

INFORMATION DIFFUSION IN COMPLEX NETWORKS VIA GOSSIPING

A Dissertation

Presented to the Faculty of the Graduate School
of Cornell University

in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by

Mehmet Ercan Yildiz

May 2010

© 2010 Mehmet Ercan Yıldız

ALL RIGHTS RESERVED

INFORMATION DIFFUSION IN COMPLEX NETWORKS VIA GOSSIPING

Mehmet Ercan Yildiz, Ph.D.

Cornell University 2010

With the rapid developments in hardware and software technology, so called networked systems have expanded significantly in the recent years. Due to their significant advantages, these networks grow continuously in size (number of agents, links) and complexity. However, this enormous growth brings potential problems, *i.e.*, complexity issues and failure consequences. For this reason, in this study, we focus on *efficient, scalable* and *reliable* information diffusion algorithms on complex networks. We first focus on so called average consensus problem under finite rate communications. We utilize increasing spatial and correlation among node states to reduce quantization error in the system and propose coding with side information schemes for quantized consensus algorithms. We analyze the convergence behavior of the quantized consensus as well as analytically modeling the mean squared error of the algorithm.

Moreover, we propose a gossiping algorithm, *i.e., broadcast gossiping*, for consensus type problems on sensor networks. Similar to other gossiping algorithms, our scheme generates only local traffic and robust to link/node failures due to the iterative update structure. The main advantage of the algorithm is the fact that our updates rely on the wireless nature of the links in between nodes rather than point to point communications. We further analyze performance characteristics of our algorithm such as the speed of convergence and mean squared error.

We also propose a gossiping based scheme for asymmetric information dif-

fusion problem. In this particular problem, a subset of the network is interested in a separable function of the data which is stored in another subsets of the nodes. Given the underlying network connectivity and the source-destination sets, we provide necessary and sufficient conditions on the update weights (referred to as codes) so that the destination nodes converge to the desired function of the source values. We show that the evolution of the source states does not affect the feasibility of the problem, and we provide a detailed analysis on the spectral properties of the feasible codes. We also study the problem feasibility under some specific topologies and provide guidelines to determine infeasibility. We also formulate different strategies to design codes, and compare the performance of our solution with existing alternatives.

Finally, we propose a gossiping based method for modeling opinion propagation through social networks. In particular, we model the individuals' opinions as binary variables (such as 0 or 1, democrat or republican, etc), and assume that there exists certain individuals in society who do not change their opinions. We analyze the behavior of individual's opinions as well as the collective opinion of the society in the long run. We completely characterize long term mean and variance of the average opinion in the networks as well as posing optimal stubborn agent replacement problem.

BIOGRAPHICAL SKETCH

Mehmet Ercan Yildiz was born in Istanbul, Turkey in 1982. He graduated from Ankara Science High School in 2000, and attended the Middle East Technical University from 2000 to 2001. In 2001, he transferred to University of Wisconsin-Madison where he received his B.S. degree in Electrical and Computer Engineering and his B.A. degree in Mathematics. After his graduation from UW-Madison, he attended Cornell University and received his M.S degree in 2007. Since Fall 2007, he has been a PhD candidate at Cornell University. His research interests include social networks, distributed/ decentralized signal processing, sensor networks, and consensus algorithms.

Dedicated to my love, wife and friend Kamer...

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Prof. Anna Scaglione, for her endless support and patience during the past five years. I would not be where I am now without her guidance. She has always been open to new ideas and has treated me as a colleague rather than a student since the first day of my studies.

I also would like to thank Prof. Lang Tong and Prof. Kevin Tang, for serving as the committee members and being good mentors. I have always benefited from their words of wisdom.

I am grateful to my office mates; Roberto, Matt, Birsen, Azadeh and Shrut, for sharing both the joy and stress of my graduate studies.

I would like to thank Prof. Robert Nowak, for being a great mentor during my junior and senior years at UW - Madison, and for helping me to realize that "PhD" was what I wanted. I also would like to thank 3610; Rui, Yolanda, Mike, Becca and Aline for their friendship and guidance.

I am thankful to Asuman Ozdaglar for introducing me to a new world of research in the last year of my studies, and for joyful and productive discussions.

A special thanks must go to Dr. Tuncer Can Aysal for being such a close friend. He has always been (and he still is) right next to me both as a friend and a colleague.

I also would like to thank Yasin Alan, Aysima-Deniz-Kubilay Kavak, Ilkay-Hasan Onogul, Aydin and Sinan Beseli for making my all these years enjoyable by being great friends.

I would like to express my deep gratitude to my parents for being right beside me in all my life and for supporting every decision I have given. I owe my achievements to their constant encouragement, support, and understanding.

I would like to share this moment of happiness with my sweet sister Elif as we have shared countless joyful moments since we were small kids.

I would also like to thank my in-laws, Bilge, Mustafa and Kartal Toker, who, barely a week into my marriage, allowed me to take their daughter so far away from them. They have supported me by all means since we first met.

Finally, I am grateful to my wife, my love, and my best friend Kamer. She has been my inspiration and my hope even during the times when there was an ocean between us. Without her loving and caring support, patience, and encouragement, I would have never completed this work.

TABLE OF CONTENTS

Biographical Sketch	iii
Dedication	iv
Acknowledgements	v
Table of Contents	vii
List of Tables	xi
List of Figures	xii
1 Introduction	1
1.1 Information Diffusion in Networks	1
1.2 Coding and Access Protocols for Scalable Wireless Average Con- sensus	2
1.2.1 Quantized Average Consensus	2
1.2.2 Broadcast Gossiping Algorithms	7
1.3 Generalized Network Computation via Gossiping	9
1.4 Opinion Diffusion in Social Networks via Gossiping	12
2 Coding with Side Information for Quantized Average Consensus	14
2.1 Motivation and Related Work	14
2.1.1 Summary of Main Contributions	16
2.1.2 Chapter Organization	17
2.2 Quantized Average Consensus Model	17
2.3 Convergence Conditions With Additive Quantization Noise	23
2.4 Predictive Coding	27
2.4.1 Broadcast Predictive Coding	28
2.4.2 Analytical Framework	30
2.4.3 Algorithm Summary	32
2.4.4 Asymptotic Rate Behavior for Predictive Coding	33
2.4.5 Feasible Rate Allocation for Bounded Convergence	34
2.4.6 Peer to Peer Predictive Coding	35
2.5 Wyner-Ziv Coding	36
2.5.1 Broadcast WZ Coding	36
2.6 Numerical Results	42
2.7 Discussions	46
3 MSE Characterization for Quantized Average Consensus for Fixed and Variable Rate Quantizers	48
3.1 Motivation and Related Work	48
3.1.1 Summary of Contributions	48
3.1.2 Chapter Organization	50
3.2 Quantized Average Consensus	50
3.3 Characterization of the MSE Performance	53
3.3.1 2-Dimensional Regular Networks	54

3.3.2	Mean Square Error	56
3.3.3	Scaling Laws in Regular Networks	63
3.4	Asymptotic Rate Behavior and Rate Allocation	65
3.4.1	Asymptotic Rate Behavior	66
3.4.2	Rate Allocation	71
3.5	Discussions	73
4	Broadcast Gossiping Algorithm	74
4.1	Motivation and Related Work	74
4.1.1	Summary of Main Contributions	74
4.1.2	Chapter Organization	76
4.2	Graph and Time Models	77
4.2.1	Graph Model	77
4.2.2	Time Model	77
4.2.3	Average Consensus	78
4.3	Broadcast Based Gossiping	78
4.4	Convergence of Broadcast Gossiping	82
4.4.1	Convergence in the Expectation	82
4.4.2	Convergence in the Second Moment	83
4.4.3	Almost Sure Convergence to Consensus	86
4.5	Optimal Mixing Parameter	87
4.6	Performance Analysis of Broadcast Gossip Algorithms	91
4.6.1	Mean Square Error	91
4.6.2	Communication Cost to Achieve Consensus	94
4.6.3	Performance Analysis: Numerical Examples	96
4.7	Discussions	102
5	Computing Along the Routes with Gossiping	103
5.1	Motivation and Related Work	103
5.1.1	Chapter Organization	107
5.2	Problem Formulation	107
5.3	Necessary and Sufficient Conditions on Feasible Codes	110
5.4	Spectral Analysis and Topology Reductions	115
5.4.1	Spectral Analysis of Feasible Codes	115
5.4.2	Reduction of Topology to Study Feasibility	120
5.5	A Necessary Condition On the Topology	122
5.6	Construction of partially directed AVT solutions	126
5.7	Complexity and Communication cost of AVT codes	131
5.7.1	Complexity	131
5.7.2	Communication cost	132
5.8	Simulations	134
5.9	Extension to Dynamic Networks	136
5.10	Discussions	143

6	Opinion Diffusion in Social Networks via Gossiping	146
6.1	Introduction	146
6.1.1	Motivation and Related Work	146
6.1.2	Chapter Organization	148
6.2	The Voter Model and the Dual Approach	148
6.2.1	The Binary Voter Model with Stubborn Agents	148
6.2.2	Effects of Stubborn Agents: From Consensus to Disagree- ment	150
6.2.3	The Dual Approach for the Voter Model with Stubborn Nodes	154
6.3	Characterization of the Average Opinion: Mean and Variance . .	157
6.3.1	Expected Value of the Average Opinion	158
6.3.2	Variance of the Average Opinion	163
6.4	Optimal placement of stubborn agents	166
6.4.1	A special case: $k=1$	167
6.4.2	The general case: $k \geq 2$	169
6.5	Discussions	170
A	Appendix of Chapter 2	171
A.1	Proof of Lemma1	171
A.2	Calculation of $\mathbb{E}\{\tilde{z}(t-l)\tilde{z}^T(t-m)\}$	173
A.3	Calculation of $\mathbb{E}\{z(t)\tilde{z}^T(t-m)\}$	174
B	Appendix of Chapter 3	175
B.1	Proof of Lemma 9	175
B.2	Proof Lemma 10	177
B.3	Proof of Corollary 2	177
B.4	Proof Lemma 11	178
B.5	Proof of Equation (3.30)	180
B.6	Proof Lemma 12	181
B.7	Proof of Lemma 13	182
B.8	Proof of Lemma 14	184
B.9	Proof of Lemma 15	185
C	Appendix of Chapter 4	189
C.1	Proof of Lemma 17	189
C.2	Proof of Theorem 2	189
C.3	Proof of Lemma 19	190
C.4	Proof of Lemma 20	193
C.5	Proof of Proposition 3	194
C.6	Proof of Corollary 6	197
C.7	Proof of Lemma 21	198

D Appendix of Chapter 5	200
D.1 Proof of Lemma 22	200
D.2 Proof of Lemma 23	200
D.3 Proof of Lemma 24	202
D.4 Proof of Lemma 26	203
D.5 Proof of Lemma 27	204
D.6 Proof of Lemma 28	206
D.7 Proof of Lemma 29	209
 Bibliography	 214

LIST OF TABLES

2.1	Cross Correlations of Noisy States	31
-----	--	----

LIST OF FIGURES

2.1	Charts for different quantization schemes.	18
2.2	Differential encoder/decoder diagram.	28
2.3	Theoretical performance comparison between WZ and Predictive coding, $N = 10, r = 0.4$	41
2.4	$r = 0.4, N = 50$, Total number of iterations = 65.	43
2.5	Simulated performance vs. theoretical performance.	44
2.6	Rate distortion curves for WZ and predictive coding schemes over 50 iterations	45
2.7	MSE performance of the predictive coding for different p values.	46
3.1	Differential encoder/decoder diagram with dithering.	51
3.2	Simulated versus theoretical MSE performance of QCPC.	61
3.3	QCPC MSE (dB) performance for random and regular networks.	62
3.4	Network size versus MSE performance of QCPC.	63
3.5	QCPC MSE versus connectivity radius.	65
3.6	MSE (dB) vs sum of the quantization rates for different β over 1000 iterations.	71
3.7	MSE (dB) vs sum of the quantization rates for different β	72
4.1	Number of radio transmissions required to achieve a given distance (per node variance) from the consensus for $N \in \{50, 100, 500\}$ with initial node values uniformly distributed.	98
4.2	Number of radio transmissions required to achieve a given distance (per node variance) from the consensus for $N \in \{50, 100, 500\}$ with initial node values zero except one in a single node.	99
4.3	The MSE performance of the randomized, geographic and broadcast gossip algorithms with respect to the number of radio transmissions for $N = 500$ with initial node values uniformly distributed.	100
4.4	The MSE and Variance performances of the broadcast gossip algorithms with respect to γ for $N = 500$ with initial node values uniformly distributed.	101
5.1	An infeasible scenario with two sources and two destinations.	119
5.2	Two networks corresponding to the same underlying connectivity but two different network codes, W_1 and W_2	120
5.3	Line network topology. S and D denotes source nodes and destinations nodes respectively.	125
5.4	Mesh network topology. S and D denotes source nodes and destinations nodes respectively.	126
5.5	A directed solution. S and D represents source and destination nodes respectively.	130

5.6	The communication costs of directed AVT.	135
5.7	Empirical Results.	144
6.1	A sample update path for the voter algorithm. x-axis denotes node index and y-axis denotes continuous time index.	156
D.1	Source clustering due to W^* . Dashed line between S_2 and S_3 represents zero weight link. Cloud represents rest of the network.	204
D.2	A network with two partitions and a single cut edge.	207

CHAPTER 1

INTRODUCTION

1.1 Information Diffusion in Networks

With the improvement of hardware technology and emergence of fiber-optic infrastructure, so called networked systems have expanded significantly. These systems, in most of the cases, consist of *static or mobile agents* which can collect, store and process data, and *communication links* which can carry information in between these agents. Networked systems have improved the quality of our lives and become an integral part of our society. For instance, the Internet which has almost 2 billions users today [87], is being utilized for e-mails, instant messaging, VOIP communications, social networking, online banking, etc. Moreover, sensor networks which rely on wireless communications in between agents, have been deployed for the detection of fire, of anomalies and of intruders in urban environments, for climate control in houses/apartments, for data collection in various applications as well as cell phone communications [3].

Due to their significant advantages, these networks grow continuously in size (number of agents, links) and complexity. However, this enormous growth brings potential problems, *i.e.*, complexity issues and failure consequences. In other words, the complexity of queries, data processing, data retrieval and network structure increases which results in more vulnerable networked systems. On the other, we become more and more dependent on these networks which brings out the issue of them becoming *too important to fail*.

For the reasons discussed above, in this study, we focus on *reliable, scalable*

and *efficient* information diffusion algorithms on complex networks. Complex nature of these networks is due to their non-trivial topological features. The first three chapters of the study, Chapters 2 – 4, will be on symmetric diffusion problems on wireless networks, while in Chapter 5, asymmetric diffusion in wired networks will be of our interest. Finally, in Chapter 6, we will focus on opinion propagation through social networks via gossiping.

1.2 Coding and Access Protocols for Scalable Wireless Average Consensus

In the first three chapters of the study, we will focus on the average consensus problem in wireless sensor networks. Our primary motivation is that, in wireless networks, there is no real link, and one can realize different network topologies with the same power and bandwidth resources depending on how it encodes the data (quantization) and how it structures the access. For instance, if one is willing to sacrifice the precision it can connect at a longer range by utilizing lower quantization rates. Moreover, if one is willing to broadcast rather than employing point to point communication, it can perform more exchanges at once. Keeping these points in my mind, we will investigate coding and access protocols for average consensus problems on wireless networks.

1.2.1 Quantized Average Consensus

In particular, in Chapters 2 – 3, we will focus on so called average consensus problems. In this special class of problems, each agent i in the network

has an initial quantity $x_i(0) \in \mathbb{R}$, and the network is interested in calculating the average of these initial quantities, *i.e.*, $1/N \sum_{i=1}^N x_i(0)$ [88, 22, 47]. Consensus problems, in general, have many practical applications including coordination of autonomous and geographically separated field agents, distributed computing and congestion control, tracking objects by several unmanned air vehicles (UAVs) and decentralized reconstruction or compression of a field [48, 75, 39, 83, 52, 94].

To calculate the average, we will utilize the so called gossiping protocols which are iterative and based on near neighbor communications. These algorithms are known to generate local traffic only and to be robust to link/node failures due to their iterative nature [88, 19, 21, 47, 62]. These protocols have been first introduced by Tsitsiklis in his PhD thesis [88], and have gained significant attention after Boyd *et.al.*'s work in [21]. Boyd *et.al.* have analyzed discrete time update case and studied wide range of problems from convergence characteristics to the design of optimal update weights for faster convergence [92, 19, 20]. Saber and Murray have analyzed the case of continuous time updates with a dynamically changing network topology [82]. Moreover, they have explored the necessary conditions on the update weights and connectivity of the sensors for convergence. Ren *et.al.* have studied consensus algorithms for the solution of Kalman filtering context [80]. Rabbat *et.al.* have focused on optimization consensus problem with binary erasure links between neighboring nodes [79]. Interested readers may refer to consensus literature for detailed analysis as well as several interesting applications [19, 20, 93, 55, 71, 72, 50].

As we have discussed above, there exists a substantial body of work on average consensus protocols under infinite precision and noiseless peer to peer

communications. However, little research has been done introducing distortions in the message exchange. In general, the networks envisioned for the application of consensus algorithms involve large numbers of possibly randomly distributed inexpensive sensors, with limited sensing, processing and communication power on board. In many of the applications, limitations in bandwidth and sensor battery power and computing resources place tight constraints in the transmission rate. Other applications such as camera networks and distributed tracking demand communication of large volumes of data. When the power and bandwidth constraints, or large volume data sets are considered, neglecting rate constraints is unrealistic.

Xiao and Boyd have studied an average consensus algorithm where each update is corrupted by an additive noise with zero mean and *fixed* variance. They have concluded that consensus is only achievable in the mean¹ and the mean squared difference between the initial average and asymptotic average is unbounded [93]. On the other hand, recent work on quantized average consensus by Aysal *et.al*, has modeled the quantization noise more accurately and suggests that convergence can be attained *almost surely* under a probabilistic quantization scheme [11, 12, 10]. More specifically, the authors have proposed a scheme where each node updates its state as a linear combination of its own quantized state and the probabilistically quantized states of its neighbors ($\tilde{x}(t)$), *i.e.*:

$$x(t+1) = W\tilde{x}(t). \quad (1.1)$$

To prove the convergence to a consensus, the quantized sensor states have been precisely modeled as a finite state Markov chain rather than assuming that quantization is equivalent to an additive analog noise.

¹ $\lim_{t \rightarrow \infty} \mathbb{E} \left\{ 1/N \sum_{i=1}^N \left[x_i(t) - 1/N \sum_{i=1}^N x_i(t) \right] \right\} = 0.$

Kashyap *et. al.* have examined the effects of quantization in consensus algorithms from a different point of view [53]. They require that the network average, ($\bar{x} = 1/N \sum_{i=1}^N x_i(t)$), be preserved at every iteration. To do this using quantized transmissions, nodes must carefully account for round-off errors. Suppose we have a network of N nodes and let Δ denote the “quantization resolution” or distance between to quantization lattice points. If \bar{x} is not a multiple of $N\Delta$, then it is not possible for the network to reach a strict consensus (*i.e.*, $\lim_{t \rightarrow \infty} \max_{i,j} |x_i(t) - x_j(t)| = 0$) while also preserving the network average, \bar{x} , since nodes only ever exchange units of Δ . Instead, Kashyap *et. al* define the notion of a “quantized consensus” to be such that all $x_i(t)$ take on one of two neighboring quantization values while preserving the network average; *i.e.*, $x_i(t) \in \{l\Delta, (l+1)\Delta\}$ for all i and some t , and $\sum_i x_i(t) = N\bar{x}$. They have shown that, under reasonable conditions, their algorithm will converge to a quantized consensus. However, the quantized consensus is clearly not a strict consensus, *i.e.*, all nodes do not have the same value.

Carli *et.al.* have studied quantized consensus algorithm from control theory perspective. In particular, they have mapped quantized average consensus into stability under quantized feedback problem [24]. They have proposed two schemes, *i.e.*, *zoom-in/zoom-out* and *logarithmic* quantizers, whose properties have been analyzed rigorously. They have provided the conditions under which the system will converge to a consensus.

