{aligned}$$

or further

$$\begin{aligned} \xi^i_j(t) &= 0 && \text{if } j < i \\ \xi^i_j(t)\frac{i-j}{D-j+1} &= \xi^i_{j-1}(t) && \text{if } j > i \end{aligned} \tag{4.5}$$

while $\xi_i^i(t)$ can be chosen as an arbitrary non-zero value to scale the eigenvector. From eq. (4.5) it is evident that the eigenvectors are independent of p_t. □

An important consequence of Lemma 4.2 is that all P_t, independent of t, have the same eigenvectors, and are thus simultaneously diagonalizable. Hence we can simplify the notation by dropping the parameter t from the eigenvectors ξ^i.

Let us now return to the initial probability distribution $\boldsymbol{\pi}(0)$, and decompose it into the eigenvector basis as

$$\boldsymbol{\pi}(0) = \sum_{i=0}^{D} \alpha_i \xi^i, \tag{4.6}$$

for some real numbers α_i.

Lemma 4.3. *$\alpha_D \xi^D = \mathbf{e}_D$, where $\mathbf{e}_i$ is the standard basis $[0, \ldots, 0, 1, 0, \ldots, 0]^T$.*

Proof. Let us consider $\xi^i \mathbb{1}$ for $i = 0, \ldots, D-1$. By equation (4.5),

$$\xi^i \mathbb{1} = \sum_{j=0}^{D} \xi_j^i = \sum_{j=i}^{D} \xi_j^i = \sum_{j=0}^{D-i} \xi_{i+j}^i$$

We will show by induction that

$$\sum_{j=0}^{k} \xi_{i+j}^i = \frac{D-i-k}{D-i} \xi_{i+k}^i. \tag{4.7}$$

The case when $k = 0$ is clearly true, thus, assume that (4.7) holds for k and consider the case when $k+1$,

$$\begin{aligned}
\sum_{j=0}^{k+1} \xi_{i+j}^i &= \sum_{j=0}^{k} \xi_{i+j}^i + \xi_{i+k+1}^i \\
&= \frac{D-i-k}{D-i} \xi_{i+k}^i + \xi_{i+k+1}^i \\
&= \frac{D-i-k}{D-i} \frac{-(k+1)}{D-i-k} \xi_{i+k+1}^i + \xi_{i+k+1}^i \\
&= \frac{D-i-(k+1)}{D-i} \xi_{i+k+1}^i
\end{aligned}$$

Using (4.7) implies that $\xi^i \mathbb{1} = 0$, $i = 0, \ldots, D-1$, and thus, $\boldsymbol{\pi}(0)\mathbb{1} = \alpha_D \xi^D \mathbb{1}$. Now, since $\boldsymbol{\pi}(0)$ is a probability distribution, we know that $\boldsymbol{\pi}(0)\mathbb{1} = 1$, but (4.5) tells us that only the last component of ξ^D is non-zero, hence the lemma follows. □

We are now ready to prove the main theorem.

Proof of Theorem 4.1. The almost sure convergence condition means that

$$\begin{aligned} \lim_{T\to\infty} \mathbb{P}\left[X(T)=0\right] = 1 \quad &\Rightarrow \\ \lim_{T\to\infty} \pi_D(T) = 1 \quad &\Rightarrow \\ \lim_{T\to\infty} \boldsymbol{\pi}(T) = \mathbf{e}_D \end{aligned}$$

Equations (4.4) and (4.6) gives us

$$\begin{aligned} \boldsymbol{\pi}(T) &= \boldsymbol{\pi}(0) \prod_{t=0}^{T-1} P_t \\ &= \sum_{i=0}^{D} \alpha_i \xi^i \prod_{t=0}^{T-1} P_t \\ &= \sum_{i=0}^{D} \alpha_i \xi^i \prod_{t=0}^{T-1} \lambda_i(t) \\ &= \sum_{i=0}^{D-1} \alpha_i \xi^i \prod_{t=0}^{T-1} \lambda_i(t) + \mathbf{e}_D. \end{aligned} \tag{4.8}$$

Consider the limit $\lim_{T\to\infty} \boldsymbol{\pi}(T)$,

$$\begin{aligned} \lim_{T\to\infty} |\boldsymbol{\pi}(T) - \mathbf{e}_D| &= \lim_{T\to\infty} \left| \sum_{i=0}^{D-1} \alpha_i \xi^i \prod_{t=0}^{T-1} \lambda_i(t) \right| \\ &\leq \sum_{i=0}^{D-1} \left|\alpha_i \xi^i\right| \cdot \lim_{T\to\infty} \prod_{t=0}^{T-1} (1-p_t). \end{aligned}$$

Clearly, this converges to zero if $\lim_{T\to\infty} \prod_{t=0}^{T}(1-p_t) = 0$.

Furthermore, the set of initial probability distributions spawns $\mathbb{R}^D$. Thus, there exists an initial probability distribution $\boldsymbol{\pi}(0)$ such that $\alpha_{D-1} \neq 0$. Assume that the limit $\lim_{T\to\infty} \prod_{t=0}^{T}(1-p_t) = c > 0$ is finite (the limit exists since it is a monotone bounded sequence), then

$$\lim_{T\to\infty} |\boldsymbol{\pi}(T) - \mathbf{e}_D| = \left| \sum_{i=0}^{D-2} \alpha_i \xi^i \left(\lim_{T\to\infty} \prod_{t=0}^{T-1} \lambda_i(t) \right) + c\alpha_{D-1}\xi^{D-1} \right| > 0 \tag{4.9}$$

since the eigenvectors are linearly independent. Thus, we have proven the theorem. □

Corollary 4.1. *The graph generated by Algorithm 4.1 converges to a gradient topology with probability 1 if and only if*

$$\lim_{T\to\infty} \sum_{t=0}^{T} p_t = \infty. \tag{4.10}$$

Proof. This follows directly from Theorem 4.1, and the relation

$$\lim_{T\to\infty} \prod_{t=0}^{T} (1 - p_t) = 0 \quad \Leftrightarrow \quad \lim_{T\to\infty} \sum_{t=0}^{T} p_t = \infty$$

for $0 < p_t < 1$. □

Convergence Rate Estimation

We will now investigate the convergence rate of $X(t)$, with a constant sampling probability $p_t = p$. Define the stochastic variable T_i as the time when $X(t)$ reaches 0, when starting at $X(0) = i$,

$$T_i = \inf_t [X(t) = 0 \mid X(0) = i].$$

Further, let $M_i = \mathbb{E}[T_i]$ denote the expected convergence time when starting at $X(0) = i$. Clearly $M_0 = 0$, and for $i = 1, \ldots, D$ we have the recursion

$$\begin{aligned} M_i &= 1 + \mathbb{P}[X(t+1) = i - 1 \mid X(t) = i] \cdot M_{i-1} \\ &\quad + \mathbb{P}[X(t+1) = i \mid X(t) = i] \cdot M_i \\ &= 1 + ipM_{i-1} + (1 - ip)M_i \end{aligned}$$

which can be further simplified to

$$M_i = \frac{1 + ipM_{i-1}}{ip} = \frac{1}{ip} + M_{i-1}.$$

By continuing with induction, we can sum up the expected convergence time as

$$M_i = \frac{1}{p} \sum_{n=1}^{i} \frac{1}{n}.$$

The worst initial case is when $X(0) = D$, where the expected convergence time is

$$M_D = \frac{1}{p} \sum_{n=1}^{D} \frac{1}{n} \leq \frac{1 + \ln(D)}{p}. \tag{4.11}$$

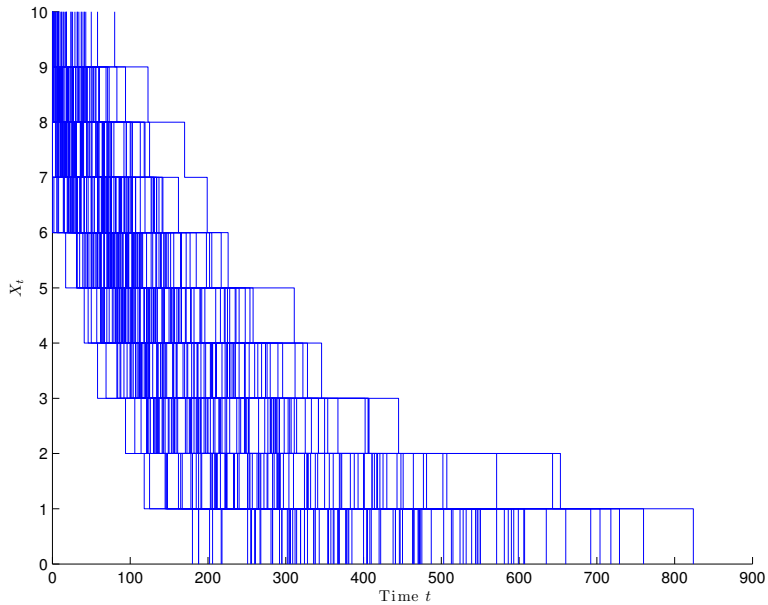
Remark. M_D is the expected time for an individual node's neighbor set to converge, not the expected time for *all* nodes to converge to a gradient topology. As such, it provides a lower bound on the convergence time.

4.3 Convergence Simulation

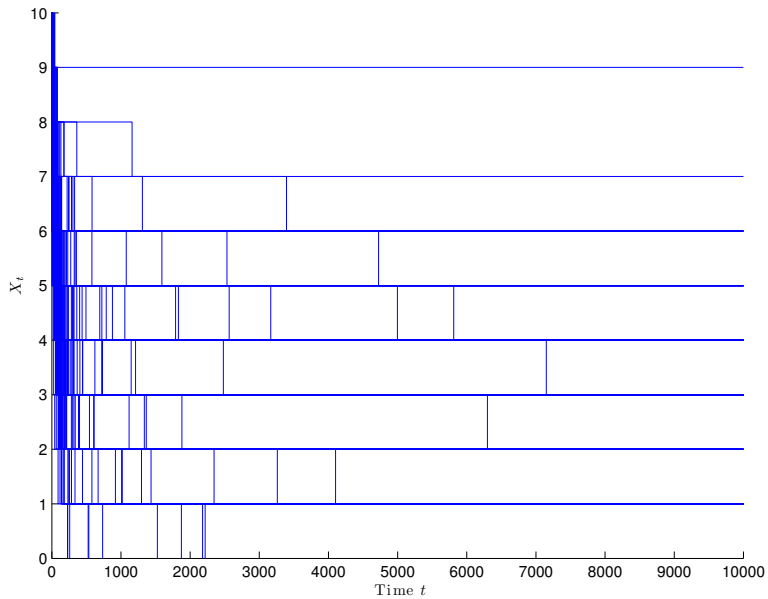
Here we examine the convergence rate of Algorithm 4.1 with numerical examples. In the first two simulations (figs. 4.3a and 4.3b) the degree of each node is $d_i = 10$ and the total number of nodes in the network is $N = 100$. For the third simulation (fig. 4.3c) the degree is $d_i = 50$, and the total number of nodes in the network is $N = 500$.

The similar view $\mathcal{N}_i^s(0)$ is initialized with d_i nodes uniformly chosen among all nodes in the network. In the first and third simulation the sampling probability p_t is held at a constant value of $\frac{1}{2N}$. Hence, for each node, and at each iteration of the algorithm, the random view is empty with 50% probability. Theorem 4.1 guarantees the convergence of the algorithm for these examples, which is also confirmed by the simulations. These two simulations should also be compared to the expected convergence rate given in eq. (4.11), 566 and 4479 iterations respectively.

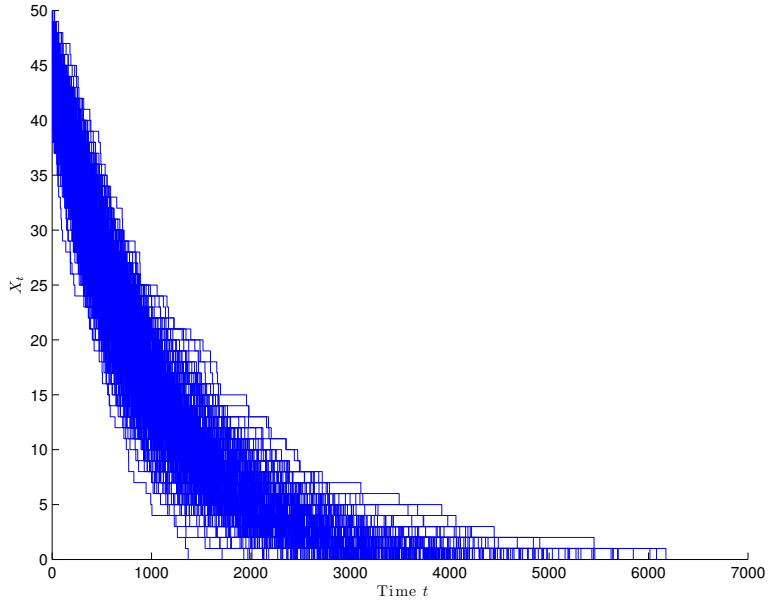
In the second simulation (fig. 4.3b), we simulate a decaying probability $p_t = \frac{1}{N}\frac{1}{(1+t/100)^2}$. Notice that $\sum_{t=0}^{\infty} Np_t < 101$, hence, by Corollary 4.1, there is a positive probability that the algorithm does not converge to a gradient topology. This is also confirmed by the simulation, in which the gradient topology is missing.



(a) Convergence in a network with 100 nodes, $d_i = 10$, and constant probability $Np_t = \frac{1}{2}$.



(b) Convergence is missing in a network with 100 nodes, $d_i = 10$, and decaying probability $Np_t = \frac{1}{(1+t/100)^2}$.



(c) Convergence in a network with 500 nodes, $d_i = 50$, and constant probability $Np_t = \frac{1}{2}$.

Figure 4.3: Convergence rate simulations. The neighbor set measurement $X(t)$, for each node in the network, is shown as a function of the iteration number t.

4.4 Evaluating Live-Streaming using the Gradient Topology

Now we turn to an evaluation of the effect of sampling nodes from the gradient overlay network compared to a random overlay network when building a P2P live-streaming application, called GLive. GLive is based on nodes cooperating to share a media stream supplied by a source node, and uses an approximate auction algorithm to match nodes that are willing and able to share the stream with one another. GLive extends the tree-based live-streaming, gradienTv (Payberah *et al.*, 2010a) and Sepidar (Payberah *et al.*, 2010b), to mesh-based live-streaming.

Nodes want to establish connections to other nodes that are as close as possible to the source. They bid for connections to the best neighbors using their own upload bandwidth, and nodes share their bounded number of connections with nodes who bid the highest (contribute the most upload bandwidth). Auctions are continuous and restarted on failures or free-riding. The desired effect of our auction algorithm is that the source will upload to nodes who contribute the most upload bandwidth, who will, in turn, upload to nodes who contribute the next highest amount of bandwidth, and so on until the topology is fully constructed.

One of the main problems with the lack of global information about nodes' upload bandwidths is that it affects the rate of convergence of the auction algorithm.