Our key observation is that both temporal and spatial correlations among the node states increase as the system progresses through time. We are proposing two communication schemes to exploit the increasing correlation to drive the distributed quantized averaging algorithms to a consensus: (1) Predictive

coding for temporal correlation and (2) Wyner-Ziv coding for spatial correlation. Approximating the error as additive quantization noise, in Chapter 2, we show that under these schemes, a consensus in the mean square sense (*i.e.*, in \mathcal{L}^2) can be achieved where such consensus is defined as:

$$\lim_{t \rightarrow \infty} \mathbb{E} \left\{ \left(\frac{1}{N} \sum_{i=1}^N \left[x_i(t) - \frac{1}{N} \sum_{i=1}^N x_i(t) \right] \right)^2 \right\} = 0. \quad (1.2)$$

We discuss that one can still achieve convergence in \mathcal{L}^2 by decreasing quantization rate incrementally. We also derive necessary and sufficient conditions on the quantization noise variances for bounded MSE convergence, where MSE is defined as:

$$\text{MSE} = \lim_{t \rightarrow \infty} \mathbb{E} \left\{ \left(\frac{1}{N} \sum_{i=1}^N x_i(t) - \frac{1}{N} \sum_{i=1}^N x_i(0) \right)^2 \right\}. \quad (1.3)$$

In Chapter 3, we focus on the predictive coder where nodes utilize only their previous state values. Similar to Chapter 2, we utilize a dithered quantizer and thus model the quantization effect as temporally and spatially uncorrelated additive analog noise which is also uncorrelated with the message. Since characterization of the MSE performance is a challenging problem for general graphs, we focus on a particular communication patterns, *i.e.*, regular graphs with symmetric connectivity. In this setting, we provide explicit expressions and scaling laws for the MSE. In particular, we show that MSE is inversely proportional with the network connectivity. In addition, we will show that MSE is a function of the network size and scales as $O(N^{-1})$ when other parameters are fixed. We further study the characteristics of the average consensus algorithm under *variable* rate quantization scheme. In a special case where the quantization rates are chosen such that quantization noise variance decreases like a geometric series, we determine the rate regions where asymptotic quantization rate approaches

zero while convergence in \mathcal{L}^2 is guaranteed. In addition, we show that the rate regions achieving zero asymptotic rate and \mathcal{L}^2 convergence with bounded MSE, also achieve finite-sum rates. Therefore, the sum of the quantization rates over the iterations, is indeed finite. We discuss that transmitting more bits in early iterations and decreasing them gradually results in a better performance than using a predictive encoder with a fixed rate. We note that our work in Chapters 2 – 3 has been published in [102, 99, 100, 97].

1.2.2 Broadcast Gossiping Algorithms

In Chapter 4, we step back and focus on *unquantized* consensus problem, and study different ways to improve convergence speed of the existing algorithms. In particular, we propose *broadcast gossiping* for consensus which utilizes wireless nature of the communications in sensor networks. In [20], the authors proposed a consensus algorithm where each update results in a pairwise average of their values, the operation preserves both the total sum, and hence also the mean, of the node values. It was shown that gossiping for consensus algorithm converges to a consensus if the graph is strongly connected on the average. Because the transmitting node must send a packet to the chosen neighbor and then wait for the neighbor's packet, this scheme is vulnerable to packet collisions and yields a communication complexity (measured by number of radio transmissions to drive the estimation error to within $\Theta(N^{-\alpha})$, for any $\alpha > 0$) on the order of $\Theta(N^2)$ over random geometric graphs [20]. The geographic gossip algorithm proposed in [34] combines gossip with geographic routing to improve the convergence rate of random gossiping. Similar to the standard gossip algorithm, a node randomly wakes up, chooses a node randomly in the *whole network*, rather

than in its *neighborhood*, and performs a pairwise averaging with this node. Geographic gossiping increases the diversity of every pairwise averaging operation. The authors show that the communication complexity is in the order of $O(N^{3/2}\sqrt{\log(N)})$, which is an improvement with respect to the standard gossiping algorithm. More recently, a variety of the algorithm that “averages around the way” has been shown to converge in $O(N \log N)$ transmissions [16].

In our algorithm, a node in the network wakes up uniformly at random according to the asynchronous time model and broadcasts its value. This value is successfully received by the nodes in the predefined radius of the broadcasting node, *i.e.*, connectivity radius. The nodes that have received the broadcasted value update their own state value and the remaining nodes sustain their value. It is shown here that by iterating this procedure, this type of gossiping algorithm is capable of achieving consensus over the network with probability one. We also show that the random consensus value is, in expectation, equal to the desired value, *i.e.*, the average of initial node measurements. Because the sum of the node state values is not preserved at each iteration, the broadcast gossiping algorithm converges to a value that is in the neighborhood of the desired average. we provide theoretical and simulation results on the mean square error and communication cost performance of the broadcast gossip algorithm. Moreover, we study the effect of the so called *mixing parameter* on the convergence rate and limiting mean square error through theoretical results and numerical experiments. In addition, we derive the optimal mixing parameter when approached from the convergence rate perspective. Although the convergence time of our algorithm is commensurate with the standard pairwise gossip algorithms, we present simulations showing that for more modest network sizes our algorithm converges to consensus faster than other algorithms based on pairwise averages

or routing. This work has been published in [13, 8, 9].

1.3 Generalized Network Computation via Gossiping

In Chapter 5, we focus on a special network computation problem, where a group of destination nodes is interested in a function that can be decomposed as a sum of functions of local variables stored by another set of source-nodes. We refer to this problem as the Computing Along Routes (CAR) problem. This particular problem has a wide range of applications including aggregation queries, distributed detection, content distribution, decentralized traffic monitoring, distributed control and coordination [107, 65, 76, 92].

We focus on the case where the set of source nodes and the set of destination nodes are *disjoint*, and hence our case does not include average consensus gossiping [92] as a special case. This particular assumption has strong practical appeal since, in many cases, the nodes which are responsible for collecting the data and the nodes that are designated to process these data are disjoint and geographically separated. For instance, in the problem of distributed detection, only the fusion centers are interested in the outcome of the sensor decisions. In the case of content distribution, the entity who is interested in the data does not necessarily have access to any part of it before the distribution occurs. In the case of the leader-follower coordination problem [49], several nodes are to follow a group of leaders, and the source and destination sets are clearly disjoint.

In fact, the problem considered in this chapter is closely related to the data aggregation and routing problems studied in the computer science literature, where spatially distributed data is to be collected by a fusion center, utilizing

data aggregation and in-network processing techniques [107, 37, 70, 25, 4]. The similarity is obvious if one observes that solutions to the so called *duplicate-sensitive* data aggregation problem (see e.g. [70]) can be generalized to find codes that solve our problem. Duplicate-sensitive data aggregation refers to the case where destinations seek a single copy of data from each source. In the case of a single source and multiple destinations, the duplicate-sensitive aggregation problem has been studied extensively, proposing energy efficient schemes based on spanning trees [64, 96], and algorithms that are robust to link/node failures [70]. In the case of multiple source and multiple destinations, solutions have been proposed including hierarchical structures and minimizing routes costs [4, 45, 25]. These methodologies are not viable for solving the CAR problem, because one would need to explicitly consider duplicating the data of all sources at the destinations. Moreover, it falls short of explaining the relationship between type of topologies and the feasible queries and it does not incorporate feedback.

One solution to the problem we posed is to unicast between each pair of source-destination nodes via the shortest path joining source and destination [32]. After receiving all the information, each destination node can compute the desired function independently. Except in very special cases, this strategy is inefficient since the exact same information flows in the network several times, and it is unreliable since it is severely affected by link failures. The second approach is multicasting the information, having a single source node transmitting at a time to all destinations, thereby allowing the computation of the sought results independently [29]. Thanks to network coding [2],[35],[57], multicasting can be done using the links efficiently. However, this decomposition of the problem is agnostic to the fact that the nodes do not want the data themselves, but

an aggregate result.

We note that a setup resembling more to our problem is considered in [55], where the authors have applied the gossiping algorithm to solve a sensor localization problem, assuming that each node in the network wants to compute a linear combination of the anchor (source) nodes, to determine their exact locations. Unfortunately, this analysis can not be utilized to solve our problem since all of the non-source nodes are destinations, and destination nodes are not necessarily interested in the same function of the source nodes. A significant work is also due to Mosk-Aoyama et.al. who have considered distributed computation of separable functions as well as information dissemination on arbitrary networks [68, 69]. Moreover, Benezit et.al. have studied average consensus problem via randomized path averaging to achieve increased convergence rates [17]. However, our problem differs from these models in the sense that source and destination sets are disjoint, and there may exist intermediate nodes which are neither sources nor destinations.

In Chapter 5, we are proposing a gossiping based algorithm for jointly routing and calculating the desired function at the destinations. The innovative aspect of this procedure is that it incorporates feedback, unlike similar problems that consider a unidirectional flow of data. On the other hand, it does not necessarily distribute the desired value to the whole network which makes our scheme more secure and flexible. we introduce necessary and sufficient conditions for the existence of solutions to the proposed problem. By focusing on non-negative update weights, we investigate spectral properties of feasible codes and show what classes of code structures leads to feasible solutions. Moreover, we introduce reductions of the network topology which can be employed to

simplify the design problem without loss of generality. By focusing on stochastic codes, we provide necessary conditions on the topology for the feasibility and discuss some infeasible cases. We introduce a formulation for so called *partially directed* solution in terms of multicommodity flow problem. We compare the performance of our solution with the existing solutions. This work has been published in [103, 101, 104].

1.4 Opinion Diffusion in Social Networks via Gossiping

In Chapter 6, we study opinion diffusion in social networks and propose a gossiping based model to capture the diffusion. The seminal work on this particular area is due to DeGroot [30] where opinions of individuals are modeled as probabilities that might be thought of the probability that a given statement is true. The interaction patterns are captured through near neighbor based linear updates. We note that the updates of the DeGroot model has the exact the same as the synchronous average consensus updates. Finally, in their recent work, Acemoglu *et.al.* have studied an extension of the DeGroot model where updates are asynchronous and certain individuals are spreading misinformation by not updating their own beliefs [1]. In Chapter 6, we propose a gossiping based model where individuals randomly meet with their neighbors and probabilistically update their opinions. However, unlike the Degroot and misinformation models, we assume that individuals' opinions are discrete rather than continuous variables and there exists so called *stubborn* individuals who do not change their decisions. Therefore, unlike the models in [30, 1], individuals' opinion do not converge to a particular number, on the contrary, the opinions keep changing. This, in return, will help us to capture correlations among the opinions of

different individuals as well as the variations in the collective opinion of the society. We note that unlike Bayesian learning and observation models [46], we do not assume that there is a true parameter which the society is interested in capturing. We are interested in the propagation of opinions on a certain subject.

CHAPTER 2
CODING WITH SIDE INFORMATION FOR QUANTIZED AVERAGE
CONSENSUS

2.1 Motivation and Related Work

In this chapter, we are exclusively concerned with the symmetric information diffusion in complex networks, *i.e.*, agreement protocols. In particular, our focus will be on *discrete time average consensus* problem. As we have discussed in Chapter 1, discrete time average consensus algorithm is an iterative protocol which is based on local interactions only. The algorithm is quite powerful in the sense that under mild connectivity conditions, and perfect communications among local agents, any network will converge to consensus¹ under this particular protocol [92]. Moreover, the consensus point is not just any random value, on the contrary, it is the average value of the initial node states [92]. Unfortunately, these results hold under the crucial and non-practical assumption that nodes can store and transmit continuous values. While one can assume that nodes can store *almost* continuous quantities thanks to moderate storage abilities, transmitting even a single value from an uncountable set would take infinite amount of time [26]. Therefore, we will focus on the case where nodes have to quantize their values before transmitting them to their local neighbors. Specifically, we focus on the attainable mean square convergence performance under source encoding rate constraints at each node.

On the one hand, if the quantization noise is modeled as an independent ad-

¹We use the notion of *consensus* in the sense each and every node in the network will hold the same value.

ditive noise with *fixed variance*, it is concluded that consensus is not achievable and the asymptotic mean squared error is unbounded [93]. On the other hand, recent work on average consensus by Aysal *et.al* suggests that convergence can be attained *almost surely* under a probabilistic quantization scheme [10]. To prove it, the quantized sensor states are more precisely modeled as a finite state Markov chain rather than assuming that quantization is equivalent to an additive analog noise.

Kashyap *et.al.* have examined the effects of quantization in consensus algorithms from a different point of view [53]. They require that the network average to be preserved at every iteration. To do this using quantized transmissions, nodes must carefully account for round-off errors at each iteration. Kashyap *et.al.* define the notion of a “quantized consensus” to be such that all the node values take on one of two neighboring quantization values while preserving the network average. They show that, under reasonable conditions, their algorithm will converge to a quantized consensus. However, the quantized consensus is clearly not a strict consensus, *i.e.*, all nodes do not have the same value.

Our work is also closely related to [51]. In this particular study, authors have focused on average consensus algorithm under dithered quantization and random link failures. By utilizing stochastic approximation theory, they have showed that the proposed algorithm converges to a consensus with probability one. They have derived the analytical relationships between the design parameters (link weight sequence, quantizer bin width, the number of quantization levels) and the convergence speed of the algorithm as well as the mean squared error.

Finally, one can refer to [73] for detailed comparison of several quantized

consensus algorithms.

2.1.1 Summary of Main Contributions

In this chapter, we employ the additive noise model for the quantization error using a variable rate quantizer. Furthermore, we observe that *consensus algorithms* offer the perfect example of network communication problems where there is correlation between the data exchanged, and that the correlation increases as the system updates its computations. We would like to emphasize that while quantized consensus has been analyzed by numerous researchers as we have discussed in the previous section, the increasing correlation in the network has never been observed and analyzed before. Moreover, the connection between quantized consensus and information theoretic notions such as *quantization rate*, *source coding* is unique.

Considering coding strategies using side information we derive necessary and sufficient conditions on the quantization noise variances for bounded mean squared convergence and prove that such a convergence is possible even under zero asymptotic quantization rates. Our main contribution is summarized in the following lemma:

Theorem 1 *There exists rate allocation schemes for which the nodes converge to a consensus with a bounded mean squared error with respect to the true mean along with the quantization rate per message converges to zero. To achieve the vanishing per message rate one needs to employ side information, at either encoder and/or decoder in the form of the predictive coding and Wyner-Ziv coding schemes.*

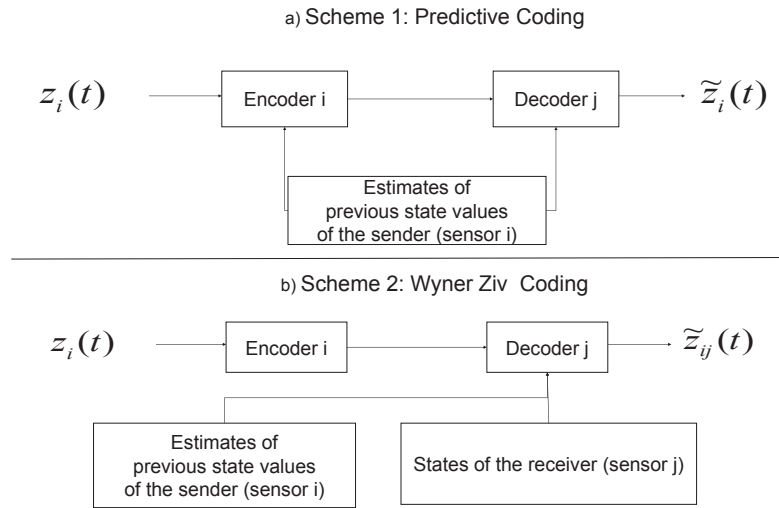
The chapter content can be, in a nutshell, summarized as the proof of the theorem above. Our results consider both the asymptotic limit of vector quantization as well as the practical case of scalar quantizers. We numerically analyze scalar quantizers based on predictive and nested lattice Wyner-Ziv encoding schemes, and validate the results of our theoretical findings with solutions that bear moderate encoding cost.

2.1.2 Chapter Organization

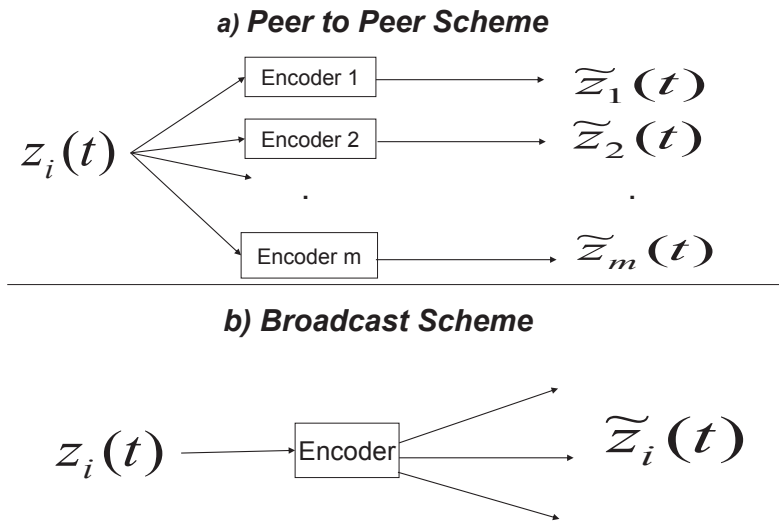
The remainder of this chapter is organized as follows: In Section 2.2, we review the main mathematical relationships characterizing average consensus algorithms. Section 2.3 explores the conditions on the noise variance under which the system converges to a consensus. We propose a predictive coding scheme which has the structure shown in Fig.2.1.2a in Section 2.4. In Section 2.5, we discuss Wyner-Ziv encoder/decoder scheme as in Fig.2.1.2b. Numerical examples evaluating the performances of the proposed algorithm and validating the theoretical findings are presented in Section 2.6. Finally, we present the end of the chapter discussions in Section 2.7.

2.2 Quantized Average Consensus Model

Denote by $z_i(t)$ the unquantized message of node i at iteration t , and by $\tilde{z}_i(t)$ its quantized value. We consider two main strategies which are represented in Fig. 2.1.2. For each of these two strategies, we consider the two possible communication scenarios in Fig. 2.1.2 called the *peer to peer* and the *broadcast* cases,



(a) Encoding/Decoding strategies for Consensus Problem.



(b) Peer to Peer vs Broadcast.

Figure 2.1: Charts for different quantization schemes.

respectively. As seen in Fig. 2.1.2 a) the peer to peer schemes utilize a different encoder for each particular destination, while the broadcast schemes in Fig. 2.1.2 b) require only one encoder for all receivers of a given sender. The reason of this distinction is that the *peer to peer* methods generally outperform the *broadcast* methods, but at the price of a more complex encoder structure and forcing to send a different message to each neighbor. While sending a different message over each link is acceptable in a wired network, it is wasteful in a wireless medium, where communications are naturally broadcast and each transmission reaches all neighbors. The *peer to peer* methods are proposed for *wired networks* and the *broadcast* methods are proposed for *wireless networks*, though our main focus will be broadcast methods.

We consider an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with the set of vertices \mathcal{V} and the set of edges \mathcal{E} . We assume that there are N nodes in the network, *i.e.*, $|\mathcal{V}| = N$ and N is finite. The set of edges \mathcal{E} consists of doubles $(i, j), i, j \in \mathcal{V}$, and if $(i, j) \in \mathcal{E}$, it means that nodes i and j are neighbors and they can communicate with each other. We note that since the network of interest is assumed to be undirected, $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$. We define the adjacency matrix of the graph \mathcal{G} as A [67]:

$$[A]_{ij} = \begin{cases} a_{ij} > 0, & \text{if there is an edge between node } i \text{ and } j \\ a_{ij} = 0, & \text{if } i = j \text{ or } i \neq j \text{ and there is no edge} \end{cases}. \quad (2.1)$$

Laplacian matrix associated with graph \mathcal{G} is defined as:

$$L = D - A, \quad (2.2)$$

where $D = \text{diag}(A\mathbf{1})$ is the degree matrix, $\text{diag}(\cdot)$ is the diagonal matrix of its arguments and $\mathbf{1}$ is all ones column vector. In this study, we consider the dis-

tributed average consensus algorithm which follows the update rule:

$$x_i(t+1) = x_i(t) - \sum_{j=1, j \neq i}^N \epsilon l_{ij} (x_j(t) - x_i(t)) \quad (2.3)$$

for $i \in \mathcal{V}$ and $t = 0, 1, \dots$. We can rewrite (2.3) in vector form as:

$$x(t+1) = (I - \epsilon L)x(t) = Wx(t) \quad (2.4)$$

where $x(t) = [x_1(t) \ x_2(t) \ \dots \ x_N(t)]^T$ and $[L]_{ij} = l_{ij}$. It was shown by Xiao and Boyd that above system converges to the average of any initial vector $x(0) \in \mathbb{R}^N$ if and only if W is balanced¹, and $\|W - \frac{1}{N}\mathbf{1}\mathbf{1}^T\| < 1$ where the norm is maximum singular value norm [92]. In other words, convergence is satisfied if 1 is an eigenvalue of W and it is also the only eigenvalue with the greatest magnitude. Similar results for continuous time iterations are in [82]. For $A = A^T$, by constraining $0 < \epsilon < 1/\max(A\mathbf{1})$, we guarantee that (2.4) asymptotically converges to average as in [92].