Nodes would ideally like to bid for connections to other nodes who they can afford to connect to, rather than win a connection to a better node and later be removed because a better bid was received. The traditional way to discover nodes to bid on is using a uniform random peer-sampling service (Jelasity *et al.*, 2007). Instead, we use the gradient overlay to sample nodes, where a node's utility value is the upload bandwidth it contributes to the system. As such, the gradient should provide other nodes with references to nodes who have well-matched upload bandwidths. Payberah *et al.* (2010b) showed that using the gradient overlay network reduced the rate of parent switching for tree-based live-streaming by 20% compared to random peer sampling. Here, we show for GLive the effect of sampling neighbors using random peer sampling (GLive/Random) versus sampling from the gradient overlay (GLive/Gradient).

GLive is implemented using Kompics' discrete event simulator (Arad *et al.*, 2009) that provides different bandwidth, latency and churn models. In our experimental setup, we set the streaming rate to 512 kbit/s, which is divided into blocks of 16 kB. Nodes start playing the media after buffering it for 5 seconds. The size of the similar view in GLive is 15 nodes, and in the auction algorithm, nodes have 8 download connections. To model upload bandwidth, we assume that each upload connection has an available bandwidth of 64 kbit/s and that the number of upload connections for the nodes is set to $2i$, where i is picked randomly from the range 1 to 10. This means that nodes have an upload bandwidth capacity between 128 kbit/s and 1.25 Mbit/s. As the average upload bandwidth of 704 kbit/s is not much higher than the streaming rate of 512 kbit/s, nodes need to find good parents to achieve the streaming performance. The media source is a single node with 40 upload connections, providing five times the upload bandwidth of the stream rate. We assume 11 utility levels, such that nodes contributing the same amount of upload bandwidth are located at the same utility level. Latencies between nodes are modeled using a latency map based on the King data-set (Gummadi *et al.*, 2002). We assume the size of sliding window for downloading is 32 blocks, such that the first 16 blocks are considered as the in-order set and the next 16 blocks are the blocks in the rare set. A block is chosen for download from the in-order set with 90% probability, and from the rare set with 10% probability. In the experiments, we measure the following metrics:

1. *Playback continuity*: the percentage of blocks that a node received before their playback time. We consider the case where nodes have a playback continuity of greater than 99%;

2. *Playback latency*: the difference in seconds between the playback point of a node and the playback point at the media source.

We compare the playback continuity and playback latency of GLive/Gradient and GLive/Random in the following three scenarios:

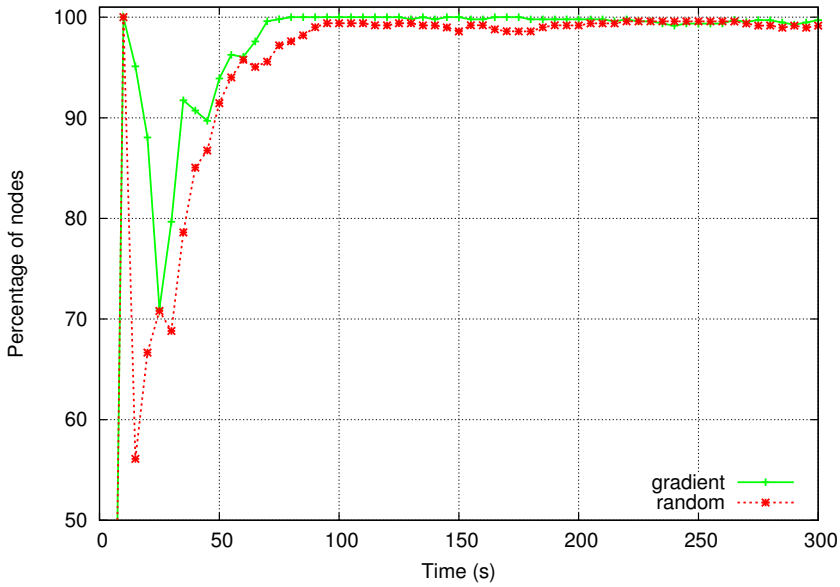
1. *Churn*: 500 nodes join the system following a Poisson distribution, with an average inter-arrival time of 100 milliseconds. Then, until the end of the simulation, nodes join and fail continuously following the same distribution with an average inter-arrival time of 1000 milliseconds;

2. *Flash crowd*: first, 100 nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds. Then, 1000 nodes join following the same distribution with a shortened average inter-arrival time of 10 milliseconds;

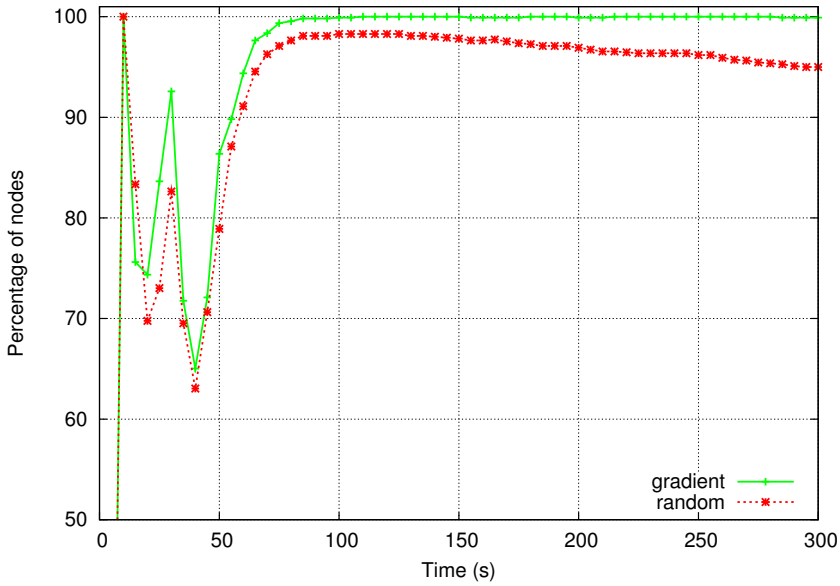
3. *Catastrophic failure*: 1000 nodes join the system following a Poisson distribution with an average inter-arrival time of 100 milliseconds. Then, 500 existing nodes fail following a Poisson distribution with an average inter-failure time of 10 milliseconds.

Figure 4.4 shows the percentage of the nodes that have a playback continuity of at least 99%. We see that all the nodes in GLive/Gradient receive at least 99% of all the blocks very quickly in all scenarios, while it takes slightly more time for GLive/Random. This is because nodes in GLive/Gradient find a good set of matches faster than nodes in GLive/Random by running the auction algorithm against nodes with similar upload bandwidth. One point to note is that the 5 seconds of buffering cause the spike in playback continuity at the start, which then drops off as nodes are joining the system. To summarize, using the gradient overlay instead of random sampling produces better performance when the system is undergoing large changes - such as large numbers of nodes joining or failing over a short period of time.

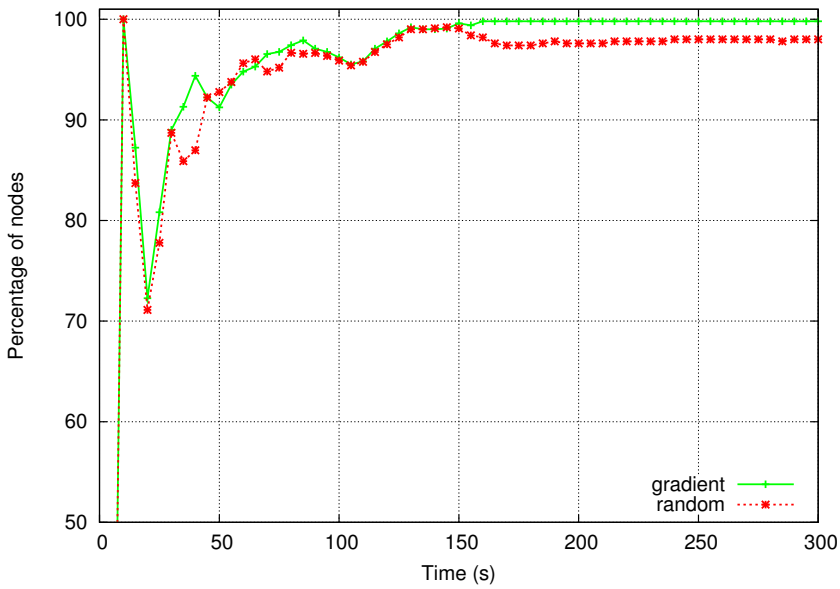
Figure 4.5 shows the playback latency of the systems in the different scenarios. As we can see, although there is only a small difference between the systems, GLive/Gradient consistently maintains relatively shorter playback latency than GLive/Random for all experiments. The playback latency includes both the 5 seconds buffering time and the time required to pull the blocks over the live-streaming overlay constructed using the auction algorithm.



(a) Churn

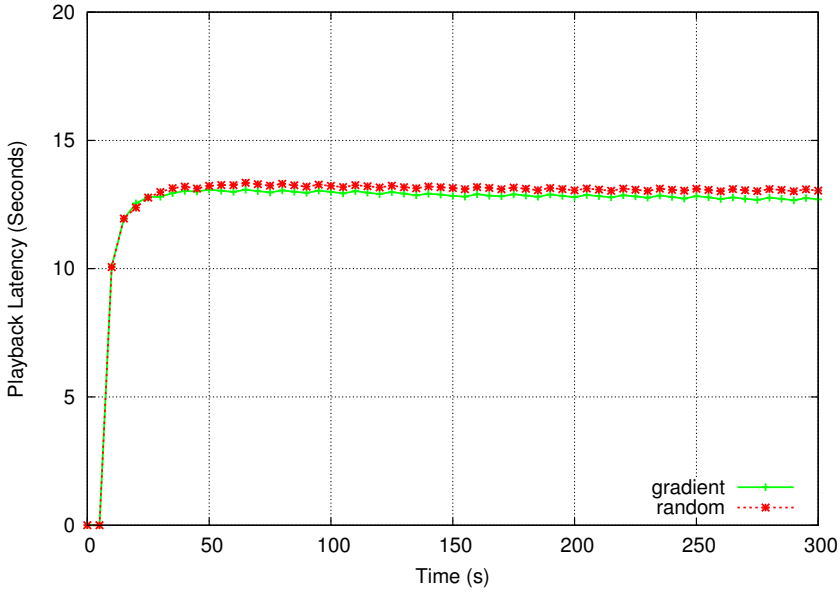


(b) Flash Crowd

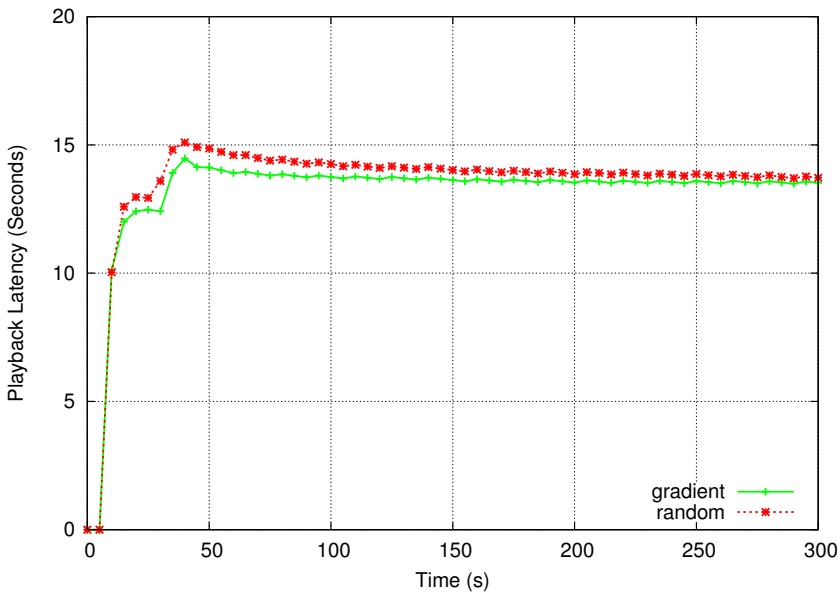


(c) Catastrophic failure

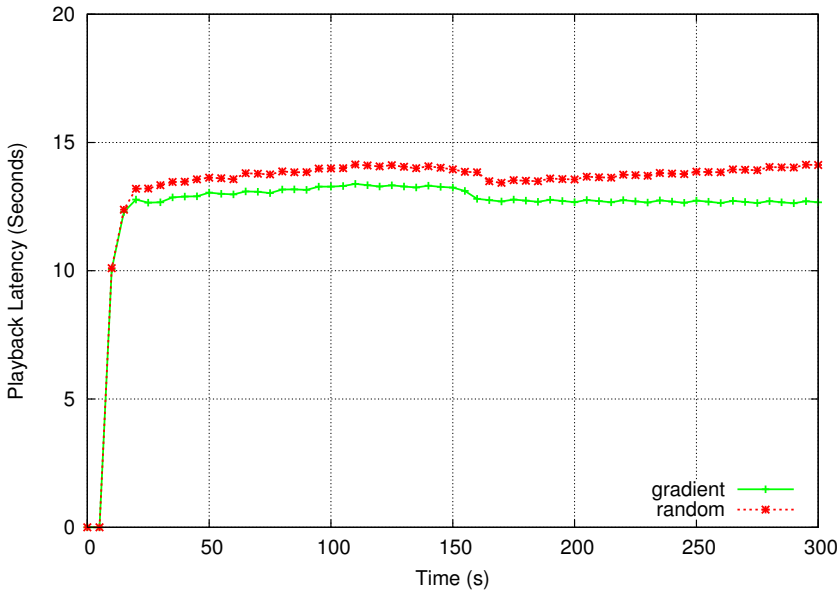
Figure 4.4: Playback continuity of the GLive/Gradient and GLive/Random systems in the churn, flash crowd and catastrophic failure scenarios.



(a) Churn



(b) Flash Crowd



(c) Catastrophic failure

Figure 4.5: Playback latency of the GLive/Gradient and GLive/Random systems in the churn, flash crowd and catastrophic failure scenarios.

4.5 Conclusions

In this chapter, we studied the network topology convergence problem for the gossip-generated gradient overlay network. We showed necessary and sufficient conditions for convergence to a complete gradient structure, and we also characterized the expected convergence time for a network without churn.

Our live-streaming experiments show the potential advantages of topologies built using preference functions. We showed how nodes can use implicit information captured in the gradient topology to efficiently find suitable neighbors compared to using random sampling. As such, our work on proving convergence properties of the gradient topology should have significance for other future information-carrying topologies.

Chapter 5

Conclusions and Future Work

"The future belongs to those who believe in the beauty of their dreams."
— Eleanor Roosevelt

5.1 Conclusions

In this thesis we have explored some important and interesting algorithms for estimation and convergence in networks. We based these algorithms on simple consensus protocols which lead to easily distributed solutions, which can be implemented in networks with unreliable communication and with moderate hardware and software requirements. These methods constitute fundamental building blocks of distributed systems, with many diverse applications.