If we decompose (2.3) as follows:

$$x(t+1) = (I - \epsilon D)x(t) + \epsilon Ax(t) \quad (2.5)$$

we see that there are two different parts in the update rule: computing $(I - \epsilon D)x(t)$ requires only local values and $\epsilon Ax(t)$ uses the neighbors' values. We can therefore define:

$$z(t) = \epsilon x(t) \quad (2.6)$$

as the vector of variables which needs to be exchanged over the links available on \mathcal{G} at each iteration t . The entries of the vector $z(t)$, quantized with finite precision, are received by the neighbor as $\tilde{z}(t) = z(t) + w(t)$, where $w(t)$ is the quantization error.

¹ $\mathbf{1}^T W = (W\mathbf{1})^T = \mathbf{1}^T$

While there exists a substantial body of work on average consensus protocols under infinite precision and noiseless peer to peer communications, little research has been done introducing distortions in the message exchange, such as the noisy update assumption made in [93]. Specifically, Xiao and Boyd consider the following extension of (2.3):

$$x_i(t+1) = x_i(t) - \epsilon \sum_{j=1, j \neq i}^N l_{ij} (x_j(t) - x_i(t)) + w_i(t) \quad (2.7)$$

where $w_i(t)$, $i \in \mathcal{V}, t \geq 0$ are independent zero mean *fixed* variance identically distributed random variables. The authors show that under these assumptions, the system converges to the initial average only in mean (*i.e.* we do not have mean squared convergence) and that the mean squared error (MSE) deviation from the actual average, increases beyond a certain iteration. The authors also show that the node values do not converge to a common value as the number of iterations increases.

We propose, under a detailed internode communication model, to characterize $w_i(t)$ as an additive quantization noise in (2.7). By utilizing the increasing correlation among the node state values, we show that the variance of the quantization noise diminishes even with zero asymptotic rate. Furthermore, we show that the node values converge to a consensus as $t \rightarrow \infty$. We derive an expression for the mean squared deviation of the consensus value for both practical scalar quantizer and infinite length vector quantizer schemes. We also show that the mean squared deviation is bounded even under vanishing quantization rate regimes.

In the rest of the chapter, we will be using the following assumptions:

- (A1) The entries of $x(0)$ are random variables with zero mean² and finite variance (not necessarily independent).
- (A2) The nodes are strongly connected³, and W satisfies convergence conditions satisfied by unquantized average consensus protocol [92].
- (A3) The quantization noise samples at each step and sensor are uncorrelated with the messages and are also spatially and temporally uncorrelated, zero mean random variables⁴.
- (A4) Knowledge of the initial node statistics and topology (W matrix) is available at each node.

Remark 1 *With regard to (A3), in [86] Synder proved that a sufficient condition for quantization noise to be modeled as uniformly distributed and uncorrelated with the input message in the uniform scalar quantizer model is that characteristic function (CF) of the input message is bandlimited where $\frac{2\pi}{\Delta}$ is the upper bound on the bandwidth and Δ is the quantization bin width. In practice, characteristic functions are not exactly band-limited and the quantization theorems apply only approximately. However, as our adaptive consensus algorithm iterates Δ converges to 0 as discussed in Sections 2.4 and 2.5. Therefore above assumption will hold closely for larger iterations. Even in the initial iterations, under sufficiently large quantization rates such an assumption closely reflects the actual system behavior. In [86], Synder also showed that similar rules apply for spatial and temporal uncorrelation, i.e. the joint CF has to be band-limited.*

In Section 2.6, we numerically show that the theoretical behavior of the system matches with the simulated behavior and conclude that such assumptions hold closely

²In the case of unknown mean, due to the fact that the predictive coding scheme transmits the difference between the current state and its Linear Minimum Mean Squared Estimate, the mean of the difference can be approximated as 0. Hence, the performance is expected to be similar.

³A graph is strongly connected if there is a path from each vertex to each of the others.

⁴ $\mathbb{E}[w_i(t)w_j(l)] = 0$ unless $t = l$ and $i = j$.

even under lower rate regimes. In the case of infinite length vector coding, Zamir et.al. showed in [105] that the quantization noise approaches to a white Gaussian process.

Remark 2 *The knowledge of the topology mentioned in (A4) is not practical in a decentralized setting. While this assumption is used to keep track of the network statistics at each node, any predictive strategy or coding with side information strategy which does not make explicit use of the statistics can still be analyzed with our methods. Moreover, in [99], we have showed that as the node density increases or in homogeneously distributed networks, the encoder-decoder coefficients become independent on the network size and specific location of a node. Therefore, (A4) can be relaxed even for relatively small network sizes (i.e.. $N = 64$ as discussed in [99]).*

2.3 Convergence Conditions With Additive Quantization Noise

In this section, we utilize assumptions (A1)-(A3) and derive necessary and sufficient conditions on the quantization noise variances at each iteration and sensor, so that the nodes converge to a common value. We note that our convergence definition is in the mean squared sense, *i.e.* the nodes converge to a consensus if

$$\lim_{t \rightarrow \infty} \sum_{i=1}^N \mathbb{E}[\{x_i(t) - (N)^{-1} \mathbf{1}^T x(t)\}^2] = 0. \quad (2.8)$$

Then, we give additional constraints that lead to a consensus where the final value is bounded from the initial mean in the mean squared sense. Our approach is similar to that in [93]. We will focus on the behavior of the transmitted random vector $z(t) = \epsilon x(t)$ rather than $x(t)$ for brevity. The noisy recursion has

the following simple form:

$$x(t) = (I - \epsilon D)x(t-1) + A\tilde{z}(t-1) \quad (2.9)$$

where $\tilde{z}(t-1) = z(t-1) + w(t-1)$ as discussed in Section 2.2. Therefore:

$$\begin{aligned} z(t) &= \epsilon x(t) = (I - \epsilon D)z(t-1) + \epsilon A\tilde{z}(t-1) \\ &= (I - \epsilon D)z(t-1) + \epsilon Az(t-1) + \epsilon Aw(t-1) \\ &= Wz(t-1) + \epsilon Aw(t-1). \end{aligned} \quad (2.10)$$

In the rest of the chapter we will be using the system model in (2.9) and (2.10). The noise vector ($w(t)$) is assumed to be spatially and temporally uncorrelated as discussed in (A3). The following lemma is in order:

Lemma 1 *The nodes converge to a consensus in mean squared sense, if and only if the noise variance at each sensor converges to 0, i.e., $\mathbb{E}\{w_i^2(t)\} \rightarrow 0$ as $t \rightarrow \infty \forall i \in \mathcal{V}$.*

Proof Proof is given in Appendix A.1.

In the rest of the chapter, we will denote convergence in mean squared sense as \mathcal{L}^2 convergence. We note that speed of the convergence, *i.e.*, how fast quantization noise $w_i(t)$ converges to zero, does not change the fact that the nodes will reach a consensus.

Corollary 1 *If the node values converge to a consensus in mean squared sense then, $\mathbb{E}\{z(t)z^T(t)\} \rightarrow \Sigma^*$ where Σ^* is in the form of $\alpha \mathbf{1}\mathbf{1}^T$.*

Proof We will prove the corollary by showing $[\Sigma^*]_{ii} = [\Sigma^*]_{ij} \forall \{i, j\}$. If the node values converge to a consensus in mean squared sense, then

$\lim_{t \rightarrow \infty} \mathbb{E}\{|z_i(t)|^2\} = \lim_{t \rightarrow \infty} \mathbb{E}\{|\frac{\mathbf{1}^T z(t)}{N}|^2\} \forall i \in \mathcal{V}$. Therefore, $\lim_{t \rightarrow \infty} \mathbb{E}\{|z_i(t)|^2\} = \lim_{t \rightarrow \infty} \mathbb{E}\{|z_j(t)|^2\} \forall \{i, j\}$ pairs. As $t \rightarrow \infty$:

$$\begin{aligned} \mathbb{E}\{|z_i(t) - z_j(t)|^2\} &= \mathbb{E}\{|z_i(t)|^2\} + \mathbb{E}\{|z_j(t)|^2\} - 2\mathbb{E}\{z_i(t)z_j(t)\} = 0 \\ \mathbb{E}\{|z_i(t)|^2\} + \mathbb{E}\{|z_j(t)|^2\} &= 2\mathbb{E}\{|z_i(t)|^2\} = 2\mathbb{E}\{z_i(t)z_j(t)\} \\ \mathbb{E}\{|z_i(t)|^2\} &= \mathbb{E}\{z_i(t)z_j(t)\}. \end{aligned}$$

Therefore, $[\Sigma^*]_{ii} = [\Sigma^*]_{ij} \forall \{i, j\}$.

We note that Corollary 1 simply shows that the node values will be perfectly correlated in the limit. Unfortunately without further constraints on the quantization noises, the nodes may agree on a value which is very far from the initial average. In fact, if the noise variances converge to zero slow enough, the consensus value may even be unbounded. For this reason, we derive the conditions for bounding the final value from the initial mean in the mean squared sense.

We denote the average of the state values at time t by $a(t)$:

$$a(t) = \frac{1}{N} \sum_{i=1}^N x_i(t) = \frac{1}{N} \mathbf{1}^T x(t). \quad (2.11)$$

We also denote the average of the exchanged vector $z(t)$ by $b(t)$ which has the simple relation of $b(t) = \epsilon a(t)$. In the rest of the section, we focus on $b(t)$ to streamline the derivations. Then,

$$\begin{aligned} b(t+1) &= \frac{1}{N} \mathbf{1}^T z(t+1) = \frac{1}{N} \mathbf{1}^T W z(t) + \frac{\epsilon}{N} \mathbf{1}^T A w(t) \\ &= b(t) + \frac{\epsilon}{N} \mathbf{1}^T A w(t) \end{aligned} \quad (2.12)$$

where (2.12) follows the fact that $\mathbf{1}^T$ is an eigenvector of W with corresponding eigenvalue 1.

Lemma 2 *In the quantized consensus, the mean of the states is preserved in expectation, i.e.. $\forall t \geq 0$:*

$$\mathbb{E}\{a(t) - a(0)\} = \mathbb{E}\left\{\frac{1}{N}\mathbf{1}^T x(t) - \frac{1}{N}\mathbf{1}^T x(0)\right\} = 0. \quad (2.13)$$

Proof The proof follows from the fact that the noise vector is an uncorrelated quantity with zero mean and initial states are assumed to be zero mean.

We are interested in the behavior of the expected mean squared distance between the asymptotic average and initial average. In particular from (2.12):

$$\begin{aligned} \mathbb{E}\{(b(t) - b(0))^2\} &= \left(\frac{\epsilon}{N}\right)^2 \mathbb{E}\left\{\left(\sum_{l=0}^{t-1} \mathbf{1}^T A w(l)\right)^2\right\} \\ &= \left(\frac{\epsilon}{N}\right)^2 \sum_{l=0}^{t-1} \mathbb{E}\{(\mathbf{1}^T A w(l))^2\} \end{aligned} \quad (2.14)$$

$$= \left(\frac{\epsilon}{N}\right)^2 \sum_{i=1}^N \left(\sum_{l=0}^{t-1} \mathbb{E}\{w_i^2(l)\} \left(\sum_{m=1}^N a_{mi}\right)^2\right) \quad (2.15)$$

where (2.14) is due to the fact that the noise is temporally uncorrelated, and (2.15) follows from the fact that the noise is spatially uncorrelated. We would like to explore the conditions under which the above sum is bounded as $t \rightarrow \infty$.

Lemma 3 *Given finite number of sensors ($N < \infty$) and bounded A matrix ($a_{li} < \infty; \forall l, i \in \mathcal{V}$), the mean squared deviation from the initial average is bounded if and only if $\lim_{t \rightarrow \infty} \sum_{l=0}^{t-1} \mathbb{E}\{w_i^2(l)\}$ converges $\forall i \in \mathcal{V}$. Therefore, a necessary and sufficient condition for $\mathbb{E}\{(b(t) - b(0))^2\}$ to be bounded is that the noise variances at each sensor form a convergent series.*

Proof Proof of the lemma is straightforward from (2.15) and omitted.

We would like to note that by bounding $\mathbb{E}\{(b(t) - b(0))^2\}$, we guarantee that $\mathbb{E}\{(a(t) - a(0))^2\}$ is also bounded since $\mathbb{E}\{(b(t) - b(0))^2\} = \epsilon^2 \mathbb{E}\{(a(t) - a(0))^2\}$.

We derived necessary and sufficient conditions on the noise variances for agreeing on a common value whose mean squared distance from the average of the initial states $a(0)$ is bounded. The implication of this is that if the quantization rates can be chosen such that noise variances at each sensor forms a convergent series one can guarantee that the nodes will converge to the same value and that the error with respect to the actual value will be bounded. For example, one way to achieve this convergence is to choose the communication rates such that the quantization noise variances decay exponentially. In fact, any convergent sequences such as p-series² with $p > 1$ and geometric series³ with $\alpha > 1$ would be sufficient. On the other hand, if one is to consider source coding without side information (*i.e.* just quantize and transmit the state values $z(t)$), one would need a non-zero quantization rate (in bits) as $t \rightarrow \infty$ for achieving consensus. In the next two sections, we will study two different coding schemes which utilize the side information at the encoder or decoder to decrease quantization rate demand for bounded convergence.

2.4 Predictive Coding

In this section, we study the predictive encoding/decoding algorithm in Fig.2.1.2 a) for the *broadcast* and *peer to peer* scenarios. The predictive coding method utilizes past quantized node values to decrease the uncertainty of the present value, thus decreasing the transmission rate. This method is suitable for

² $\sum_{t=1}^{\infty} \frac{1}{p^t}$
³ $\sum_{t=1}^{\infty} \frac{1}{t^\alpha}$

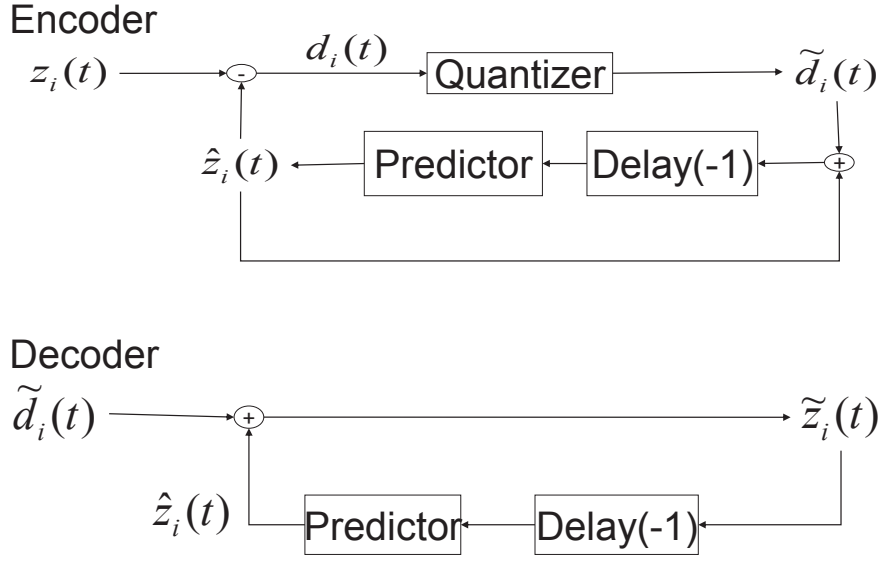


Figure 2.2: Differential encoder/decoder diagram.

the consensus problem since as the algorithm iterates, the past quantized values become more and more correlated with the current value, thus decreasing the prediction error. We explore both the *optimal* vector and scalar quantization schemes and, derive necessary analytical expressions.

2.4.1 Broadcast Predictive Coding

To exploit the local temporal correlation, the nodes can digitize $z_i(t)$ in (2.6) via the differential encoding/decoding scheme depicted in Fig.2.2. For each node i and time instant t , define:

$$\begin{aligned} \hat{z}_i(t) &= \text{prediction} \\ d_i(t) &= \text{prediction error} \\ \tilde{d}_i(t) &= \text{quantized prediction error} \end{aligned}$$

$$\begin{aligned}\tilde{z}_i(t) &= \text{noisy reconstruction} \\ w_i(t) &= \text{quantization error}\end{aligned}$$

The noisy reconstruction $\tilde{z}_i(t)$ is obtained through the following steps:

$$\hat{z}_i(t) = \sum_{l=1}^{p \leq t} a_i^{(t)}(l) \tilde{z}_i(t-l) \quad (2.16)$$

$$d_i(t) = z_i(t) - \hat{z}_i(t) = z_i(t) - \sum_{l=1}^{p \leq t} a_i^{(t)}(l) \tilde{z}_i(t-l) \quad (2.17)$$

$$\tilde{d}_i(t) = Q[d_i(t)] = Q[z_i(t) - \hat{z}_i(t)] = z_i(t) - \hat{z}_i(t) + w_i(t) \quad (2.18)$$

$$\tilde{z}_i(t) = \hat{z}_i(t) + \tilde{d}_i(t) = z_i(t) + w_i(t) \quad (2.19)$$

where in (2.16) $\hat{z}_i(t)$ is a linear minimum mean squared estimate (LMMSE) of $z_i(t)$ of order p ; $d_i(t)$ in (2.17) is the prediction error, to be quantized and transmitted; (2.18) is due to the fact that quantization error can be modeled as additive noise, and (2.19) is the reconstruction of $z_i(t)$ at the decoder. (2.19) shows us that $z_i(t)$ can be reconstructed at the receiver within some noise $w_i(t)$. We note that in predictive coding scheme prediction error is applied to the quantizer, and the output is transmitted to the decoder. Since linear predictor $\hat{z}_i(t)$ is also available at the decoder, $z_i(t)$ can be reconstructed. Interested readers may refer to [60] for more details on predictive coding.

Remark 3 *The analysis we will be making is actually valid with minor changes if the prediction error coefficients are fixed and the strategy (rate allocation and quantizer parameters) are fixed. For example, one could completely fix the strategy by using $a_1^{(t)} = 1$, $a_i^{(t)} = 0$ for $i > 1$ and strictly decreasing quantizer range, irrespective of the topology. In the following, we consider an optimized scheme using all of the information available.*

The optimum linear prediction coefficients in the mean squared sense are:

$$a_i^{(t)} = v_{z_i(t)}^T M_{\tilde{z}_i(t-1)}^{-1} \quad (2.20)$$

where, for $l, m = 1, \dots, p \leq t$:

$$[M_{\tilde{z}_i(t-1)}]_{lm} = \mathbb{E}\{\tilde{z}_i(t-l)\tilde{z}_i(t-m)\} \quad (2.21)$$

$$[v_{z_i(t)}]_m = \mathbb{E}\{z_i(t)\tilde{z}_i(t-m)\}. \quad (2.22)$$

Hence:

$$VAR[d_i(t)] = VAR[z_i(t)] - v_{z_i(t)}^T M_{\tilde{z}_i(t-1)}^{-1} v_{z_i(t)} \quad (2.23)$$

is the prediction error variance of a given node i and iteration t .

2.4.2 Analytical Framework

In this section, we will be utilizing (A1)-(A4). To be able to compute $[M_{\tilde{z}_i(t-1)}]$ and $[v_{z_i(t)}]$, we need to calculate $\mathbb{E}\{\tilde{z}_i(t-l)\tilde{z}_i(t-m)\}$ and $\mathbb{E}\{z_i(t)\tilde{z}_i(t-l)\}$ for $t, l \in \{1, \dots, p\}$. Either terms can be obtained taking the ii element of the cross-covariance matrices for a given t and l :

$$\begin{aligned} \mathbb{E}\{\tilde{z}_i(t-l)\tilde{z}_i(t-m)\} &= [\mathbb{E}\{\tilde{z}(t-l)\tilde{z}^T(t-m)\}]_{ii} \\ \mathbb{E}\{z_i(t)\tilde{z}_i(t-m)\} &= [\mathbb{E}\{z(t)\tilde{z}^T(t-m)\}]_{ii} \end{aligned}$$

which are easier to calculate because the recursions are more compact to express in terms of vectors.