Estimation in Anonymous Networks

We derived distributed estimators to estimate both the network size and probability mass functions in the specific framework where agents are anonymous. These estimators were based on probabilistic methods and the max consensus protocol. One of the main advantages with the max consensus protocol compared to other strategies is that it has the fastest possible convergence time, and it is also very easy to implement, even in networks with unreliable communication through pairwise gossiping communication. Even though the accuracy of these methods is not perfect in general, it can be made arbitrarily good by increasing the packet sizes.

We designed these methods to specifically handle time dependent networks through a regularization term, which penalizes hypotheses conflicting with a-priori assumptions on the network's behavior. We explicitly considered and characterized the class of quadratic regularization terms, which resulted in closed-form estimators.

Faulty Nodes in Consensus Protocols

We presented a multi-agent framework for tracking a misbehaving agent through consensus strategies. The faulty agent in this scheme was measuring the position of its neighboring agents, and acting against them using positive state feedback. We derived both sufficient conditions and necessary conditions for the multi-agent system to converge to consensus, and considered both fixed and time dependent communication topologies. In particular, we related these conditions to the network topology.

Convergence in Peer-to-Peer Networks

We studied a P2P network for live-streaming video applications, and considered the advantages of using a gradient topology for finding suitable neighbors. The advantage of this method is that the peers can find a good set of neighbors more quickly, and thus avoid rapid switching, which increases the latency and probability of interruptions. We derived both necessary and sufficient conditions for convergence to a complete gradient topology, and we also characterized the expected convergence time.

5.2 Future Work

There are many natural extensions of the work presented in this thesis. The future work can be divided into general formulations in the field of dynamical network systems, and specific problem formulations related to the problems studied in this thesis. We summarize below some ideas for future research.

General Research Directions

Dynamical network systems is an extremely rich field, with many interesting open problems. The general research directions can be divided into two tracks: the experimental and the theoretical.

From a theoretical perspective, one fundamental task is to develop general tools for analyzing dynamical networks, that could be used in cross disciplinary settings. In the field of automatic control, we have for a long time been studying feedback loops. What role does feedback loops play in the formation of network structures? How can feedback loops that govern the evolution of networks be found? Can concepts such as controllability and observability aid in the design of high performance dynamical network systems?

With an abundance of practical applications, it would be interesting to work with, and analyze real world data. For example studying community formation in social networks, and study the interaction between the network structure and the information propagation through the network.

Estimation in Anonymous Networks

We estimated two specific properties in the anonymous network framework, but a natural continuation is to ask what can be computed or estimated in this framework. There has been some work done on deterministic computation in anonymous networks, but through the introduction of random number generation, these results could be extended to a broader class of functions. Specifically, what can be learned about the network topology besides the size, and how can these estimates be used to create intelligent networks that automatically reconfigure themselves to be robust against failures.

It was shown that the accuracy could be improved by increasing the packet size. An important question is to consider the quantization effects due to limited number precision, and to design an optimal alphabet which results in the highest precision per bit transmitted. A natural direction would also consist of developing on-line algorithms for tuning the estimation parameters, especially the packet size based on the current performance requirements in the application.

We assumed that the communication was synchronized, but it would be interesting to relax this assumption. Either by introducing distributed clock synchronization schemes, or by changing our algorithms to an event-triggered version. This could be accomplished by assuming that any agent could initiate a new epoch using a randomized time-out event. Could additional properties be estimated by using the transient information while the consensus algorithm is converging to the maximum?

Another research direction could be to study the robustness of this scheme against different types of attacks. Either modeled as randomized noise or failures among the nodes, or with malicious agents who are actively trying to influence the estimate in a particular direction.

Faulty Nodes in Consensus Protocols

In our numerical simulations we hypothesized that there is a strong threshold phenomenon: the system is globally trackable if and only if $b_* > n$. We would like to validate this hypothesis and further explore the threshold.

One of the main assumptions in this chapter was that the faulty agent was still following the same consensus protocol, where only the weights had been perturbed. It would be interesting to extend this into broader classes of faults, and especially in the direction of network games, where the worst possible actions are considered. This would also relate to fault detection, and trying to find faulty agents which should be expunged from the network.

Another assumption was that the network should be undirected, and a natural extension would be to consider directed networks, where nodes could have different in-degree and out-degree. The connectivity assumption could then be replaced by a strongly connected assumption, which should still allow the arguments in the

time dependent case, where paths of influence are built. Would there be any other fundamental changes in the results?

This work was motivated by robustness against faults among the agents in a multi-agent system, but in our work, we studied the case where the faulty agent was known. This leads to the natural extension to consider a global robustness property, and conditions where the system will reach consensus irrespective of the faulty agents.

Convergence in Peer-to-Peer Networks

As we saw in our live experiments, P2P networks often exhibit stochastic behavior, with nodes both joining and leaving the network (churn). In our analysis, we instead considered a static network in order to define the convergence time to an optimal configuration. We would like to extend this analysis to networks with churn, and examine how high churn rates could be tolerated while still maintaining a reasonable gradient overlay structure, and also the relationship with the sampling rate.

Another property that we ignored was that there is an underlying physical network, where the latency can be specified as a pairwise map. Thus, the utility value of a node might not be a truly universal property, and we could instead define the utility value as a pairwise function describing the utility a node could provide to a specific other node.

The sampling was performed with given probabilities, so an important question is how to tune these probabilities, and could they be tuned on-line as topology changes are detected? Furthermore, we used uniform sampling among all peers, but some of the random peer sampling services give biased samples, most notably those based on random walks. What would the effect be of these biased samples, and in particular, if the samples where generated by random walks within the same gradient overlay topology?

Bibliography

"A dwarf standing on the shoulders of a giant may see farther than a giant himself"

— Robert Burton

Masoom M. Ali, Manisha Pal, and Jungsoo Woo. 2007. On the ratio of inverted gamma variates. *Austrian Journal of Statistics*, 36(2):153–159.

Redouane Ali, Suksant Sae Lor, and Miguel Rio. 2009. Two Algorithms for Network Size Estimation for Master/Slave Ad Hoc Networks. In *3rd IEEE International Symposium on Advanced Networks and Telecommunication Systems*, pages 1–3.

Sara Alouf, Eitan Altman, Chadi Barakat, and Philippe Nain. 2004. On the dynamic estimation of multicast group sizes. In *16th International Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium.

Sara Alouf, Eitan Altman, and Philippe Nain. 2002. Optimal on-line estimation of the size of a dynamic multicast group. In *21st Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1109–1118.

Cosmin Arad, Jim Dowling, and Seif Haridi. 2009. Developing, Simulating, and Deploying Peer-to-Peer Systems using the Kompics Component Model. In *4th International ICST Conference on COMmunication System softWAre and middlewaRE*, Dublin, Ireland. ACM.

Tuncer Can Aysal, Boris N. Oreshkin, and Mark J. Coates. 2009. Accelerated Distributed Average Consensus via Localized Node State Prediction. *IEEE Transactions on Signal Processing*, 57(4):1563–1576.

Carlos Baquero, Paulo Sérgio Almeida, Raquel Menezes, and Paulo Jesus. 2012. Extrema Propagation: Fast Distributed Estimation of Sums and Network Sizes. *IEEE Transactions on Parallel and Distributed Systems*, 23(4):668–675.

Albert-László Barabási. 2002. *Linked: The New Science of Networks.* Perseus Books Group.

Tamer Başar and Geert Jan Olsder. 1999. *Dynamic noncooperative game theory*. Siam.

Vincent D. Blondel, Julien M. Hendrickx, and John N. Tsitsiklis. 2009. On Krause's Multi-Agent Consensus Model With State-Dependent Connectivity. *IEEE Transactions on Automatic Control*, 54(11):2586–2597.