Further details on the calculation of $\mathbb{E}\{\tilde{z}(t-l)\tilde{z}^T(t-m)\}$ and $\mathbb{E}\{z(t)\tilde{z}^T(t-m)\}$ are given in Appendix A.2 and A.3. We define state and noise vector covariances as follows:

$$\begin{aligned} \Sigma(t-m) &\triangleq \mathbb{E}\{z(t-m)z^T(t-m)\} \\ \Upsilon(t-m) &\triangleq \mathbb{E}\{w(t-m)w^T(t-m)\}. \end{aligned}$$

Table 2.1: Cross Correlations of Noisy States

	$\mathbb{E}\{\tilde{z}(t-l)\tilde{z}^T(t-m)\} =$
$l = m$	$\Sigma(t-m) + \Upsilon(t-m)$
$l < m$	$W^{m-l}\Sigma(t-m) + \epsilon W^{m-l-1}A\Upsilon(t-m)$
$l > m$	$\Sigma(t-l)(W^{l-m})^T + \Upsilon(t-l)\epsilon(W^{l-m-1}A)^T$

$\mathbb{E}\{\tilde{z}(t-l)\tilde{z}^T(t-m)\}$ is presented in Table 2.1 for a given t, m, l triplet in terms of the state and noise vector covariances, W , A , and ϵ . Similarly, $\mathbb{E}\{z(t)\tilde{z}^T(t-m)\}$ can be written in terms of these quantities as:

$$\mathbb{E}\{z(t)\tilde{z}^T(t-m)\} = W^m\Sigma(t-m) + \epsilon W^{m-1}A\Upsilon(t-m). \quad (2.24)$$

The values of $\Sigma(t-m)$, and $\Upsilon(t-m)$, that change with the index $t-m$, can be calculated in an iterative fashion. In fact, using (2.10), we can express state covariances in terms of *known* previous state covariances and noise covariances as follows:

$$\Sigma(t-m) = W\Sigma(t-m-1)W^T + \epsilon^2 A\Upsilon(t-m-1)A^T.$$

Last but not least, based on our Remark 1 and due to (A3), we model the covariance matrix due to quantization $\Upsilon(t-m)$ as a diagonal matrix where i th diagonal element represents the i th node's quantization noise variance. Given that *optimal* lattice quantizers are utilized at node i ([105]):

$$[\Upsilon(t-m)]_{ii} = \left(\frac{K}{K+2}\right) \text{VAR}[d_i(t-m)] 2^{-2R_i(t-m)} \quad (2.25)$$

where $R_i(t)$ is the quantization rate of node i at time $(t-m)$ and (K) is the dimension of the lattice quantizer. Throughout the symbol $\Delta_i(t)$ will denote the volume of the quantization tile used by node i at iteration t .

Remark 4 *In the case of infinite length vector coding (i.e. $K \rightarrow \infty$), $\frac{K}{K+2} \rightarrow 1$. In this scenario, we assume that each sensor observes a long data stream as opposed to a scalar state $x_i(0)$. While state vector at each sensor is encoded with K dimensional quantizer, each node runs K independent average consensus in parallel to update their states. In the case of scalar quantizer, $K = 1$ and variances are calculated accordingly.*

Remark 5 *Provided that the conditions necessary to apply (A3) hold, the performance of coding without side information can be analyzed simply setting $p = 0$.*

2.4.3 Algorithm Summary

In this subsection, we simply summarize our algorithm based on the predictive coder. At iteration $t \geq 0$, each node $i \in \mathcal{V}$:

1. obtains the linear predictor coefficients $a_i^{(t)}$, the prediction error variance $\text{VAR}[d_i(t)]$, and the quantization interval length $\Delta_i(t)$,
2. quantizes and transmits prediction error $d_i(t)$,
3. updates state and noise covariances matrices for the next iteration.

Receiving transmitted value, neighbor $j \in \mathcal{V}$:

1. obtains the linear predictor coefficients $a_i^{(t)}$,
2. reconstructs $z_i(t)$ as $\tilde{z}_i(t)$,
3. updates state and noise covariances matrices for the next iteration.

Once, transmissions are complete for all nodes, state values are updated by (2.9).

2.4.4 Asymptotic Rate Behavior for Predictive Coding

In this section, we show that the proposed scheme achieves a consensus with zero asymptotic rate per dimension in the case of vector coding. To prove, we will first introduce Lemma 4.

Lemma 4 *The system converges to a consensus if and only if $\sigma_{d_i}^2(t) \rightarrow 0$ as $t \rightarrow \infty$ where $d_i(t) = z_i(t) - \hat{z}_i(t)$.*

Proof Forward: If $\sigma_{d_i}^2(t) \rightarrow 0$, then the node values converge to a consensus.

Define $\sigma_{w_i(t)}^2 \triangleq \mathbb{E}\{w_i^2(t)\}$. If $\sigma_{d_i(t)}^2 \rightarrow 0$ then:

$$\sigma_{w_i(t)}^2 = 2^{-2R_i(t)} \sigma_{d_i(t)}^2 \rightarrow 0. \quad (2.26)$$

Reverse: If the nodes values converge to a consensus, then $\sigma_{d_i}^2(t) \rightarrow 0$.

Assume that the nodes converge. Then by Lemma 1, $\mathbb{E}\{w(t)w^T(t)\} \rightarrow 0$ and by Corollary 1, $\Sigma(t) \rightarrow \Sigma^* = \alpha \frac{\mathbf{1}\mathbf{1}^T}{N}$. By (2.24),

$$\begin{aligned} \mathbb{E}\{z(t)\tilde{z}^T(t-m)\} &= W^m \mathbb{E}\{z(t-m)z^T(t-m)\} \\ &+ \epsilon W^{m-1} A \mathbb{E}\{w(t-m)w^T(t-m)\}. \end{aligned}$$

Then, $\mathbb{E}\{z(t)\tilde{z}^T(t-m)\} \rightarrow W^m \Sigma^* + 0 = \Sigma^*$; $\forall m \in \{1, \dots, p\}$. Thus, $[v_{z_i(t)}]_m \rightarrow \Sigma_{ii}^*$.

By Table 2.1,

$$\mathbb{E}\{\tilde{z}(t-l)\tilde{z}^T(t-m)\} \rightarrow \Sigma^* + 0 = \Sigma^*. \quad (2.27)$$

Thus, $[M_{\tilde{z}_i(t-1)}]_{lm} \rightarrow \Sigma_{ii}^*$; $\forall l, m \in \{1, \dots, p\}$.

At step t , and sensor i , equation (2.20) can be rewritten as:

$$M_{\tilde{z}_i(t-1)} a_i^{(t)} = v_{z_i(t)}^T. \quad (2.28)$$

As $t \rightarrow \infty$, all entries of M matrix and v vector converges to Σ_{ii}^* as in Corollary 1. Therefore, (2.28) has infinitely many MMSE solutions. We pick $a_i^{(t)} = [1 \ 0 \ \dots \ 0]^T$. In other words, $d_i(t) = z_i(t) - \tilde{z}_i(t-1)$. Then,

$$\sigma_{d_i(t)}^2 \rightarrow \sigma_{z_i(t)}^2 - \frac{(\mathbb{E}\{z_i(t)\tilde{z}_i(t-1)\})^2}{\mathbb{E}\{\tilde{z}_i(t-1)\tilde{z}_i(t-1)\}} = \Sigma_{ii}^* - \frac{(\Sigma_{ii}^*)^2}{\Sigma_{ii}^*} = 0.$$

Lemma 5 *In the class of coding strategies which requires bounded rates $R_i(t) < \infty$ per iteration, predictive coding can use a vanishing rate per sensor, i.e. have $R_i(t) \rightarrow 0$ as $t \rightarrow \infty$, while guaranteeing converge to a consensus in mean squared sense.*

Proof By Lemma 1 and Lemma 4, the nodes converge to a common value if and only if $\sigma_{w_i(t)}^2$ and $\sigma_{d_i(t)}^2 \rightarrow 0$. By (2.26), $R_i(t)$ can be chosen as any non-negative number. Then, $R_i(t) \rightarrow 0 \ \forall i \in \mathcal{V}$.

By Lemma 5, we have showed that under the predictive coding scheme, a consensus can be achieved with zero asymptotic rate. While we do not have an analytical expression for the rate regions which satisfies the unbounded consensus, one strategy is given in the following subsection.

2.4.5 Feasible Rate Allocation for Bounded Convergence

We choose an arbitrary initial quantization rate $R_i(0) > 0 \ \forall i$. In the next iteration, i.e.. $t = 1$, quantization rates are chosen such that:

$$\frac{\sigma_{w_i(1)}^2}{\sigma_{w_i(0)}^2} = \frac{2^{-2R_i(1)}\sigma_{d_i(1)}^2}{2^{-2R_i(0)}\sigma_{d_i(0)}^2} = \left(\frac{t+1}{t+2}\right)^\beta = \left(\frac{1}{2}\right)^\beta \quad (2.29)$$

and $\beta > 1$. Then:

$$R_i(1) = 2^\beta(R_i(0) + \frac{1}{2} \log_2 \frac{\sigma_{d_i(1)}^2}{\sigma_{d_i(0)}^2}). \quad (2.30)$$

By recursion, at t th step, quantization rate is:

$$R_i(t+1) = \left(\frac{t+2}{t+1}\right)^\beta \left(R_i(t-1) + \frac{1}{2} \log_2 \frac{\sigma_{d_i(t)}^2}{\sigma_{d_i(t-1)}^2}\right). \quad (2.31)$$

Such constraint on the noise variances guarantees bounded convergence since quantization noises form a convergent p -series. If one did not use a coding scheme with no side information, under the same conditions quantization rate is:

$$R_i(t+1) = \left(\frac{t+2}{t+1}\right)^\beta \left(R_i(t-1) + \frac{1}{2} \log_2 \frac{\sigma_{z_i(t)}^2}{\sigma_{z_i(t-1)}^2}\right). \quad (2.32)$$

In Section 2.6 We numerically analyze quantization rate behavior for a specific β and initial quantization rates for both coding with side information and no side information.

2.4.6 Peer to Peer Predictive Coding

At a given time instant t , sensor i knows not only its own previously quantized values $(\tilde{z}_i(t-1), \tilde{z}_i(t-2), \dots)$, but also neighbors' previously quantized values, *i.e.* $\tilde{z}_j(t-1), \tilde{z}_j(t-2), \dots$ where j is a neighbor of i . If the link is bidirectional the prediction error at node i can be further reduced by having:

$$\hat{z}_i(t) = \sum_{l=1}^p a_i^{(t)}(l) \tilde{z}_i(t-l) + b_i^{(t)}(l) \tilde{z}_j(t-l) \quad (2.33)$$

where $a_i^{(t)}$ and $b_i^{(t)}$ are corresponding LMMSE coefficients. The derivations related to this method follow exactly the same logic steps in Section 2.4.1. For brevity, the detailed calculations are given in [98].

Remark 6 In [95], the authors have proposed a protocol to avoid average drift for quantized consensus problem where the message exchange is asymmetric, *i.e.* one node

sends its quantized state and the corresponding receiver sends the difference between the states. Such a scheme will be a special case of our peer to peer coding algorithm when $a_i^{(t)}(l) = 0 \forall i, t, l$ and $b_i^{(t)}(l) = 1$ for $l = 1$, 0 otherwise.

2.5 Wyner-Ziv Coding

In this section, we exploit the fact that a node can use its own sequence of present and past values locally, all or in part, as *side information* ([91],[85]) and therefore can improve the accuracy in the reconstruction of the neighbors' state in a way that is comparable to predictive coding. To this end, we note that the calculations done in Section 2.4.2 provide the covariance matrices of the current states and the cross covariance of current states with previous states of all nodes. We propose the use of a simple nested lattice code to utilize the side information in the scalar case. We analyze two schemes which are the *broadcast* and the *peer to peer* versions of the strategy.

2.5.1 Broadcast WZ Coding

Coding with side information or Wyner-Ziv (WZ) coding is the encoding strategy that leads to reduced rate or improved performance by relying on the fact that the decoder can make use of side information correlated with the incoming message. In the classical WZ scheme, there is a single decoder-encoder pair and the decoder has access to a corrupted version of the encoded data. Parallel to this scheme, in average consensus problem the decoder has access to the past values of the encoder's quantized states and its own current and past states.

Let us denote the neighbor set of sensor i as N_i . Then for each sensor $j \in N_i$, available side information is:

$$\hat{z}_{ji}(t) = \sum_{l=1}^p b_{ji}(l) \tilde{z}_{ji}(t-l) + \sum_{l=0}^p c_{ji}(l) z_j(t-l) \quad (2.34)$$

where $\hat{z}_{ji}(t)$ is the linear estimate of $z_i(t)$ at neighbor j and $b_{ji}(l)$, $c_{ji}(l)$ are the corresponding LMMSE coefficients. The reason why we utilize LMMSE as the side information is the following: WZ coding has been extensively studied for the case where the side information is a corrupted version of the message to be decoded and the noise is assumed to be additive and independent from the message ([91],[89]). Therefore by utilizing LMMSE of the transmit sensor's value as side information, we analyze the rate constrained average consensus problem with classical WZ coding scheme.

Note that, although there is only one encoder used for all neighbors, the reconstruction of the state value $\tilde{z}_{ji}(t-l)$ (and its quantization noise) is different at each receiver due to the heterogenous side information $\hat{z}_{ji}(t)$. To compute the side information, the decoder uses the supervector $\gamma_{ji}(t)$:

$$\gamma_{ji}(t) = [z_j(t) \ z_j(t-1) \ \dots \ z_j(t-p) \ \tilde{z}_{ji}(t-1) \ \dots \ \tilde{z}_{ji}(t-p)]^T \quad (2.35)$$

and the covariance matrix and cross-correlation vector:

$$M_{ji}(t) = \mathbb{E}\{\gamma_{ji}(t)\gamma_{ji}^T(t)\} \quad (2.36)$$

$$v_{ji}(t) = \mathbb{E}\{z_i(t)\gamma_{ji}^T(t)\}. \quad (2.37)$$

Then, the side information in (2.34) can be written in a compact form as:

$$\hat{z}_{ji}(t) = v_{ji}^T(t)M_{ji}^{-1}(t)\gamma_{ji}(t). \quad (2.38)$$

The reader should note that $M_{ji}(t)$ is a $(2p+1) \times (2p+1)$ matrix for a given order p . The upper left $(p+1) \times (p+1)$ block of the matrix contains cross correlations

of the states $t, \dots, t - p$ at sensor j and can be calculated by (A.13). The lower right $p \times p$ block is the covariance of the noisy reconstructions of sensor i , and is calculated via the recursion shown in Table 2.1. The upper right and lower left blocks of the matrix are cross correlations between reconstructions of sensor i and states of sensor j and are derived in (2.24). The necessary equations to derive $v_{ji}(t)$ vector are (2.24) and (2.25).

Once all the neighbors' information is received and decoded, the sensor i performs the following state update:

$$x_i(t) = (1 - \epsilon \sum_{j=1}^N a_{ij})x_i(t-1) + \sum_{j=1}^N a_{ij}\tilde{z}_{ij}(t-1). \quad (2.39)$$

To write the network equations we introduce $\tilde{Z}(t)$ as a $N \times N$ matrix whose ij entry is reconstruction of $z_j(t)$ by sensor i . The network equation is given as:

$$x(t) = (I - \epsilon D)x(t-1) + (A \odot \tilde{Z}(t-1))\mathbf{1} \quad (2.40)$$

where \odot represents entry by entry matrix multiplication and $\mathbf{1}$ is all ones column vector.

Once all sensors update their current states (at time t) as in (2.39), the new step requires knowing $M_{ji}(t+1)$, and $v_{ji}(t+1)$ defined in (2.36) and (2.37). As mentioned before, these quantities can be written in terms of the state and the noise covariances and calculated recursively as indicated in Section 3.1.1. The only difference here compared to Section 3.1.1 is that, because each term $\tilde{z}_{ij}(t-1)$ has its own specific quantization error $w_{ij}(t-1)$ associated, the update of the covariance matrix of the states has a different form compared to (2.25). By 2.6 and (2.40):

$$z(t) = Wz(t-1) + \beta(t-1) \quad (2.41)$$

where $[\beta(t-1)]_i = \epsilon \sum_{j=1}^N a_{ij} w_{ij}(t-1)$. Then:

$$\mathbb{E}\{z(t)z^T(t)\} = WE\{z(t-1)z^T(t-1)\}W^T + \mathbb{E}\{\beta(t-1)\beta^T(t-1)\}$$

where, approximating the $w_{ij}(t-1)$ as being spatially uncorrelated:

$$[\mathbb{E}\{\beta(t-1)\beta^T(t-1)\}]_{ii} = \epsilon^2 \sum_{j=1}^N a_{ij}^2 \text{VAR}[w_{ij}(t-1)]$$

and the non-diagonal entries are 0. In the case of infinite length vector coding Wyner has shown that if the source and the side information are jointly Gaussian with i.i.d entries, WZ quantization rate is equal to the rate which would be required if the encoder is informed of the side information as well as the decoder [90]. In this case, we can treat the problem as *predictive coding* where there is extra side information with respect to Section 2.4, *i.e.*, receiver's past and current values. In this case, the quantization noise is:

$$\begin{aligned} \text{VAR}[w_{ji}(t)] &= \mathbb{E}\{(z_i(t) - \hat{z}_{ji}(t))^2\} 2^{-2R_i(t)} \\ &= (\text{VAR}[z_i(t)] - v_{ji}^T(t)M_{ji}^{-1}(t)v_{ji}(t)) 2^{-2R_i(t)}. \end{aligned}$$

In the case of scalar coding, our implementation will be based on a *nested lattice code* construction (see e.g. [78] and [106]). We will use a simple encoding approach discussed in [63] and [89]. While we are not going to repeat the algorithm itself, we show how to choose *quantization step* parameter for the average consensus problem. Quantization step for sensor i is chosen such that average error variance per neighbor is minimized. Given quantization rate, the optimal quantization step is found by solving the following convex optimization problem:

$$\arg \min_{\Delta > 0} \sum_{j \in N_i} \Phi(R_i(t), \Delta, 2^{2R_i(t)} \text{VAR}[w_{ji}(t)]) \quad (2.42)$$

such that

$$\begin{aligned} \Phi(R, \Delta, 2^{2R_i(t)} \text{VAR}[w_{ji}(t)]) &= 2^{2R_i(t)} \Delta^2 \sum_{t=0}^{\infty} (2t+1) Q \left(\frac{(t + \frac{1}{2}) 2^{R_i(t)} \Delta}{2^{2R_i(t)} \text{VAR}[w_{ji}(t)]} \right) \\ &+ \frac{\Delta^2}{12}. \end{aligned}$$

Define $\Delta_i(t)$ as the solution of the above optimization problem. Then, node i 's state is reconstructed at the neighbor j with some noise where the noise variance is equal to $\text{VAR}[w_{ji}(t)] = \Phi(R, \Delta_i(t), 2^{2R_i(t)})$.

Remark 7 *In the peer to peer WZ coding scheme, each sensor utilizes a different encoder for each neighbor. In this case, each sensor (i) will solve $|N_i|$ optimization problems in parallel to obtain optimum quantization steps for $|N_i|$ neighbors.*

For WZ coding the results pertaining convergence that we state are limited to the Gaussian infinitely long vector quantizer case. In this case we add the following:

Lemma 6 *For Gaussian vectors of infinite length and for an identical connectivity matrix, quantization rates and initial correlation the WZ scheme outperforms the purely predictive coding scheme.*

Since for the Gaussian case WZ coding is asymptotically equivalent to a scheme that shares the same side information at both encoder and decoder [91], this means that asymptotically at each iteration the scheme is equivalent to scheme that encodes a prediction error with smaller variance $\text{VAR}[d_i(t)]$ compared to the scheme that utilizes exclusively past states to calculate the prediction error. Hence, greater precision is attained for the same rate, leading to more (or simply no less) accurate bounded convergence. In Fig.2.3, we simulate the infinite length vector coding case (by calculating noise variances iteratively under some

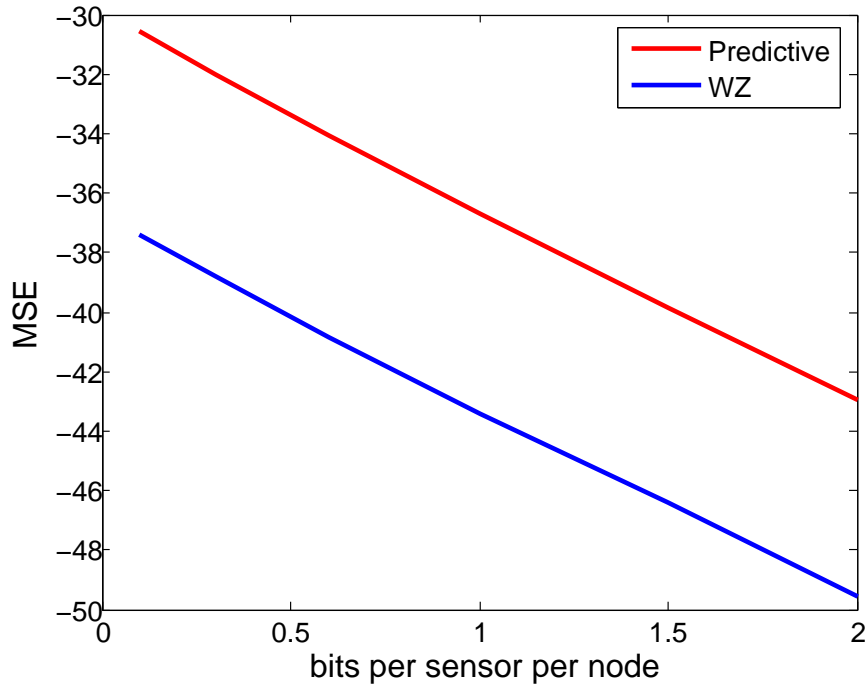


Figure 2.3: Theoretical performance comparison between WZ and Predictive coding, $N = 10, r = 0.4$

initial conditions) where at each iteration all the side information is used, *i.e.* $p = t - 1$. The x -axis represents quantization bits per sensor per dimension and y -axis represents MSE per dimension. As Lemma 6 suggests WZ outperforms the predictive coding scheme due to extra side information available at the decoder.