Vincent D. Blondel, Julien M. Hendrickx, and John N. Tsitsiklis. 2010. Opinion dynamics for agents with opinion-dependent connections. In *49th IEEE Conference on Decision and Control*, pages 6626–6632.

Saverio Bolognani and Sandro Zampieri. 2013. A Distributed Control Strategy for Reactive Power Compensation in Smart Microgrids. *IEEE Transactions on Automatic Control*, PP(99):1.

Miguel Borges, Paulo Jesus, Carlos Baquero, and Paulo Sérgio Almeida. 2012. Spectra: Robust Estimation of Distribution Functions in Networks. *Distributed Applications and Interoperable Systems*, 7272:96–103.

Stephen P. Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. 2006. Randomized Gossip Algorithms. *IEEE Transactions on Information Theory*, 52(6): 2508–2530.

Yongcan Cao and Wei Ren. 2009. Containment control with multiple stationary or dynamic leaders under a directed interaction graph. In *48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 3014–3019.

Jorge C. S. Cardoso, Carlos Baquero, and Paulo Sérgio Almeida. 2009. Probabilistic Estimation of Network Size and Diameter. In *4th Latin-American Symposium on Dependable Computing*, pages 33–40, João Pessoa, Brazil, September 2009. IEEE Computer Society.

Ruggero Carli, Fabio Fagnani, Paolo Frasca, and Sandro Zampieri. 2010. Gossip consensus algorithms via quantized communication. *Automatica*, 46(1):70–80.

George Casella and Roger L. Berger. 2002. *Statistical inference*. Thomson Learning.

Vinton G. Cerf and Robert E. Kahn. 1974. A Protocol for Packet Network Intercommunication. *IEEE Transactions on Communications*, 22(5):637–648.

Yousra Chabchoub and Georges Hébrail. 2010. Sliding HyperLogLog: Estimating Cardinality in a Data Stream over a Sliding Window. In *10th IEEE International Conference on Data Mining Workshops*, pages 1297–1303, Sydney, Australia, December 2010.

Philippe Chassaing and Lucas Gerin. 2006. Efficient estimation of the cardinality of large data sets. In *4th Colloquium on Mathematics and Computer Science Algorithms, Trees, Combinatorics and Probabilities*, pages 419–422, France. IECN.

Siyao Cheng, Jianzhong Li, Qianqian Ren, and Lei Yu. 2010. Bernoulli Sampling Based (ε, δ)-Approximate Aggregation in Large-Scale Sensor Networks. In *29th IEEE International Conference on Computer Communications*, pages 1181–1189, San Diego, CA, March 2010. IEEE Press.

Jacek Cichon, Jakub Lemiesz, Wojciech Szpankowski, and Marcin Zawada. 2012a. Two-phase cardinality estimation protocols for sensor networks with provable precision. In *IEEE Wireless Communications and Networking Conference*, pages 2009–2013.

Jacek Cichon, Jakub Lemiesz, and Marcin Zawada. 2011. On Cardinality Estimation Protocols for Wireless Sensor Networks. In *10th International Conference on Ad-hoc, Mobile, and Wireless Networks*, pages 322–331, Paderbom, Germany, July 2011. Springer Berlin Heidelberg.

Jacek Cichon, Jakub Lemiesz, and Marcin Zawada. 2012b. On Message Complexity of Extrema Propagation Techniques. In *11th International Conference on Ad-hoc, Mobile, and Wireless Networks*, pages 1–13, Belgrade, Serbia, July 2012. Springer-Verlag.

Frank H. Clarke, Yu S. Ledyaev, Ronald J. Stern, and Peter R. Wolenski. 1998. *Nonsmooth Analysis and Control Theory*. Springer-Verlag.

Peter Clifford and Ioana A. Cosma. 2012. A Statistical Analysis of Probabilistic Counting Algorithms. *Scandinavian Journal of Statistics*, 39:1–14.

Edith Cohen. 1997. Size-Estimation Framework with Applications to Transitive Closure and Reachability. *Journal of Computer and System Sciences*, 55(3): 441–453.

Jorge Cortes and Francesco Bullo. 2005. Coordination and Geometric Optimization via Distributed Dynamical Systems. *SIAM Journal on Control and Optimization*, 44(5).

Iain D. Couzin, Jens Krause, Nigel R. Franks, and Simon A. Levin. 2005. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025): 513–516.

John M. Danskin. 1966. The Theory of Max-Min, with Applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664.

Herbert A. David and H. N. Nagaraja. 2004. *Order Statistics*. John Wiley & Sons.

Reinhard Diestel. 2005. *Graph theory*. Springer Verlag.

Danny Dolev, Osnat Mokryn, and Yuval Shavitt. 2006. On Multicast Trees: Structure and Size Estimation. *IEEE/ACM Transactions on Networking*, 14(3):557–567.

Fabio Fagnani and Sandro Zampieri. 2008. Randomized Consensus Algorithms over Large Scale Networks. *IEEE Journal on Selected Areas in Communications*, 26 (4):634–649.

Nicolas Falliere, Liam O Murchu, and Eric Chien. 2011. W32.Stuxnet Dossier. Technical report, Symantec.

Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier. 2007. HyperLogLog : the analysis of a near-optimal cardinality estimation algorithm. In *Conference on Analysis of Algorithms*, pages 127–146, Juan des Pins, France.

Pedro A. Forero, Alfonso Cano, and Georgios B. Giannakis. 2008. Consensus-based distributed expectation-maximization algorithm for density estimation and classification using wireless sensor networks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1989–1992, Las Vegas, NV, USA.

Éric Fusy and Frédéric Giroire. 2007. Estimating the Number of Active Flows in a Data Stream over a Sliding Window. In *Workshop on Analytic Algorithmics and Combinatorics*.

Federica Garin, Damiano Varagnolo, and Karl Henrik Johansson. 2012. Distributed estimation of diameter, radius and eccentricities in anonymous networks. In *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*.

Frédéric Giroire. 2009. Order statistics and estimating cardinalities of massive data sets. *Discrete Applied Mathematics*, 157(2):406–427.

Christos Gkantsidis, Milena Mihail, and Amin Saberi. 2006. Random walks in peer-to-peer networks: Algorithms and evaluation. *Performance Evaluation*, 63 (3):241–263.

Michael B. Greenwald and Sanjeev Khanna. 2004. Power-Conserving Computation of Order-Statistics over Sensor Networks. In *23rd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 275–285, Paris, France, June 2004. ACM Press.

Dongbing Gu and Zongyao Wang. 2009. Leader–Follower Flocking: Algorithms and Experiments. *IEEE Transactions on Control Systems Technology*, 17(5): 1211–1219.

Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble. 2002. King: Estimating Latency between Arbitrary Internet End Hosts. In *2nd ACM SIGCOMM Workshop on Internet Measurement*, pages 5–18.

Meng Guo, Dimos V. Dimarogonas, and Karl Henrik Johansson. 2012. Distributed real-time fault detection and isolation for cooperative multi-agent systems. In *American Control Conference, 2012*, pages 5270–5275.

Tove Gustavi, Dimos V. Dimarogonas, Magnus B. Egerstedt, and Xiaoming Hu. 2010. Sufficient conditions for connectivity maintenance and rendezvous in leader–follower networks. *Automatica*, 46(1):133–139.

Maya Haridasan and Robbert van Renesse. 2008. Gossip-based Distribution Estimation in Peer-to-Peer Networks. In *7th International Workshop on Peer-to-Peer Systems*.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*, 2 edition. Data Mining, Inference, and Prediction. Springer.

Rainer Hegselmann and Ulrich Krause. 2002. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artifical Societies and Social Simulation*, 5(3).