Combined with Lemma 5 a small corollary of this lemma, is that also WZ asymptotically will guarantee convergence with bounded MSE with respect to the true mean. Note that one can repeat the construction given in to determine one achievable rates that grants bounded convergence with vanishing asymptotic rate per sensor. For brevity we will omit its straightforward extension.

Unfortunately, WZ coding requires quite large vectors in order to achieve near to optimal performance and finite dimensional cases that we explore in

our numerical sections do not perform better than the predictive schemes, if one uses the same number of states (past or present) to encode.

2.6 Numerical Results

In this section, we show the numerical performance of the *broadcast* algorithms proposed, illustrating their convergence characteristics in the network scenario defined next. *Peer to peer* schemes are not included in the section for brevity; their performance are naturally better than those of the *broadcast* schemes. For simulation purposes, several random geometric graphs $\mathcal{G}(N; r)$ are generated with nodes uniformly distributed over a unit square. According to the definition of $\mathcal{G}(N; r)$, there exists a link between any two nodes if their range is less than r . We assume that there are no channel errors between two nodes that are connected. Moreover, we assume that matrix A in (2.1) is such that:

$$a_{ij} = \left\{ \begin{array}{l} a_{ij} = 1, \quad \text{if there is an edge between node } i \text{ and } j \\ a_{ij} = 0, \quad \text{if } i = j \text{ or } i \neq j \text{ and there is no edge} \end{array} \right\} \quad (2.43)$$

First, we investigate the behavior of the quantization rate and compare it among three different schemes: broadcast, Wyner-Ziv and coding without any side information. Our adaptive quantization scheme without any side information encodes the sensor state without removing any redundancy using previous values or side information at the decoder. It uses a uniform quantizer with a finite range. The range changes as a function of the state variance as discussed in the predictive coding scheme. Therefore, more quantization bits are needed to keep the state variances and thus the noise variances decreasing. Fig.2.4 shows the behavior of the average number of quantization bits per sensor over 40 iter-

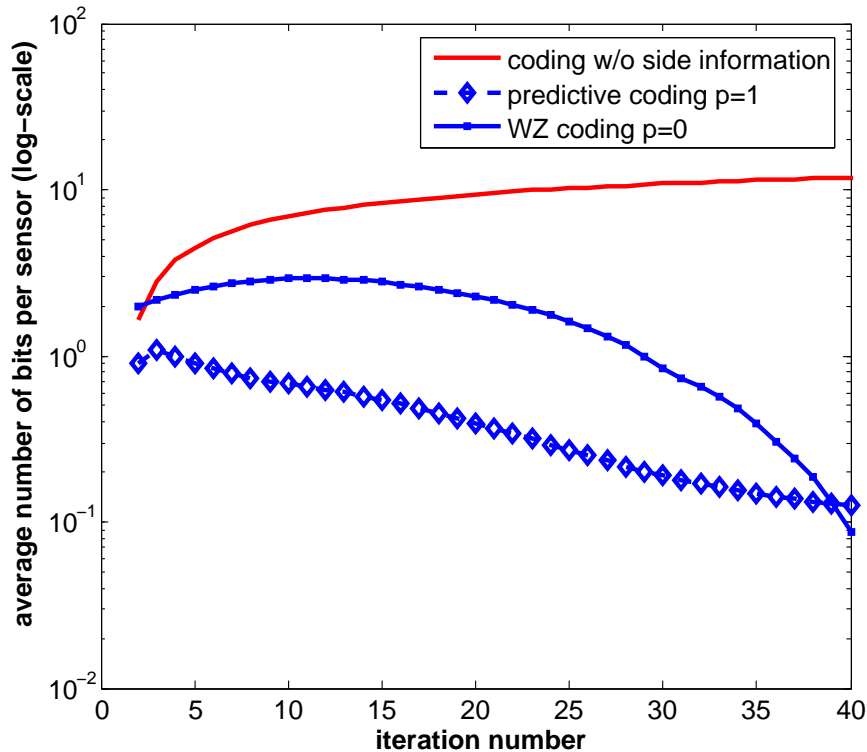


Figure 2.4: $r = 0.4$, $N = 50$, Total number of iterations = 65.

ations. The simulation parameters are $r = 0.4$, $N = 50$. The data at each sensor is initialized as zero mean unit variance Gaussian random variable. At each iteration the quantization rates are chosen such that $\frac{\sigma_{w_i}^2(t)}{\sigma_{w_i}^2(t-1)} = \left(\frac{t+1}{t+2}\right)^\beta$ and $\beta = 4$ to guarantee bounded MSE convergence as discussed in Section 2.5.1. Under these conditions, we observe that WZ coding requires higher quantization rates compared to predictive coding strategy in the initial steps. On the other hand, WZ quantization rates approach to 0 faster.

In Fig. 2.5, we plot the simulated performance (rate-distortion) and the theoretical performance of the predictive coding algorithm using a uniform scalar quantizer. The p value is chosen as 1 and c as 20. The results are averaged over 10 random networks with 64 nodes and 1000 independent Monte Carlo simulations. We observe that when the quantization rate is sufficiently high (5 bits per

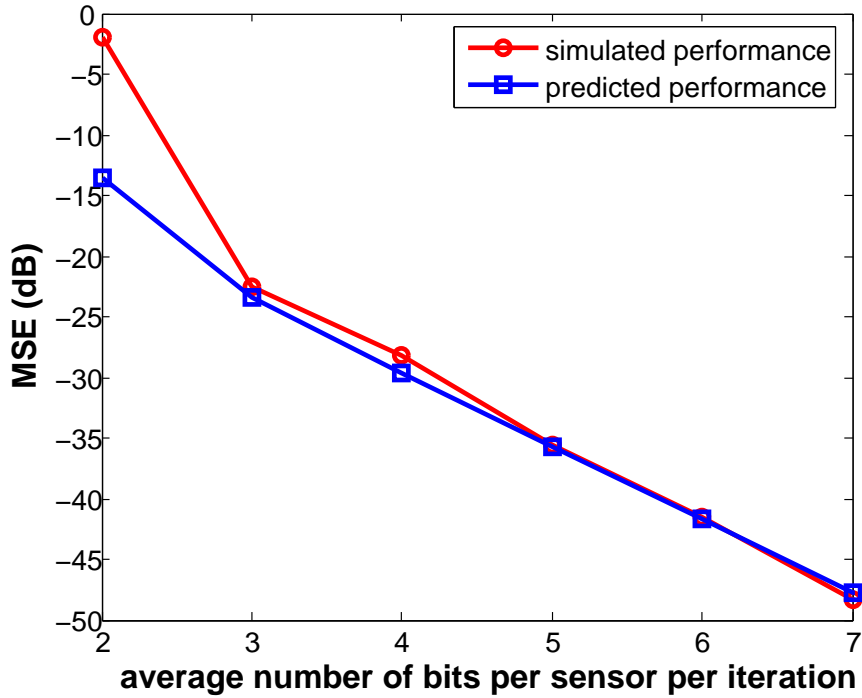


Figure 2.5: Simulated performance vs. theoretical performance.

sensor per iteration) then (2.15) approximates very accurately the performance of the predictive coding method. Unfortunately in the lower rate regimes, the actual performance is not close to the analytical one, due to the inaccuracy of assumption (a3) that we discussed in length in Section 2.2. Therefore, (2.15) will be underestimating the actual error performance under the lower rate regimes.

In Fig. 2.6, we numerically compare the MSE error performances of two schemes on a random graph with $N = 25$, $r = 0.5$ and fixed quantization rates per sensor per iteration. Empirical MSE performance is defined as:

$$MSE(t) = \left(\frac{1}{N} \sum_{i=1}^N z_i(t) - \frac{1}{N} \sum_{i=1}^N z_i(0) \right)^2.$$

All algorithms are initialized by the same zero mean unit variance Gaussian random data. The initial sensor observations are assumed to be uncorrelated. The results are averaged over 1000 Monte-Carlo simulations. We observe that both

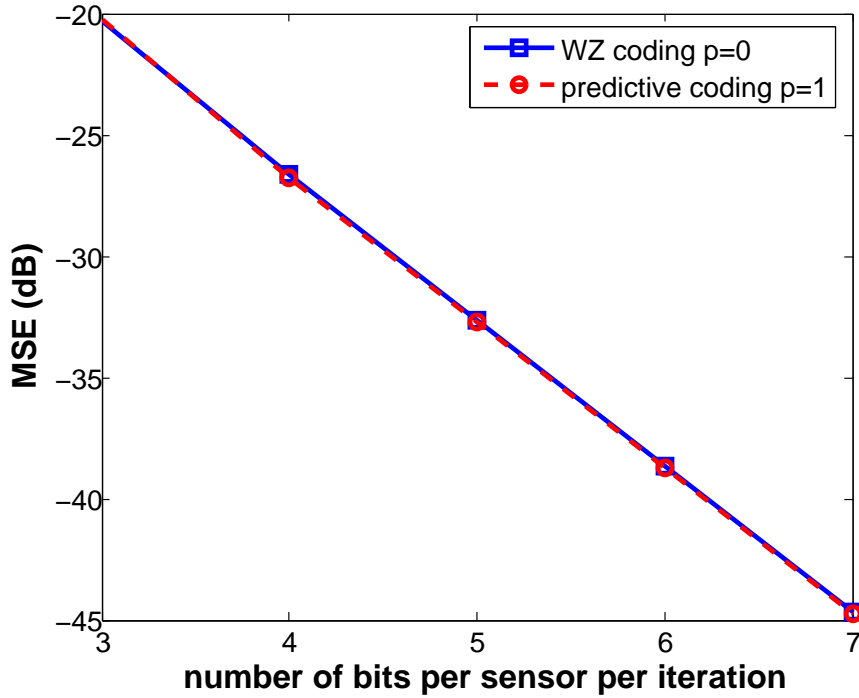


Figure 2.6: Rate distortion curves for WZ and predictive coding schemes over 50 iterations

schemes have similar MSE performances although the WZ scheme uses only the current state of the receiver as the side information, *i.e.* $p = 0$. The predictive coding scheme stores and utilizes only the previous state value, *i.e.* $p = 1$. Due to the brute force minimization of (2.42), if the quantization parameters are not pre-computed, the WZ scheme is more complex than the predictive coding scheme. WZ is advantageous because it does not require storing past quantized states.

In Fig. 2.7, we compare the MSE performance of the algorithm for different amounts of side information (*i.e.*, $p = 1, p = 5$, and $p = 10$) under the predictive coding scheme. We conclude from Fig. 2.6 that the increasing amount of the past quantized state information does not increase the MSE performance significantly. The behavior indicates that the correlation between the current state

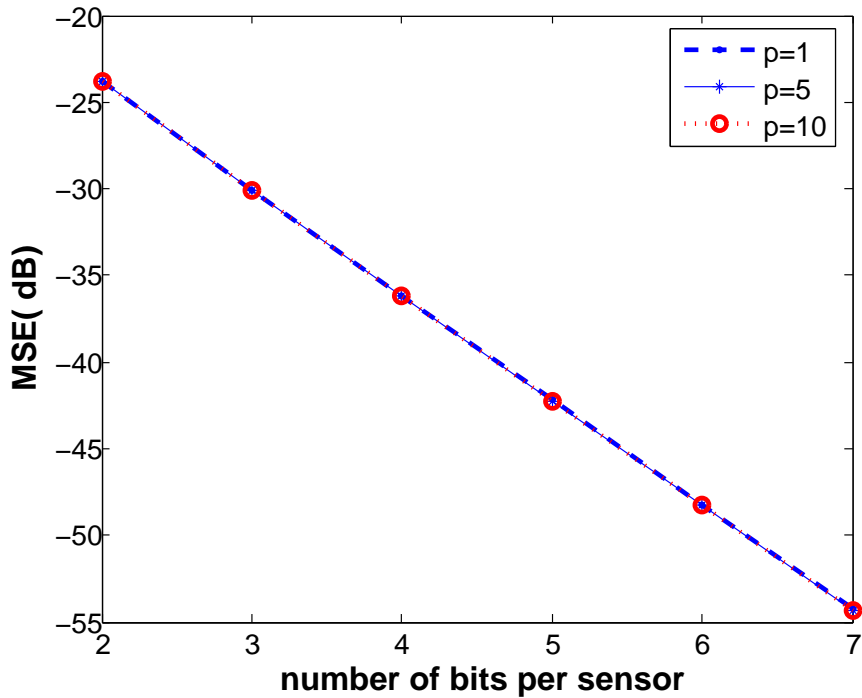


Figure 2.7: MSE performance of the predictive coding for different p values.

and the past states is captured almost entirely in the previous state only, and it is unnecessary to use predictors of order greater than 1. This is intuitively pleasing, since prediction error filter somewhat reverses the averaging process, which is an AR process of order 1.

2.7 Discussions

We have explored conditions on the quantization noise variances which are required for convergence of the nodes in the *average consensus* problem and proposed and investigated two source coding strategies which satisfy bounded convergence constraints with zero average rate asymptotically. We have given the mathematical framework for predictive coding, and nested lattice coding

for both *peer to peer* and *broadcast* communications scenarios, providing the details on how to implement each strategy. Our most significant contribution is showing that, using the temporal correlation and the ever increasing spatial correlation, bounded consensus can be achieved with zero asymptotic rate.

CHAPTER 3

MSE CHARACTERIZATION FOR QUANTIZED AVERAGE CONSENSUS FOR FIXED AND VARIABLE RATE QUANTIZERS

3.1 Motivation and Related Work

In this chapter, we will continue investigating coding strategies for quantized average consensus problem. While our results in Chapter 2 are significant in terms of exploring positive effects of side information coding for consensus type problems, we have not provided closed form expressions and scaling laws for the system characteristics such as MSE. Moreover, our main result in Chapter 2 was Theorem 1 which argues that under side information coding schemes, one can gradually decrease the quantization bit rate and still achieve a network wide consensus. However, we have not analyzed what good strategies are for choosing quantization rates over the iterations. In this chapter, by focusing on simpler coding strategies, we will seek answers to the problems mentioned above.

3.1.1 Summary of Contributions

In this chapter, we focus on the predictive coder where nodes utilize only their previous state values. Similar to Chapter 2, we utilize a dithered quantizer and thus model the quantization effect as temporally and spatially uncorrelated additive analog noise which is also uncorrelated with the message. Under these assumptions, we prove that our scheme converges to a consensus in \mathcal{L}^2 . Since

characterization of the MSE performance, where MSE is defined as:

$$\text{MSE} = \lim_{t \rightarrow \infty} \mathbb{E} \left\{ \left(\frac{1}{N} \sum_{i=1}^N x_i(t) - \frac{1}{N} \sum_{i=1}^N x_i(0) \right)^2 \right\}, \quad (3.1)$$

is a challenging problem for general graphs, we focus on a particular communication patterns, *i.e.*, regular graphs with symmetric connectivity. In this setting, we provide explicit expressions and scaling laws for the MSE. In particular, we show that MSE is inversely proportional with the network connectivity and initial observation correlation. In addition, we show that MSE is a function of the network size and scales as $O(N^{-1})$ when other parameters are fixed. We further study the characteristics of the average consensus algorithm under *variable* rate quantization scheme. In a special case where the quantization rates are chosen such that quantization noise variance decreases like a geometric series, we determine the rate regions where asymptotic quantization rate approaches zero while convergence in \mathcal{L}^2 is guaranteed. In addition, we show that the rate regions achieving zero asymptotic rate and \mathcal{L}^2 convergence with bounded MSE, also achieve finite-sum rates. Therefore, the sum of the quantization rates over the iterations, is indeed finite. We conclude that transmitting more bits in early iterations and decreasing them gradually results in a better performance than using a predictive encoder with a fixed rate.

In the rest of the chapter, the quantization operation is denoted as $Q[\cdot]$ and variables with a tilde mark are quantized quantities $\tilde{x} = Q[x]$. Throughout our analysis, we assume that each node encodes and transmits a long block (length K) of state variables where the block entries are i.i.d. random variables. i.i.d. assumption on the entries of the message blocks results in utilization of optimal lattice quantizers in \mathbb{R}^K . Nice properties regarding quantization noise for optimal lattice quantizers are discussed in Section 3.2. Therefore, the quantization

rate per iteration, indicated by $R(t)$, is per symbol rate where $2^{2KR(t)}$ is the total number of lattice cells in \mathbb{R}^K , in general changing with the iteration index (t) and vector quantizer dimension (K). While our notation considers one symbol at a time, the use of vector quantization allows us to consider rates per symbol which are not integers. Finally, Assumptions $A(1) - A(4)$ of Chapter 2 are still in order throughout this chapter.

3.1.2 Chapter Organization

In Section 3.2, we introduce a predictive coding scheme for quantized average consensus problem. Section 3.3 summarizes some properties of the regular networks, discussed convergence proof and characterizes MSE performance of the scheme using fixed rate quantization under the discussed regular networks. We derive asymptotic rate behavior and optimum rate allocation for the variable rate quantization scenario in Section 3.4. Finally, we conclude the chapter with Section 3.5.

3.2 Quantized Average Consensus

In the rest of the chapter, we assume that initial sensor observations are zero mean random variables with identical distributions (not necessarily independent). As we have described in Chapter 2 in detail, sensors will exchange their values with their neighbors at each iteration and update their state values as a linear combination of these quantities. We are interested in the case where sensors quantize their values before exchanging their state values.

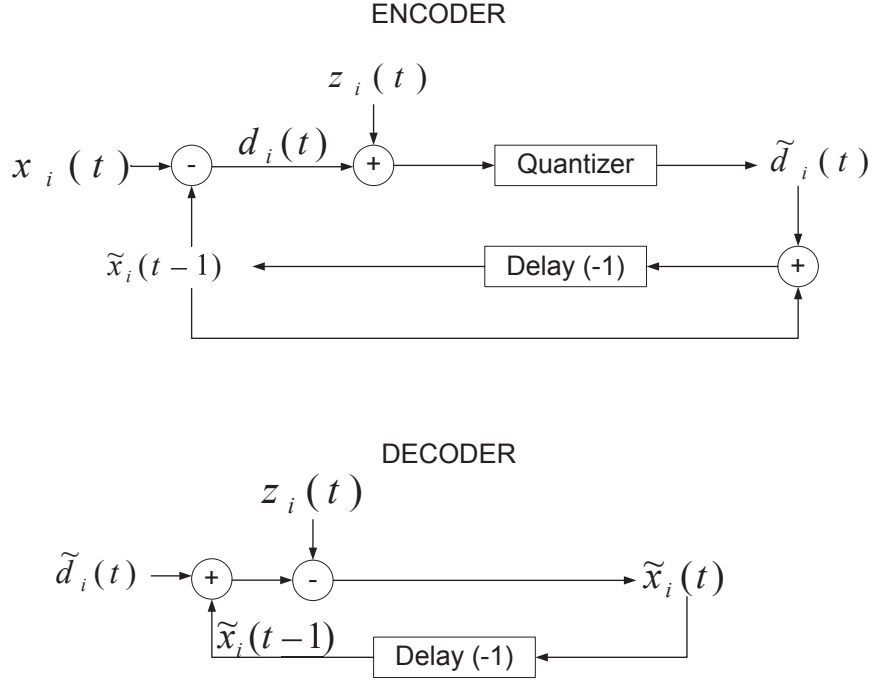


Figure 3.1: Differential encoder/decoder diagram with dithering.

Although, nodes encode a long block of data, state and noise values and their statistics mentioned in the chapter are per dimension quantities. As the source coding scheme, we will be using a simple first order differential encoder/decoder depicted in Fig. 3.1 to explore increasing temporal correlation. This coding scheme has been first introduced in Section 2.4. Moreover, we will utilize a dithered quantization to whiten the quantization noise. Our channel model is based on protocol model where there are no channel errors between neighboring nodes [42].

We denote $d_i(t)$ as the innovation to be transmitted from node i to its neighbor at time t , $z_i(t)$ as the dither to be added (which is uniformly distributed in K dimensional lattice and independent from the message $d_i(t)$), $\tilde{d}_i(t)$ as the quantized message, $\tilde{x}_i(t)$ as the noisy state reconstruction and $w_i(t)$ as the quan-

tization error. Then, the noisy state reconstruction is follows:

$$d_i(t) = x_i(t) - \tilde{x}_i(t-1) \quad (3.2)$$

$$\begin{aligned} \tilde{d}_i(t) &= Q[d_i(t) + z_i(t)] = Q[x_i(t) - \tilde{x}_i(t-1) + z_i(t)] \\ &= x_i(t) - \tilde{x}_i(t-1) + z_i(t) + w_i(t) \end{aligned} \quad (3.3)$$

$$\begin{aligned} \tilde{x}_i(t) &= \tilde{d}_i(t) + \tilde{x}_i(t-1) - z_i(t) \\ &= x_i(t) + w_i(t) \end{aligned} \quad (3.4)$$

where (3.2) is due to the fact that at each iteration the difference between the current state and the previous quantized state is transmitted, (3.3) follows from utilizing a uniform dither and modeling the quantization error as additive noise, and (3.4) follows since the previous quantized state value and dither are also known at the neighbors' decoders.

Once the prediction is transmitted and the noisy states are reconstructed at the nodes, network update is performed through:

$$x(t+1) = W\tilde{x}(t) = W(x(t) + w(t)) \quad (3.5)$$

where $w(t) = [w_1(t), w_2(t), \dots, w_N(t)]^T$ and W is a doubly stochastic matrix. In the rest of the chapter, we denote our algorithm defined in (3.2)-(3.5) as quantized consensus algorithm with predictive coding (QCPC). We note that (3.5) is slightly different than our formulation in (2.9). In Chapter 2, we have utilized the fact that each node i has access to the unquantized version of its own value $x_i(t)$, therefore it should utilize this particular value in the updates to further reduce the quantization error. In this chapter, for the sake of mathematical brevity, we focus on the case where nodes always use quantized state values in the updates. We note that $w(t)$ is assumed to be uncorrelated with the messages and is also spatially and temporally uncorrelated, zero mean random vector, uniformly distributed on the K dimensional lattice. While such

assumptions require strict conditions on the distribution of the message and quantization rates [86, 100], since we have utilized uniform subtractive dithering, this approximation is accurate for all quantization levels [105]. Moreover, since we utilize the K dimensional optimal lattice quantizers and assume that input message data is i.i.d., second order statistics of the noise variances is given by [99, 105]:

$$\mathbb{E}\{w_i^2(t)\} = C_K \mathbb{E}\{d_i^2(t)\} 2^{-2R(t)} \quad (3.6)$$

where $\mathbb{E}\{\cdot\}$ denotes the statistical expectation and C_K is a constant whose value is a function of the quantizer dimension K . We note that state and noise statistics, and quantization rates are per symbol (dimension).

At this point we would like to remind our readers Lemma 1 of Chapter 2.

Lemma 7 *The nodes converge to a consensus in mean squared sense, if and only if the noise variance at each sensor converges to 0, i.e.. $\mathbb{E}\{w_i^2(t)\} \rightarrow 0$ as $t \rightarrow \infty \forall i \in \mathcal{V}$.*

We will use the above lemma to prove the convergence of both constant rate and variable rate schemes in Sections 3.3 and 3.4.

3.3 Characterization of the MSE Performance

In this section, we first detail the graph model adopted throughout the rest of the chapter. Then, we give the convergence proof of the quantized consensus under fixed quantization rate and derive the explicit MSE performance of the proposed algorithm. Moreover, by presenting an upper-bound on the MSE expression, we relate the performance of the algorithm to the rate, graph connectivity and the initial conditions.

3.3.1 2-Dimensional Regular Networks

MSE characterization of the quantized average consensus is quite challenging for general networks. For this reason, we restrict ourselves to a narrower communication pattern, namely 2 dimensional symmetric regular graph topology and in return we can provide explicit MSE expression and scaling laws. On the other hand, by using the relationship between large random geometric graphs and regular graphs, we can show that our MSE characterization indeed models the behavior of the quantized consensus algorithm on large sensor networks.

A random geometric graph denoted by $\mathcal{G}(N, r)$ is a graph where N are nodes uniformly distributed onto the surface of the d -dimensional unit torus in a random fashion and two nodes are connected if and only if internode distances are less than some value r . Such graphs are discussed to closely model the behavior of the sensor networks [42, 19] in the case of $d = 2$. It was also shown that the degree (the number of neighbors) of each node becomes $\Theta(Nr^2)$ with high probability for sufficiently large N and $r > 2\sqrt{\log(N)/N}$ where $\Theta(\cdot)$ is the asymptotic upper and lower bound of its argument [19]. Moreover, a regular graph is a graph where each vertex has the same number of neighbors. Therefore, for large N and $r > 2\sqrt{\log(N)/N}$, the connectivity of a geometric random network becomes identical to that of a regular deployment. We will summarize our discussion with the following Remark.

Remark 8 *For $r > 2\sqrt{\log(N)/N}$, the connectivity of a geometric random network becomes identical to that of a regular deployment as $N \rightarrow \infty$.*

Before we discuss some of the nice properties of the regular graphs, we give the definition of circulant and block circulant matrices:

Definition 1 An $N \times N$ matrix A is called circulant and denoted $A = \text{circ}(a_0, a_1, \dots, a_{N-1})$ if its elements satisfy

$$A_{j,k} = a_{(N-1)j+k \bmod(N)} \quad \forall j, k \in \{1, \dots, N\}. \quad (3.7)$$

Definition 2 An $N^2 \times N^2$ matrix A is called block circulant with circulant blocks if it is of the form:

$$A = \text{circ}(A_0, A_1, \dots, A_{N-1}) \quad (3.8)$$

where A_i is $N \times N$ and circulant $\forall i \in \{0, \dots, N-1\}$.

Following the definitions above, the adjacency matrix (2.1) of a regular network satisfies the following property:

Remark 9 The adjacency matrix A of a regular graph with N nodes on a 2-dimensional square grid is block circulant with each $(\sqrt{N} \times \sqrt{N})$ block is circulant in itself [28].

Since we are to model the behavior of (3.5), we are interested in weight matrix W which can be expressed in terms of adjacency matrix A as:

$$W = I - \epsilon(\text{diag}(A\mathbf{1} - A)) \quad (3.9)$$

where I is the $N \times N$ identity matrix, $\mathbf{1}$ is all ones column vector, diag represents diagonal matrix. $0 < \epsilon < 1/\max(A\mathbf{1})$ constraint on ϵ guarantees convergence in ideal communication case [76], so $\epsilon \in (0, 1/\max(A\mathbf{1}))$. Since all of the matrices in (3.9) are block circulant with circulant blocks (BCCB) and BCCB matrices form a commutative algebra, W is also a BCCB matrix. Since the underlying connectivity is symmetric, so is W matrix. Therefore, $(W\mathbf{1})^T = \mathbf{1}^T W = \mathbf{1}^T$. Thus, such W matrix is not only row stochastic but also column stochastic, *i.e.*, doubly stochastic.

Remark 10 All $(N \times N)$ block circulant matrices with $(\sqrt{N} \times \sqrt{N})$ circulant blocks are simultaneously diagonalizable by the unitary matrix $F \otimes F$ where F is an $(\sqrt{N} \times \sqrt{N})$ matrix whose columns are FFT basis of length \sqrt{N} and \otimes is the Kronecker product operation. The eigenvalues of such a matrix is the transformation of the first column of the matrix with $F \otimes F$.

We will utilize Remarks 9 and 10 throughout the chapter to derive MSE performance and scaling laws of quantized average consensus algorithm.

3.3.2 Mean Square Error

We first show that quantized consensus algorithm converges under fixed rate quantization. Since necessary and sufficient condition for convergence is given by Lemma 7 as the convergence of noise variances to 0 at each sensor, we first model the behavior of the noise variances.

State covariance matrix of the network which evolves by (3.5) follows the recursion:

$$\mathbb{E}\{x(t+1)x^T(t+1)\} = W\mathbb{E}\{x(t)x^T(t)\}W + W\mathbb{E}\{w(t)w^T(t)\}W. \quad (3.10)$$

We note that $\mathbb{E}\{w(t)w^T(t)\}$ is a diagonal matrix due to the correlation assumptions on the quantization noises. Moreover, we can show that covariance matrix of the network states preserves BCCB property through iterations by the following Lemma:

Lemma 8 Given a regular graph, BCCB initial sensor correlation matrix $\mathbb{E}\{x(0)x^T(0)\}$ and uniform quantization rate among the sensors, i.e, each sensor encodes with rate R at each step, then $\mathbb{E}\{x(t)x^T(t)\}$ and $\mathbb{E}\{w(t)w^T(t)\}$ are BCCB $\forall t \geq 1$.

Proof The proof is by induction. Since initial sensor correlation matrix is BCCB and quantization rates are uniform, initial noise covariance matrix is also BCCB. Since BCCB matrices form a commutative algebra and the matrices on the right side of (3.10) are BCCB for $t = 0$, then state correlation matrix is also BCCB for $t = 1$. Above argument is repeated for $t \in \{1, 2, \dots\}$ and the result follows.

Let's denote

$$\Sigma(t) = \mathbb{E}\{x(t)x^T(t)\} \quad (3.11)$$

$$\Upsilon(t) = \mathbb{E}\{w(t)w^T(t)\} \quad (3.12)$$

Since for a given $t \geq 0$ both $\Sigma(t)$ and $\Upsilon(t)$ are BCCB by Lemma 8, and W is also BCCB by construction, they are all diagonalizable by $F \otimes F$ eigenmatrix, *i.e.*, for any BCCB matrix Z we have that $(F \otimes F)^{-1} Z (F \otimes F) = \text{diag}(z_1, z_2, \dots, z_N)$ where z_1, z_2, \dots, z_N represents the eigenvalues of Z . Denote eigenvalues of W corresponding to the eigenmatrix $(F \otimes F)$ as $\omega_j, 1 \leq j \leq N$. WLOG, assume that the columns of $(F \otimes F)$ are ordered such that $1 = w_1 > w_2 \geq w_3 \dots \geq w_N > -1$. Denote the eigenvalues of $\Sigma(t)$ and $\Upsilon(t)$ corresponding to the ordered eigenmatrix as $\sigma_j(t)$ and $v_j(t)$. We note that $\Upsilon(t)$ is a diagonal matrix and diagonal elements are equal due to BCCB structure. Thus, $v_j(t) = v_k(t) = E\{w_i^2(t)\} \quad k, t, i \in \{1, \dots, N\}$. For mathematical brevity, we will denote these repeated eigenvalues by a single variable, namely $v(t)$.

Following, the update equation in (3.10) can be written as a system of N difference equations as follows:

$$\sigma_j(t) = \omega_j^2(\sigma_j(t-1) + v(t-1)), \quad (3.13)$$

$\forall t \geq 1, 1 \leq j \leq N$ and initial conditions $\sigma_j(0)$ and $v(0)$. On the other hand,

combining (3.2) and (3.6), it is easy to see that:

$$v(t) = \frac{C_K}{N2^{2R}} \left(\sum_{j=1}^N (\omega_j - 1)^2 (\sigma_j(t-1) + v(t-1)) \right). \quad (3.14)$$

We note that $\omega_1 = 1$, therefore the index of the summation in (3.14) may be switched to $\{j\}_2^{2^N}$. One can combine equations (3.13) and (3.14) to obtain a linear difference system with constant coefficients. In particular, defining $Y(t) = [\sigma_1(t), \sigma_2(t), \dots, v(t)]^T$, $Y(t)$ vector follows the recursion $\forall t \geq 1$:

$$Y(t) = HY(t-1), \quad (3.15)$$

where

$$H = \begin{bmatrix} 1 & 0 & 0 & \dots & 1 \\ 0 & \omega_2^2 & 0 & \dots & \omega_2^2 \\ \vdots & 0 & \ddots & \vdots & \vdots \\ 0 & \vdots & \dots & \omega_N^2 & \omega_N^2 \\ 0 & \frac{C_K(\omega_2-1)^2}{N2^{2R}} & \dots & \dots & \frac{C_K \sum_{j=2}^N (\omega_j-1)^2}{N2^{2R}} \end{bmatrix}. \quad (3.16)$$

We note that (3.16) tracks the eigenvalues of the state covariance and the noise variance matrices. Since $v(t)$ is also equal to the quantization noise at any node $i \in \mathcal{V}$ at time t , the linear difference system above also identifies the evolution of the quantization noise in the system.

The following lemma shows that quantized consensus algorithm with predictive coding scheme converges to a consensus in \mathcal{L}^2 , *i.e.*, mean squared sense.

Lemma 9 *QCPC scheme converges to a consensus in \mathcal{L}^2 for any R such that*

$$R > \log \left(C_K \frac{\max_{i \in \mathcal{V}'} (\omega_i - 1)^2 \max_{i \in \mathcal{V}'} \omega_i^4}{\min_{i \in \mathcal{V}'} \omega_i^2 \min_{i \in \mathcal{V}'} 1 - \omega_i^2} + 4 \right), \quad (3.17)$$

where $\mathcal{V}' = \{i | i \in \{2, \dots, N\}, w_i^2 > 0\}$.

Proof The proof is given in Appendix B.1.

In the rest of the section, we will assume that the quantization rate R satisfies the condition given above. We denote the average of the state values at time step t by $a(t) = 1/N \sum_{i=1}^N x_i(t)$, and recall that we are interested in the behavior of the expected mean squared distance between the asymptotic average and initial average, *i.e.*,

$$\delta_\infty \triangleq \lim_{t \rightarrow \infty} \delta(t) = \lim_{t \rightarrow \infty} \mathbb{E}\{(a(t) - a(0))^2\}. \quad (3.18)$$

It is easy to see that

$$a(t+1) = a(t) + \frac{1}{N} \sum_{i=1}^N w_i(t) \Rightarrow a(t) = a(0) + \frac{1}{N} \sum_{l=0}^{t-1} \sum_{i=1}^N w_i(t). \quad (3.19)$$

Moreover, recall that the noise samples are taken to be uncorrelated both temporally and spatially. Thus, integrating the uncorrelated assumption along with (3.19) reduces the limiting MSE expression to

$$\delta_\infty = \lim_{t \rightarrow \infty} \frac{1}{N^2} \sum_{l=0}^{t-1} \sum_{i=1}^N \mathbb{E}\{w_i^2(t)\}. \quad (3.20)$$

Moreover, due to the special structure of the underlying graph, $E[w_i^2(t)] = v(t), \forall i, t$ and the expression given above becomes:

$$\delta_\infty = \lim_{t \rightarrow \infty} \frac{1}{N} \sum_{l=0}^{t-1} v(t). \quad (3.21)$$

We note that we have a complete characterization of $v(t)$ in (3.15). Therefore, MSE of our algorithm can be calculated by analyzing the linear difference equation given in (3.15).

Lemma 10 *For sufficiently large N and R , MSE of the QCPC with constant quantization rate R is given by*

$$\delta_\infty = \frac{C_K}{N^2 2^{2R}} \left(\sum_{j \in \mathcal{V}'} \frac{(1 - \omega_j)^2}{1 - \omega_j^2} \sigma_j(0) \right) \quad (3.22)$$

where ω_j , are the ordered eigenvalues of W matrix and $\sigma_j(0)$ are the eigenvalues of $\mathbb{E}\{x(0)x^T(0)\}$ corresponding to the ordered eigenvectors of W matrix.

Proof The proof is given in Appendix B.2.

In the following, we derive upper-bounds on the MSE performance of the quantized distributed average consensus where we relate the performance to parameter of interests, such as the connectivity of the underlying graph modeling the sensor network which can be inferred from the second largest eigenvalue of the weight matrix [93, 92]. Thus, the following corollary relates the MSE performance of the proposed algorithm to the second largest eigenvalue of the weight matrix.

Corollary 2 *Suppose $|\omega_2| > |\omega_j|, \forall j \geq 2$, then, the limiting MSE of the QCPC is upper bounded by*

$$\delta_\infty < \frac{4C_K}{N2^{2R}} \frac{1}{1 - \omega_2^2} \max_{j \neq 1} \sigma_j(0). \quad (3.23)$$

Proof The proof is give in Appendix B.3.

Upper bounds on the MSE performance of the quantized consensus algorithms are also given in [20, 10]. While corresponding scenarios are slightly different (*i.e.*, initial states are deterministic rather than random and quantizer is probabilistic rather than adaptive), the MSE dependence on the second largest eigenvalue of the weight matrix has the same form.

In the following, we compare theoretical MSE expression and bound with the simulated performance. Fig. 3.2 depicts the performance of the algorithm where uniform *scalar* quantizers are used at each sensor. A regular graph with 400 nodes is generated where each node has 48 neighbors. Initial observations are i.i.d Gaussian samples $\mathcal{N} \sim (0, 1)$. The algorithm is simulated over 1000

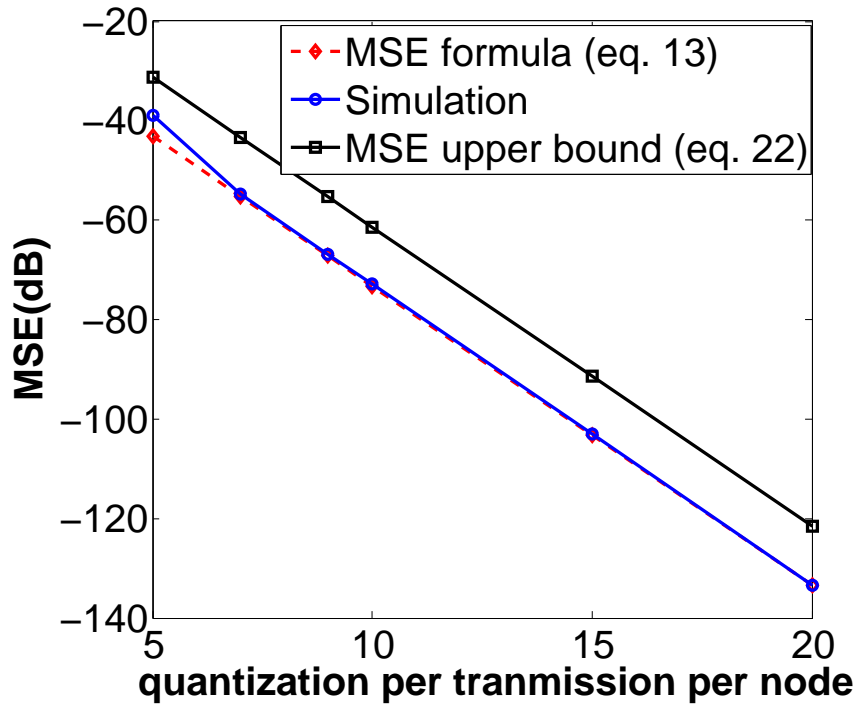


Figure 3.2: Simulated versus theoretical MSE performance of QCPC.

Monte-Carlo runs. The quantization parameter C_K is chosen as 24 in this sub-optimum scenario. We conclude that the MSE expression in (3.22) closely reflects the behavior of the actual system. Of note is that when the rate is small enough, *i.e.* less than five bits in our simulation, the gap between the simulated and predicted performance is large. Although MSE expression in (3.22) is valid under sufficiently large network and rate, simulation results show that even a network of 400 nodes and 5 bits per sensor is large enough. We also note that upper bound in (3.23) follows the scaling of the actual performance.