Ryan Herring, Aude Hofleitner, Saurabh Amin, Tania Abou Nasr, Amin Abdel Khalek, Pieter Abbeel, and Alexandre Bayen. 2009. Using Mobile Phones to Forecast Arterial Traffic through Statistical Learning. In *89th Annual Meeting of the Transportation Research Board*.

Yu-Chi Ho, Arthur E. Bryson, and Sheldon Baron. 1965. Differential games and optimal pursuit-evasion strategies. *IEEE Transactions on Automatic Control*, 10 (4):385–389.

Yiguang Hong, Jiangping Hu, and Linxin Gao. 2006. Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica*, 42(7): 1177–1182.

Shafiul Azam Howlader, Michael R. Frater, and Michael J. Ryan. 2008. Estimating the Number and Distribution of the Neighbors in an Underwater Communication Network. In *Second International Conference on Sensor Technologies and Applications*, pages 693–698.

Yusuo Hu, Jian-Guang Lou, Hua Chen, and Jiang Li. 2007. Distributed Density Estimation Using Non-parametric Statistics. In *27th IEEE International Conference on Distributed Computing Systems*, Toronto, ON.

Franck Iutzeler, Philippe Ciblat, and Jérémie Jakubowicz. 2012. Analysis of Max-Consensus Algorithms in Wireless Channels. *IEEE Transactions on Signal Processing*, 60(11):6103–6107.

Ali Jadbabaie, Jie Lin, and A. Stephen Morse. 2003. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001.

Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu. 2009. T-Man: Gossip-based fast overlay topology construction. *Computer networks*, 53(13):2321–2339.

Márk Jelasity, Spyros Voulgaris, Rachid Guerraoui, Anne-Marie Kermarrec, and Maarten van Steen. 2007. Gossip-based peer sampling. *ACM Transactions on Computer Systems*, 25(3).

Erik Jenelius and Haris N. Koutsopoulos. 2013. Travel time estimation for urban road networks using low frequency probe vehicle data. *Transportation Research Part B: Methodological*, 53:64–81.

Paulo Jesus. 2012. *Robust Distributed Data Aggregation*. PhD thesis, Universidade do Minho.

Meng Ji, Giancarlo Ferrari-Trecate, Magnus B. Egerstedt, and Annalisa Buffa. 2008. Containment Control in Mobile Networks. *IEEE Transactions on Automatic Control*, 53(8):1972–1975.

Meng Ji, Abubakr Muhammad, and Magnus B. Egerstedt. 2006. Leader-based multi-agent coordination: controllability and optimal control. In *American Control Conference, 2006*.

Hongbo Jiang and Shudong Jin. 2006. Scalable and Robust Aggregation Techniques for Extracting Statistical Information in Sensor Networks. In *26th International Conference on Distributed Computing Systems*, Lisboa, Portugal, July 2006.

Anne-Marie Kermarrec and Maarten van Steen. 2007. Gossiping in Distributed Systems. *ACM SIGOPS Operating Systems Review*, 41(5):2–7.

Matthias Klusch, Stefano Lodi, and Gianluca Moro. 2003. Distributed Clustering Based on Sampling Local Density Estimates. In *18th International Joint Conference on Artificial Intelligence*, pages 485–490.

Dionysios Kostoulas, Dimitrios Psaltoulis, Indranil Gupta, Kenneth P. Birman, and Alan J. Demers. 2007. Active and passive techniques for group size estimation in large-scale and dynamic distributed systems. *Journal of Systems and Software*, 80(10):1639–1658.

Ulrich Krause. 2006. Convergence of the multidimensional agreement algorithm when communication fades away. In *2nd Multidisciplinary International Symposium on Positive Systems: Theory and Applications*, pages 217–222.

Gunnar Kreitz and Fredrik Niemelä. 2010. Spotify - Large Scale, Low Latency, P2P Music-on-Demand Streaming. In *10th IEEE International Conference on Peer-to-Peer Computing*, Delft, Netherlands.

Renaud Lambiotte, Vincent D. Blondel, Cristobald de Kerchove, Etienne Huens, Christophe Prieur, Zbigniew Smoreda, and Paul Van Dooren. 2008. Geographical dispersal of mobile communication networks. *Physica A*, 387(21):5317–5325.

Amir Leshem and Lang Tong. 2005. Estimating Sensor Population via Probabilistic Sequential Polling. *IEEE Signal Processing Letters*, 12(5):395–398.

Zhiyun Lin, Bruce Francis, and Manfredi Maggiore. 2005. Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Transactions on Automatic Control*, 50(1):121–127.

Zhiyun Lin, Bruce Francis, and Manfredi Maggiore. 2007. State Agreement for Continuous-Time Coupled Nonlinear Systems. *SIAM Journal on Control and Optimization*, 46(1):288–307.

Ji Liu, Brian D. O. Anderson, Ming Cao, and A. Stephen Morse. 2009. Analysis of accelerated gossip algorithms. In *48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 871–876, Shanghai. IEEE.

Jérémie Lumbroso. 2010. An optimal cardinality estimation algorithm based on order statistics and its full analysis. In *21st International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms*, pages 489–504.

Kevin M. Lynch, Ira B. Schwartz, Peng Yang, and Randy A. Freeman. 2008. Decentralized Environmental Modeling by Mobile Sensor Networks. *IEEE Transactions on Robotics*, 24(3):710–724.

Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. 2002. TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks. In *5th Symposium on Operating Systems Design and Implementation*, pages 131–146.

Laurent Massoulié, Erwan Le Merrer, Anne-Marie Kermarrec, and Ayalvadi Ganesh. 2006. Peer Counting and Sampling in Overlay Networks: Random Walk Methods. In *25th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, pages 123–132. ACM Request Permissions.

Ziyang Meng, Zongli Lin, and Wei Ren. 2012. Leader–follower swarm tracking for networked Lagrange systems. *Systems & Control Letters*, 61(1):117–126.

Milena Mihail, Christos Papadimitriou, and Amin Saberi. 2003. On Certain Connectivity Properties of the Internet Topology. In *44th Annual IEEE Symposium on Foundations of Computer Science*, pages 28–35, Washington, DC, USA, October 2003. IEEE Computer Society.

Damon Mosk-Aoyama and Devavrat Shah. 2008. Fast Distributed Algorithms for Computing Separable Functions. *IEEE Transactions on Information Theory*, 54 (7):2997–3007.

Shinji Motegi, Kiyohito Yoshihara, and Hiroki Horiuchi. 2006. DAG based In-Network Aggregation for Sensor Network Monitoring. In *International Symposium on Applications and the Internet*, pages 292–299, Phoenix, AZ, USA.

Angelia Nedić and Behrouz Touri. 2012. Multi-Dimensional Hegselmann-Krause Dynamics. In *51st IEEE Conference on Decision and Control*, pages 68–73, Maui, Hawaii, USA, December 2012.

XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. 2005. Nonparametric decentralized detection using kernel methods. *IEEE Transactions on Signal Processing*, 53(11):4053–4066.

Robert Nowak. 2003. Distributed EM Algorithms for Density Estimation and Clustering in Sensor Networks. *IEEE Transactions on Signal Processing*, 51(8):2245–2253.

Reza Olfati-Saber and Richard M. Murray. 2004. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533.

Reza Olfati-Saber and Jeff S. Shamma. 2005. Consensus Filters for Sensor Networks and Distributed Sensor Fusion. In *44th IEEE Conference on Decision and Control and European Control Conference*, pages 6698–6703.

Alex Olshevsky and John N. Tsitsiklis. 2006. Convergence Rates in Distributed Consensus and Averaging. In *45th IEEE Conference on Decision and Control*, pages 3387–3392.