Finally, we simulate the MSE performance of QCPC algorithm in both regular and random graphs in order to show that the performance of QCPC is similar for both networks. Fig. 3.3 shows the MSE performances of the networks with two different sizes ($N = 64$ and $N = 400$) and two different configurations;

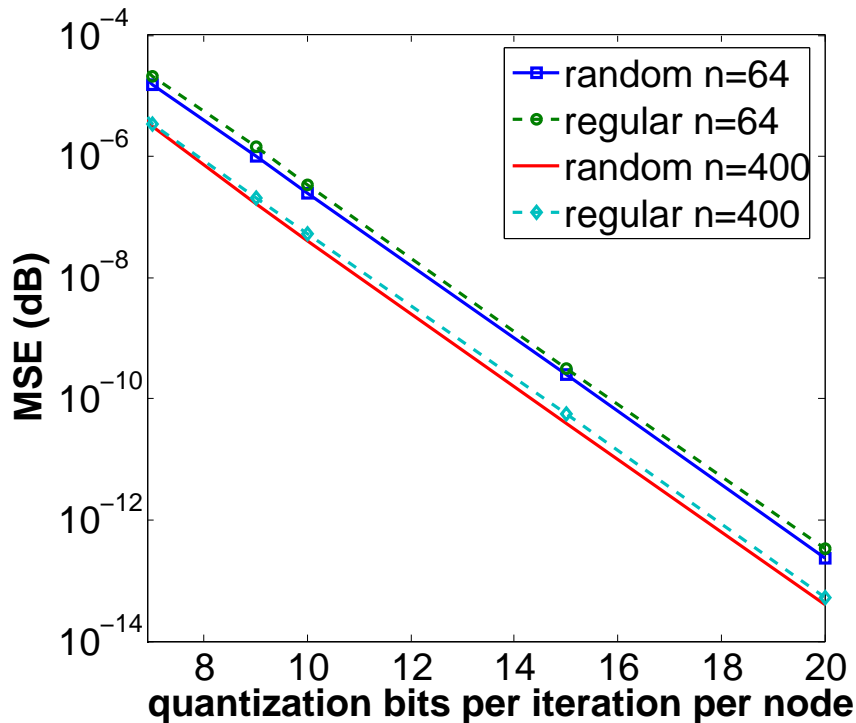


Figure 3.3: QCPC MSE (dB) performance for random and regular networks.

random deployment and regular deployment. For each N , a regular network with connectivity radius 0.35 is generated. Furthermore, 5 random networks with the same connectivity radius are also generated. We simulate the algorithm over 1000 Monte-Carlo runs where the initial observations are i.i.d. realizations of the normal distribution (Independence and Gaussian distribution indeed characterizes the worst-case behavior of the algorithm). We average the MSE behavior over the Monte-Carlo runs and the random networks and employ this average as the behavior of the random network. We conclude that the regular network analysis closely reflects the behavior of the random networks in average consensus algorithms with quantization noise. We observe the exponentially decreasing behavior of the MSE with respect to the quantization rate as predicted by (3.22). We also note that larger networks have smaller MSE.

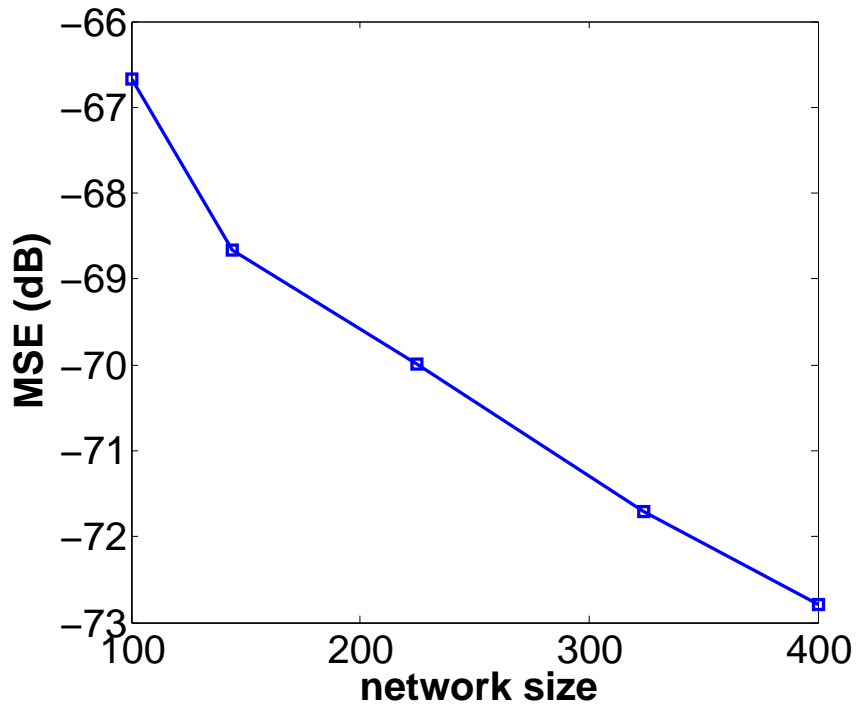


Figure 3.4: Network size versus MSE performance of QCPC.

3.3.3 Scaling Laws in Regular Networks

Note that MSE expression is $\mathcal{O}(N^{-1})$, see (3.22). Therefore, cumulative quantization error tends to zero as $N \rightarrow \infty$. This is a pleasing result, but the caveat is that it is valid provided that the quantizer adaptively changes so that it is possible to consider the quantization error uncorrelated. If one had not used a differential encoding scheme, one would have required increasing quantization rate over the iterations.

Next, we simulate the performance of the network in order to understand the actual behavior of the MSE with respect to the network size. Fig. 3.4 shows the MSE behavior of the system with respect to the network size. Simulation parameters are fixed as given above. We conclude that the actual MSE scales as $\mathcal{O}(N^{-1})$. We note that such a behavior is also given in both MSE formula and

MSE upper bound, (3.22) and (3.23) respectively.

We further calculate the scaling laws of the MSE for two dimensional regular graphs in terms of the number of neighbors. We consider the model studied in [19]: An edge between two vertices exists if the maximum norm (L_∞) distance between them is less than k/\sqrt{N} where $k \in \mathbb{Z}$. The following lemma characterizes the scaling behavior of MSE with respect to the number of neighbors of each node:

Lemma 11 *Given $k \ll \sqrt{N}$ and initial conditions (node correlation, quantization rate) are fixed,*

$$\delta_\infty = \mathcal{O}(k^{-2}). \quad (3.24)$$

Proof The proof is given in Appendix B.4.

For a given $k \geq 1$, the number of neighbors is simply $4k(k+1) = \mathcal{O}(k^2)$. Therefore, one can conclude that MSE decreases almost linearly with the number of neighbors for large k in 2-D regular graphs. We note that, if the number of neighbors is fixed as the network size increases, the graph becomes less connected and $\mathcal{O}(1/N)$ scaling in (3.23) is canceled out and MSE does not scale with respect to N , *i.e.*, $\mathcal{O}(1)$.

In the following, we have simulated MSE performance of the system with respect to the connectivity radius and results are given in Fig. 3.5. Of note is that connectivity radius is simply $\sqrt{4k(k+1)/N}$. As we have discussed above, MSE decreases as radius of connectivity increases. We also note that as the number of nodes increases, decrease in MSE is much sharper and it saturates around smaller connectivity radius.

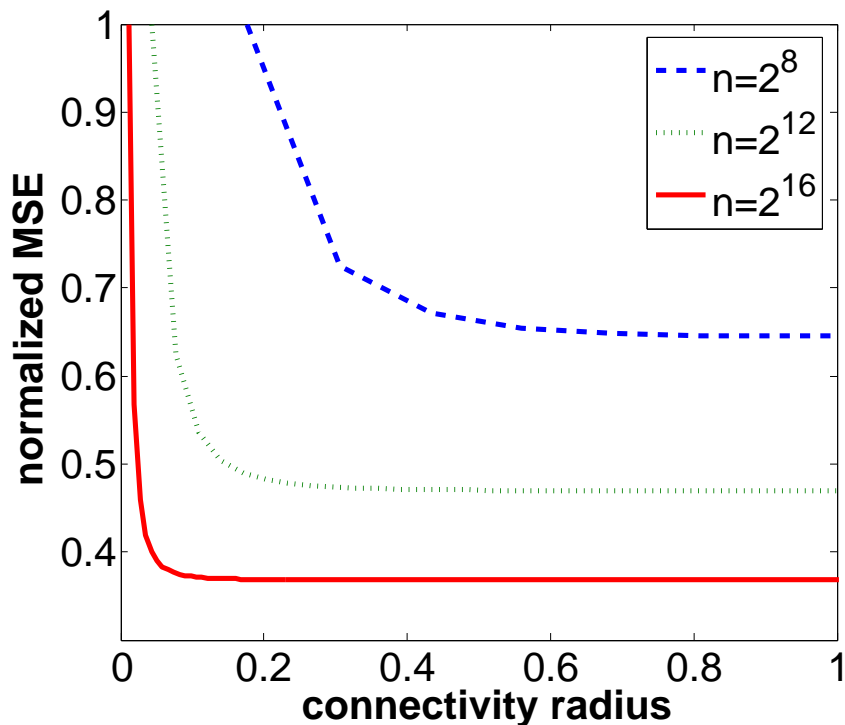


Figure 3.5: QCPC MSE versus connectivity radius.

3.4 Asymptotic Rate Behavior and Rate Allocation

In this section, we focus on the case where the quantization rate can change from iteration to iteration. First, we study the asymptotic rate behavior of the consensus algorithm under bounded convergence. Moreover, we completely characterize the rate regions achieving consensus with bounded MSE with vanishing quantization rate and finite-sum rate. We also study optimal rate allocation among the iterations when the allowable sum-rate is fixed.

3.4.1 Asymptotic Rate Behavior

In this section, we explore the consensus behavior under variable rates. Under the condition that the noise variances form a convergent geometric series, we fully characterize the asymptotic rate behavior in terms of the initial conditions, quantization parameters and the connectivity of the network. Furthermore, we calculate quantization rate regions which guarantee both vanishing rates and bounded convergence. We further show that such rate regions also achieve bounded rate-sum.

We will use the difference system defined in (3.13) and (3.14) which are reiterated here for completeness:

$$\sigma_j(t) = \omega_j^2(\sigma_j(t-1) + v(t-1)) \quad (3.25)$$

$$v(t) = \frac{C_K}{N2^{2R(t)}} \left(\sum_{j=1}^N (\omega_j - 1)^2 (\sigma_j(t-1) + v(t-1)) \right) \quad (3.26)$$

$$R(t) > 0, \forall t \geq 1. \quad (3.27)$$

The constraint on $R(t) > 0$ guarantees that the above system represents a valid quantization scheme (non-positive rate is not possible, and since we are interested in the asymptotic behavior we exclude the case where $R(t) = 0$ for finite t). At each iteration, $(t+1)$, we would like the noise variances (eigenvalues) to reduce as:

$$\frac{v(t+1)}{v(t)} = \beta \quad (3.28)$$

where $0 < \beta < 1$. Of note is that this range of β guarantees bounded convergence, *i.e.*:

$$\delta_\infty = \frac{1}{N} \sum_{t=0}^{\infty} v(t) = \frac{v(0)}{N} \sum_{t=0}^{\infty} \beta^t = \frac{v(0)}{N} \frac{1}{1-\beta} < \infty. \quad (3.29)$$

Clearly, the geometric convergence of the noise variances is not the only way to obtain convergence to a consensus with bounded MSE. More general cases can be captured by allowing β to be a function of the iteration index. For mathematical simplicity, we focus on the special case of geometric convergence. First, having the geometric reduction of quantization noise variances in mind, one need to characterize the rate recursion needed to achieve such reduction. Given $(0 < \beta < 1)$, network connectivity $(\omega_1, \dots, \omega_N)$, and initial conditions $(\sigma_1(0), \dots, \sigma_N(0), R(0))$, quantization rates follow the non-linear recursion for all $t \geq 1$ (see Appendix B.5):

$$R(t+1) = \frac{1}{2} \log \left(2^{2R(t)} \frac{\sum_{j=2, \omega_j \neq 0}^N (\omega_j - 1)^2 \sigma_j(t)}{\sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2}{\omega_j^2} \sigma_j(t)} + \frac{C_K}{N} \sum_{j=2}^N (\omega_j - 1)^2 \right) - \frac{1}{2} \log \beta. \quad (3.30)$$

Equation (3.30) has no closed form solution, but further insight on its behavior is gained in the following lemma:

Lemma 12 *The quantization rate for the average consensus under predictive coding scheme defined through (3.25)-(3.27) converges to:*

$$R^* \triangleq \lim_{t \rightarrow \infty} R(t) = \frac{1}{2} \log \left(\frac{C_K}{N(\beta - K^*)} \sum_{i=2}^N (\omega_i - 1)^2 \right) \quad (3.31)$$

where:

$$K^* = \frac{\sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2 \omega_j^2}{\beta - \omega_j^2}}{\sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2}{\beta - \omega_j^2}} \quad (3.32)$$

and $\omega_j^2 < \beta < 1$ for all $j \in \{1, \dots, N\}$.

Proof The proof is given in Appendix B.6.

Equation (3.31) is the characterization of the asymptotic rate subject to the constraint that convergence to a consensus with bounded MSE is achieved. Note

that above expression is a function of the network connectivity, noise variance behavior and quantization parameters. It is interesting to observe that it is neither a function of the initial quantization rate nor the initial observation correlation. However, the initial quantization rate and the initial correlation will affect the limiting MSE which depends on the initial noise variance $v(0)$.

Of note is that the noise decay factor β is restricted as follows by Lemma 12: $\max_j \omega_j^2 < \beta < 1$. As the network connectivity increases, *i.e.*, $\max_j \omega_j^2$ decreases, a wider range of β can be utilized and a sharper decrease in noise variance, hence, faster convergence, is possible. Given initial conditions and noise decaying factor, asymptotic MSE is simply $v(0)(1 - \beta)^{-1}/N$ by (3.29). Hence, smaller β (faster decrease in the noise variance) will allow the algorithm to converge to a consensus with a smaller MSE for given $v(0)$.

We are particularly interested in the rate allocation schemes achieving consensus with bounded MSE with vanishing rate, *i.e.*, $\delta_\infty < \infty$ and $\lim_{t \rightarrow \infty} R(t) = 0$. Such a behavior, if it exists, has a particular importance since it is an insight to the fact that consensus can still be achieved with decreasing communication cost. As a first step for deriving such rate regions, we make the following observation:

Lemma 13 *Let R^* given as in (3.31) and $\omega_j^2 < \beta < 1$ for $j \in \{2, \dots, N\}$. Then, R^* decreases with increasing β given the network and quantization parameters.*

Proof The proof is given in Appendix B.7.

Above result indicates that, as we decrease the speed of noise decay, the limiting quantization rate decreases. Such a behavior is intuitive since one can

think this phenomena as a trade-off between rate and distortion (MSE). The next question of interest is the existence of a $\beta \in (\max_j \omega_j^2, 1)$ which satisfies the equality $R^* = 0$, *i.e.*, are there particular noise decay rate regions tending to zero cost in the limit? The following lemma proves that such β indeed exists.

Lemma 14 *If $|\omega_2| > |\omega_j|$ for $j = 3, 4, \dots, N$, and the eigenvalues satisfy the following inequality for some small $\epsilon > 0$,*

$$\omega_2^2 - L(\omega) + \epsilon < \omega_j^2 < 1 - L(\omega) - \epsilon, \forall j \quad (3.33)$$

where

$$L(\omega) = \frac{1}{N} \sum_{j=2}^N (1 - \omega_j)^2 \quad (3.34)$$

and $\omega = \{\omega_j : j = 2, 3, \dots, N\}$, then, there exists a unique β such that $\omega_2^2 < \beta < 1$ and $R^* = 0$.

Proof The proof is given in Appendix B.8.

Since in general, the noise variance does not have to decrease as a geometric series the lemma above gives conditions on the underlying graph that permit this behavior with vanishing rate. We note that the regular graphs we considered throughout meet condition (3.33).

A more important result from the communication cost perspective is the following:

Lemma 15 *Assume \exists a $\beta \in (0, 1)$ achieving vanishing asymptotic quantization rate ($R^* = 0$) as given in Lemma 14. Then under such β , the sum of the quantization rates over the iterations is finite, *i.e.*,*

$$\sum_{t=0}^{\infty} R(t) < \infty. \quad (3.35)$$

and MSE is bounded.

Proof The proof is given in Appendix B.9.

In words, although QCPC requires infinite number of iterations, the total transmission time per node is finite. In a similar fashion, one can conclude that the energy required for inter-node transmission is also finite.

Since our proofs are constructive, it is possible to formulate a simple rate allocation policy that will satisfy both Lemmas 14 and 15. Under the conditions given in Lemma 14, rate-distortion (MSE) regions satisfying zero asymptotic rate with bounded error is calculated by the following algorithm:

1. Determine β converging quantization rate R to 0 by (3.31).
2. Given target MSE, *i.e.* MSE_∞ , calculate initial quantization rate by (3.29).
3. Determine quantization rates at each step, $R(t)$ via (3.30).

Remark 11 *Similar to the behavior fixed rate consensus in Section 3.3, MSE of the variable rate QCPC is $\mathcal{O}(N^{-1})$ when the rest of the parameters are fixed (see (3.29)). The result is interesting in a way that although decreasing number of bits is utilized, scaling with respect to network size does not change.*

We also observe that MSE expression for the variable rate consensus in (3.29) is an upper bound for the MSE performance of the variable rate consensus in (3.23) under the same initial conditions. The statement is clear if one notes that $\beta > \omega_2^2$ and initial noise variance ($v(0)$) is a function of the rate, the network size and the trace of the initial node correlation function.

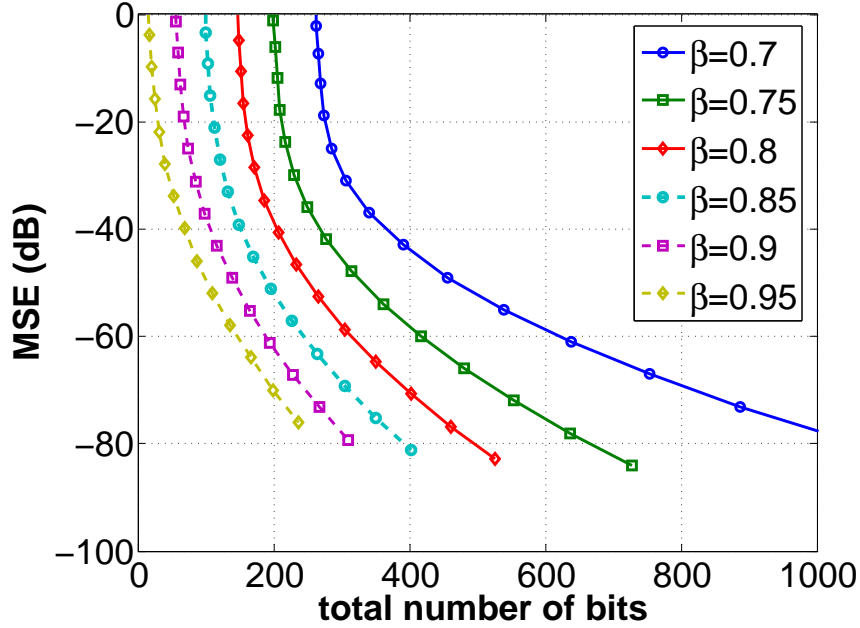


Figure 3.6: MSE (dB) vs sum of the quantization rates for different β over 1000 iterations.