Amir H. Payberah, Jim Dowling, Fatemeh Rahimian, and Seif Haridi. 2010a. *gradienTv*: Market-Based P2P Live Media Streaming on the Gradient Overlay. *Distributed Applications and Interoperable Systems*, 6115:212–225.

Amir H. Payberah, Fatemeh Rahimian, Seif Haridi, and Jim Dowling. 2010b. Sepidar: Incentivized Market-Based P2P Live-Streaming on the Gradient Overlay Network. In *12th IEEE International Symposium on Multimedia*, pages 1–8.

Shao-Liang Peng, Shan-Shan Li, Xiang-Ke Liao, Yu-Xing Peng, and Nong Xiao. 2009. Estimation of a Population Size in Large-Scale Wireless Sensor Networks - Springer. *Journal of Computer Science and Technology*, 24(5):987–997.

Sanja Petrovic and Patrick Brown. 2009. A New Statistical Approach to Estimate Global File Populations in the eDonkey P2P File Sharing System. In *21st International Teletraffic Congress*, pages 1–8, Paris, France.

Dimitrios Psaltoulis, Dionysios Kostoulas, Indranil Gupta, Kenneth P. Birman, and Alan J. Demers. 2004. Practical algorithms for Size estimation in Large and Dynamic groups. Technical report, University of Illinois, Urbana-Champaign.

Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. 2007. *Numerical Mathematics*. Springer Verlag.

Amirreza Rahmani, Meng Ji, Mehran Mesbahi, and Magnus B. Egerstedt. 2009. Controllability of Multi-Agent Systems from a Graph-Theoretic Perspective. *SIAM Journal on Control and Optimization*, 48:162–186.

Amirreza Rahmani and Mehran Mesbahi. 2006. On the controlled agreement problem. In *American Control Conference, 2006.*

Wei Ren and Randal W. Beard. 2008. *Distributed Consensus in Multi-Vehicle Cooperative Control.* Theory and Applications. Springer.

Bruno Ribeiro and Don Towsley. 2010. Estimating and Sampling Graphs with Multidimensional Random Walks. In *10th ACM SIGCOMM Conference on Internet Measurement*, pages 390–403, Melbourne, Australia, November 2010.

Gordon F. Royle. 2001. *Algebraic Graph Theory.* New York : Springer.

Jan Sacha. 2009. *Exploiting Heterogeneity in Peer-to-Peer Systems using Gradient Topologies.* PhD thesis, Trinity College, Dublin.

Jan Sacha, Jeff Napper, Corina Stratan, and Guillaume Pierre. 2010. Adam2: Reliable Distribution Estimation in Decentralised Environments. In *30th International Conference on Distributed Computing Systems*, pages 697–707.

Bernhard Schölkopf and Alexander J. Smola. 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press.

Tallat M. Shafaat, Ali Ghodsi, and Seif Haridi. 2008. A Practical Approach to Network Size Estimation for Structured Overlays. In *Self-organizing Systems*, pages 71–83. Springer Berlin Heidelberg.

Iman Shames, André Teixeira, Henrik Sandberg, and Karl Henrik Johansson. 2012. Agents Misbehaving in a Network: A Vice or a Virtue? *IEEE Network*, 26(3): 35–40.

Guodong Shi and Yiguang Hong. 2009. Global target aggregation and state agreement of nonlinear multi-agent systems with switching topologies. *Automatica*, 45 (5):1165–1175.

Guodong Shi, Yiguang Hong, and Karl Henrik Johansson. 2012. Connectivity and Set Tracking of Multi-Agent Systems Guided by Multiple Moving Leaders. *IEEE Transactions on Automatic Control*, 57(3):663–676.

Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. 2004. Medians and Beyond: New Aggregation Techniques for Sensor Networks. In *2nd International Conference on Embedded Networked Sensor Systems*, pages 239–249.

Herbert G. Tanner. 2004. On the controllability of nearest neighbor interconnections. In *43rd IEEE Conference on Decision and Control*, pages 2467–2472.

Herbert G. Tanner, Ali Jadbabaie, and George J. Pappas. 2007. Flocking in Fixed and Switching Networks. *IEEE Transactions on Automatic Control*, 52(5):863–868.

Herbert G. Tanner, George J. Pappas, and Vijay Kumar. 2004. Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3):443–455.

André Teixeira, Daniel Pérez, Henrik Sandberg, and Karl Henrik Johansson. 2012. Attack models and scenarios for networked control systems. In *1st ACM International Conference on High Confidence Networked Systems*, pages 55–64, Beijing, China, April 2012.

Håkan Terelius, Guodong Shi, Jim Dowling, Amir H. Payberah, Ather Gattami, and Karl Henrik Johansson. 2011a. Converging an Overlay Network to a Gradient Topology. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 7230–7235, Orlando, FL, USA, December 2011.

Håkan Terelius, Guodong Shi, and Karl Henrik Johansson. 2013a. Consensus Control for Multi-agent Systems with a Faulty Node. In *4th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 425–432, Koblenz, Germany, September 2013.

Håkan Terelius, Ufuk Topcu, and Richard M. Murray. 2011b. Decentralized Multi-Agent Optimization via Dual Decomposition. In *18th IFAC World Congress*, pages 11245–11251, Milan, Italy, August 2011.

Håkan Terelius, Damiano Varagnolo, Carlos Baquero, and Karl Henrik Johansson. 2013b. Fast Distributed Estimation of Empirical Mass Functions over Anonymous Networks. In *52nd IEEE Conference on Decision and Control*, Florence, Italy. To appear.

Håkan Terelius, Damiano Varagnolo, and Karl Henrik Johansson. 2012. Distributed Size Estimation of Dynamic Anonymous Networks. In *51st IEEE Conference on Decision and Control*, pages 5221–5227, Maui, HI, USA, December 2012.

Jeffrey Travers and Stanley Milgram. 1969. An Experimental Study of the Small World Problem. *Sociometry*, 32(4):425–443.

John N. Tsitsiklis, Dimitri P. Bertsekas, and Michael Athans. 1986. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812.

Vladimir Naumovich Vapnik. 1998. *Statistical learning theory*. Wiley-Interscience, New York.

Damiano Varagnolo, Gianluigi Pillonetto, and Luca Schenato. 2010. Distributed Statistical Estimation of the Number of Nodes in Sensor Networks. In *49th IEEE Conference on Decision and Control*, pages 1498–1503.

Damiano Varagnolo, Gianluigi Pillonetto, and Luca Schenato. 2013. Distributed Cardinality Estimation in Anonymous Networks. *IEEE Transactions on Automatic Control*.

Nikos Vlassis, Yiannis Sfakianakis, and Wojtek Kowalczyk. 2005. Gossip-Based Greedy Gaussian Mixture Learning. In *Advances in Informatics*, pages 349–359. Springer Berlin Heidelberg, Berlin, Heidelberg.

Grace Wahba. 1990. *Spline Models for Observational Data*. SIAM.

Viktor Witkovský. 2001. Computing the distribution of a linear combination of inverted gamma variables. *Kybernetika*, 37(1):79–90.

Lin Xiao and Stephen P. Boyd. 2003. Fast linear iterations for distributed averaging. In *42nd IEEE Conference on Decision and Control*, pages 4997–5002.

Masafumi Yamashita and Tiko Kameda. 1988. Computing on an anonymous network. *7th Annual ACM Symposium on Principles of Distributed Computing*, pages 117–130.

Ye Yuan, Guy-Bart Stan, Mauricio Barahona, Ling Shi, and Jorge Gonçalves. 2011. Decentralised minimal-time consensus. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 4282–4289, Orlando, FL, December 2011.