3.4.2 Rate Allocation

In this section, we focus on the numerical solutions of (3.30) and infer some intuition about the optimal distributions of the quantization rates in the MSE sense. For a 2-D regular graph of 64 nodes, connectivity radius 0.25 and initial observations uncorrelated (*i.e.*, $\sigma_i(0) = \sigma_j(0)$, $\forall \{i, j\}$), the total number of bits spent over one thousand iterations versus the final MSE is given in Fig. 3.6. For each MSE- β pair, the algorithm is initialized with a quantization rate $5 \leq R(0) \leq 20$ and $R(t)$ is calculated by (3.30) for a given $0.70 \leq \beta \leq 0.95$. We note that such a family of β satisfies $\beta > \max_j \omega_j^2$. We observe that given a total number of bits to be spent, one can achieve a lower MSE by utilizing a larger β . Fig. 3.7 shows quantization rates over iterations for, $\beta = 0.95$, $\beta = 0.90$ and $\beta = 0.85$ such that the total number of bits spent is fixed, *i.e.*, $\sum_{t=0}^{200} R(t) \simeq 225$. For $\beta = 0.95$, the algorithm starts with 20 bits per node and slowly converges to 0.0050 bits

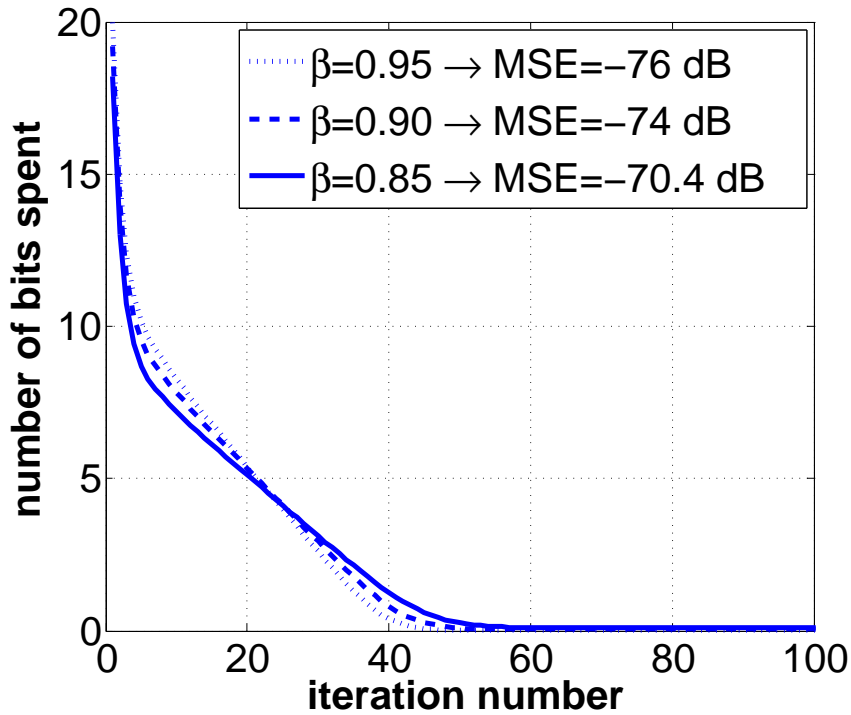


Figure 3.7: MSE (dB) vs sum of the quantization rates for different β .

per node. On the other hand, for $\beta = 0.90$, the algorithm initializes with 19.2 bits per node, stays under the $\beta = 0.95$ curve initially, then crosses the curve and follows a similar behavior converging to 0.047. The $\beta = 0.85$ curve starts at 18.2 bits per node and converges to 0.091 slower than both 0.90 and 0.95 curves. The MSE performances of these schemes are different, *i.e.*, -76dB , -74dB and -70.4dB respectively, since as we noted in Fig. 3.6, a larger β results in a better MSE performance for a fixed number of total bits spent.

Compiling the theoretical and numerical results, we note that a slower decrease in the noise variances will result in a smaller MSE when the total number of quantization bit to be spent is fixed. In other words, transmitting more bits in the beginning and decreasing transmission cost gradually results in a better performance.

3.5 Discussions

In this study, we have studied rate constrained average consensus problem under both *fixed* and *variable* rate quantization rate scenarios. We have focused on simple predictive coding scenario where at time instant t , nodes transmit only the innovation with respect to previous state value. In the case of fixed rate quantization, we have presented a closed form MSE expression and corresponding upper bounds in terms of the network connectivity, quantization rate and initial node correlation. We have also derived asymptotic behavior of the quantization rate in a scenario where quantization rates are variable and consensus with finite MSE is achieved. In a special case where the quantization rates are chosen such that quantization noises decrease like a geometric series, we have determined the rate regions where asymptotic quantization rate vanishes such that sum-rate and MSE is bounded.

CHAPTER 4

BROADCAST GOSSIPING ALGORITHM

4.1 Motivation and Related Work

As it was discussed in Chapter 1, geographic-type gossiping improves upon the convergence speed of the standard gossip by increasing the diversity of pairwise exchanges. However, the problem of packet loss is exacerbated by the requirement that messages must be sent on long routes, creating congestion issues. Moreover, it does not mitigate the major bottleneck associated with the fact that the messages between two peers need to be routed and exchanged to perform two updates. Finally, successfully setting up a two-way route exacerbates the problem by requiring information about the location of the nodes in the network.

The wireless medium has the advantage of being inherently broadcast and, at the cost of one transmission, one can reach several terminals. Our objective in this chapter is to analyze a broadcasting-based gossip algorithm that enables *all nodes in range* to perform an update by exploiting the wireless medium, and thereby avoiding the need of complex routing and problematic pairwise exchange operations.

4.1.1 Summary of Main Contributions

To overcome the drawbacks of the standard packet based gossip algorithms, we study a broadcast based gossiping algorithm for wireless sensor networks. In

the studied algorithm, a node in the network wakes up uniformly at random according to the asynchronous time model and broadcasts its value. This value is successfully received by the nodes in the predefined radius of the broadcasting node, *i.e.*, connectivity radius. The nodes that have received the broadcasted value update their own state value and the remaining nodes sustain their value. It is shown here that by iterating this procedure, this type of gossiping algorithm is capable of achieving consensus over the network with probability one. We also show that the random consensus value is, in expectation, equal to the desired value, *i.e.*, the average of initial node measurements. Because the sum of the node state values is not preserved at each iteration, the broadcast gossiping algorithm converges to a value that is in the neighborhood of the desired average.

The question that motivates this chapter is investigating if it is possible to avoid the partner selection process altogether, analyzing a broadcast communication protocol where each random transmission triggers an update by all nodes within range, without a mechanism of reply in place to maintain the network average. We initiated this study in [9]. Fagnani and Zampieri have concurrently studied the convergence to consensus characteristics of general randomized algorithms which do not necessarily converge to the initial average (such as asymmetric gossip, broadcast gossip and packet-drop gossip) [38]. In particular, the authors have shown that random consensus algorithms in general achieve probabilistic consensus, and discussed their mean squared error characteristics (Proposition 4.4 and Corollary 3.2). In this chapter, we provide an in depth study of broadcast gossip algorithms' speed of convergence and mean squared error characteristics. Our results also address the choice of the *mixing parameter* and its effect on both the mean square error and the convergence rate,

which provides insight for implementation.

More specifically, we provide theoretical and simulation results on the mean square error and communication cost performance of the broadcast gossip algorithm. Moreover, we study the effect of the so called *mixing parameter* on the convergence rate and limiting mean square error through theoretical results and numerical experiments. In addition, we derive the optimal mixing parameter when approached from the convergence rate perspective. Although the convergence time of our algorithm is commensurate with the standard pairwise gossip algorithms, we present simulations showing that for more modest network sizes our algorithm converges to consensus faster than other algorithms based on pairwise averages or routing.

4.1.2 Chapter Organization

The remainder of this chapter is organized as follows. Section 4.2 introduces the average consensus problem and the graph and time models adopted in this study. The studied broadcast gossip algorithm is introduced in Section 4.3 and its convergence characteristics are studied in Section 4.4. In Section 4.5 we derive the optimal mixing parameter considering the worst-case convergence rate and analyze the effects of various network parameters on the optimal value. The MSE characterization and communication complexity analysis are given in Section 4.6 along with the convergence rate expression. Finally, we conclude with some discussion and future directions in Section 4.7.

4.2 Graph and Time Models

In the following, we briefly discuss the graph and time models adopted in this chapter. Then, we describe briefly the distributed average consensus problem.

4.2.1 Graph Model

We model our wireless sensor network as a random geometric graph $G(N, R)$, where the N sensor locations are chosen uniformly and independently in a unit square area, and each pair of nodes is connected if their Euclidean distance is smaller than some transmission radius R named *connectivity radius* [41]. For our analysis, we assume that a communication within this transmission radius always succeeds. It is well known that in order to have a fully connected network asymptotically while minimizing interference, the connectivity radius R has to scale like $\Theta(\sqrt{\log(N)/N})$ [41, 20]. The N -node topology of $G(N, R)$ is represented by the $N \times N$ adjacency matrix Φ , where for $i \neq j$, $\Phi_{ij} = 1$ if nodes i and j are in their neighborhood, and $\Phi_{ij} = 0$, otherwise. Moreover, we define $\mathcal{N}_i = \{j \in \{1, 2, \dots, N\} : \Phi_{ij} \neq 0\}$ and D as a diagonal matrix with entries $D_{ii} = \mathcal{N}_i$. Finally, the Laplacian of a graph is defined as $L = D - \Phi$.

4.2.2 Time Model

We use the asynchronous time model, which is well-matched to the distributed nature of sensor networks [20, 34]. In this model, each sensor node is assumed to have a clock which ticks independently according to a rate μ Poisson process.

Consequently, the inter-tick times are exponentially distributed and independent across nodes and over time. This process is equivalent to a single clock whose ticking times form a Poisson process of rate $N\mu$. Let Z_t be the arrival times of this global process. In expectation, there are approximately $N\mu$ clock ticks per unit of absolute time but we will always measure time in number of ticks of this (virtual) global clock. We therefore think of time as discretized with the interval $[Z_t; Z_{t+1})$ corresponds to the t -th timeslot. We can adjust time units relative to the communication time so that only one broadcast event occurs in the network at each time slot with high probability.

4.2.3 Average Consensus

At time slot $t \geq 0$, each node $i = 1, 2, \dots, N$ has an estimate $x_i(t)$ of the global average, and we use $x(t)$ to denote the N -vector of these estimates. The ultimate goal is to use the minimal amount of communication to drive the estimate $x(t)$ as close as possible to the average vector $\bar{x}(0)\mathbf{1}$, where $\mathbf{1}$ is the vector of all 1's and

$$\bar{x}(0) = \frac{1}{N} \sum_{i=1}^N x_i(0). \quad (4.1)$$

Because our algorithms are randomized, the quantity $x(t)$ for $t > 0$ is a random vector even though we assume $x(0)$ is deterministic.

4.3 Broadcast Based Gossiping

Informally, the asynchronous broadcast gossip algorithm is described as follows. Suppose node i 's clock is the t -th that ticked. Then, node i broadcasts

its own state value which is received by all neighboring nodes within distance R from it. Once the broadcasted value is received, the neighboring nodes set their values equal to the (weighted) average of their current value and the value broadcasted by the node i . Formally, node i activates and the following events occur:

- Node i broadcasts wirelessly its current state value, $x_i(t)$.
- The broadcasted value is successfully received by the nodes that are within the radius R .
- All nodes in the set of node i 's neighbors \mathcal{N}_i receive the broadcasted value $x_i(t)$, and update their state values according to the following equation:

$$x_k(t+1) = \gamma x_k(t) + (1 - \gamma)x_i(t), \forall k \in \mathcal{N}_i \quad (4.2)$$

with $\gamma \in (0, 1)$ denoting the *mixing parameter*.

- The remaining nodes in the network, including i , update their state values as

$$x_k(t+1) = x_k(t), \forall k \notin \mathcal{N}_i. \quad (4.3)$$

This procedure takes place at every clock tick.

Let $x(t)$ denote the vector of values at the end of the t -th ticking event. Then,

$$x(t+1) = W(t)x(t) \quad (4.4)$$

where the random matrix $W(t)$, with probability $1/N$ is (assuming that the i -th clock ticks)

$$W_{jk}^{(i)} = \begin{cases} 1 & j \notin \mathcal{N}_i, k = j \\ \gamma & j \in \mathcal{N}_i, k = j \\ 1 - \gamma & j \in \mathcal{N}_i, k = i \\ 0 & \text{elsewhere} \end{cases} \quad (4.5)$$

where $W^{(i)}$ denotes the weight matrix corresponding to the case where node i 's clock ticks.

The following lemma discusses two important properties of the weight matrices.

Lemma 16 *The weight matrices $\{W^{(i)} : i = 1, 2, \dots, N\}$ satisfy the following:*

- (i) $\mathbf{1}$ is a right eigenvector of all $W^{(i)}$, i.e., $W^{(i)}\mathbf{1} = \mathbf{1}, \forall i$.
- (ii) $\mathbf{1}$ is not a left eigenvector of any $W^{(i)}$, i.e., $\mathbf{1}^T W^{(i)} \neq \mathbf{1}^T, \forall i$.

Proof Let us consider the first claim. It suffices to show that all rows of all $W^{(i)}$ matrices sum to unity. We have

$$\sum_{k=1}^N W_{jk}^{(i)} = \mathbb{1}\{j \in \mathcal{N}_i\}(\gamma + (1 - \gamma)) + \mathbb{1}\{j \notin \mathcal{N}_i\}1 = 1, \quad (4.6)$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. Thus, the proof of first item is complete.

We now turn to the second claim. Note that

$$\sum_{j=1}^N W_{jk}^{(i)} = 1 + (1 - \gamma)|\mathcal{N}_i|, \text{ for } k = i. \quad (4.7)$$

Since R is chosen to make the graph connected, we have $|\mathcal{N}_i| \geq 1$, which implies that $\sum_{j=1}^N W_{jk}^{(i)}|_{k=i} > 1$. This in turn shows that for all i there exists at least one

column, namely the $k = i$ column, with sum different than one, which implies that $\mathbf{1}^T W^{(i)} \neq \mathbf{1}^T$ for all i .

The above Lemma reveals that $c\mathbf{1}$ for some $c \in \mathbb{R}$ is a fixed point of the broadcasting gossip algorithm, so $W^{(i)}c\mathbf{1} = c\mathbf{1}$ for all i . If the algorithm converges to a consensus, the network will not leave the consensus state. However, it also shows that the sum (and therefore the average) of the vector of node values is not preserved at each step.

Suppose node k is transmitting at time t . It is easy to check that the discrepancy between the sum at the next and current time-slots is nonzero whenever $x_k(t) \neq |\mathcal{N}_k|^{-1} \sum_{i \in \mathcal{N}_k} x_i(t)$. Suppose, that $x_k(t)$ is closer to the maximum of its neighbors than the minimum, or $|x_k(t) - \min_{i \in \mathcal{N}_k} x_i(t)| > |x_k(t) - \max_{i \in \mathcal{N}_k} x_i(t)|$. Then the sum difference between time-slots is bounded:

$$\left| \sum_{i=1}^N (x_i(t+1) - x_i(t)) \right| \leq (1 - \gamma) |\mathcal{N}_k| \left| x_k(t) - \min_{i \in \mathcal{N}_k} x_i(t) \right|. \quad (4.8)$$

Clearly, the difference between the states sum at consecutive iterations is small if the node state values are close to each other.

Let us denote the mean of i.i.d. $W(t)$ as $\mathbb{E}\{W(t)\} = W$. The following lemma gives some properties of the average weight matrix that would prove useful for the remainder of the chapter. Of note is that the following is a specific case of the general weight matrix given [38].

Lemma 17 *The average weight matrix W is given by*

$$W = I - \frac{1 - \gamma}{N} \text{diag}\{\Phi \mathbf{1}\} + \frac{1 - \gamma}{N} \Phi \quad (4.9)$$

and, for all γ , satisfies the following equation:

$$W\mathbf{1} = \mathbf{1}, \mathbf{1}^T W = \mathbf{1}^T, \rho(W - J) < 1, \quad (4.10)$$

where $\rho(\cdot)$ denotes the spectral radius of its argument and $J = (N)^{-1}\mathbf{1}\mathbf{1}^T$.

Proof See Appendix C.1.

The Lemma shows that, unlike the individual weight matrices, $\mathbf{1}$ is both a left and right eigenvector of the average weight matrix. Moreover, the spectral radius of the weight matrix is less than unity, a property that will prove to be useful throughout the rest of the chapter.

4.4 Convergence of Broadcast Gossiping

In this section, we will study the convergence of the asynchronous broadcast gossip algorithms considering a slightly more general setting, where the consensus algorithm is governed by a product of identically distributed random matrices with the only restriction that each matrix is stochastic but not doubly stochastic. Of note is that the almost sure convergence result shown in this section, *i.e.*, Theorem 2, is focused on and specific to broadcast gossip algorithms and is a special cases of the almost sure convergence result presented in [38].

4.4.1 Convergence in the Expectation

We consider the convergence in expectation of the broadcasting gossip algorithm. The next result reveals that, although the sum is not preserved per iteration, it is preserved in expectation. We consider the initial state as deterministic, and hence all expectations are averaging the mixing matrices only.

Proposition 1 *The limiting random vector obtained through broadcast gossip iterations (if it exists) is, in expectation, equal to the average of initial node measurements, i.e.,*

$$\mathbb{E} \left\{ \lim_{t \rightarrow \infty} x(t) \right\} = \frac{1}{N} \mathbf{1} \mathbf{1}^T x(0). \quad (4.11)$$

Proof By the Lebesgue dominated convergence theorem [23], we have

$$\mathbb{E} \left\{ \lim_{t \rightarrow \infty} x(t) \right\} = \lim_{t \rightarrow \infty} \mathbb{E} \{ x(t) \}. \quad (4.12)$$

Moreover, since the matrices $W(t)$ are i.i.d., we have

$$\lim_{t \rightarrow \infty} \mathbb{E} \{ x(t+1) \} = \lim_{t \rightarrow \infty} W^t x(0). \quad (4.13)$$

Thus it suffices to prove that $\lim_{t \rightarrow \infty} W^t = (N)^{-1} \mathbf{1} \mathbf{1}^T$. From [92], we know that this statement will hold if $W \mathbf{1} = \mathbf{1}$, $\mathbf{1}^T W = \mathbf{1}^T$, and $\rho(W - J) < 1$. Since Lemma 17 indicates that these conditions are satisfied, the proof is complete.

The proposition indicates that the expectation of the limiting random vector of the broadcasting gossip algorithm, given a certain initial state vector, is equal to the vector whose entries are equal to average of such initial state.

4.4.2 Convergence in the Second Moment

To study the convergence of the algorithm, we analyze the convergence of the vector $\beta(t)$ defined as the vector of deviations of the components of $x(t)$ from their average at the t th iteration. This can be expressed in component form as $\beta_i(t) = x_i(t) - \bar{x}(t)$, or as

$$\beta(t) = x(t) - Jx(t) = (I - J)x(t). \quad (4.14)$$

Let $\lambda_i(A)$ denote the i th largest eigenvalue of a matrix A . In the following, we present a sufficient condition guaranteeing the convergence of the expectation of the deviation vector norm to zero.

Lemma 18 *The expectation of the norm of the deviation vector of the broadcast gossip, i.e., $\mathbb{E}\{\|\beta(t)\|_2^2\}$, converges to zero if*

$$\lambda_1(\mathbb{E}\{W(t)^T(I - J)W(t)\}) < 1, \quad (4.15)$$

where I denotes the identity matrix.

Proof Utilizing the properties of $W(t)$ matrices, we find that the deviation vector $\beta(t)$ obeys the following recursion with probability one:

$$\beta(t + 1) = (W(t) - JW(t))\beta(t). \quad (4.16)$$

Note that this iteration is different from the one tracking the distance to initial node measurements average in gossip-based algorithms which preserve the sum, and this difference impacts all our proof methodologies.

Let $Y(t) = (W(t) - JW(t))$ so that $\beta(t + 1) = Y(t)\beta(t)$. Now, taking the expected norm of $\beta(t + 1)$ given $\beta(t)$ and using the fact that $\|u\|_2^2 = u^T u$ for $u \in \mathbb{R}^N$, yields

$$\mathbb{E}\{\|\beta(t + 1)\|_2^2 \mid \beta(t)\} = \beta(t)^T \mathbb{E}\{Y(t)^T Y(t)\} \beta(t) \quad (4.17)$$

$$\leq \lambda_1(\mathbb{E}\{Y(t)^T Y(t)\}) \cdot \|\beta(t)\|_2^2, \quad (4.18)$$

where the last line follows from the fact that all $Y(t)^T Y(t)$ matrices are symmetric and the Rayleigh–Ritz theorem [44]. Then, repeatedly conditioning and using the linear iteration obtained above, we have,

$$\mathbb{E}\{\|\beta(t)\|_2^2\} \leq \lambda_1^t(\mathbb{E}\{Y(t)^T Y(t)\}) \cdot \|\beta(0)\|_2^2. \quad (4.19)$$

Thus, now one can see that $\lim_{t \rightarrow \infty} \mathbb{E}\{\|\beta(t)\|_2^2\} = 0$ if $\lambda_1(\mathbb{E}\{Y(t)^T Y(t)\}) < 1$. Algebraic manipulations reduce this sufficient condition to the one stated in the Lemma.

It is important to emphasize that the Lemma 18 gives a sufficient condition for any consensus protocol that does not preserve network sum. Moreover, note that the condition $\lambda_1(\mathbb{E}\{W(t)^T(I - J)W(t)\}) < 1$ is different than the convergence condition obtained for the standard pairwise gossip algorithms where one only need to have $\lambda_2(\mathbb{E}\{W(t)\}) < 1$ to ensure the second-order convergence to the initial node measurements average [20, 34]. Note, however, that the sufficiency condition derived for the broadcast gossip algorithms reduces to the one for average-preserving gossip algorithms when $\mathbf{1}^T W^{(i)} = \mathbf{1}^T, \forall i$.

In the following, we show that the broadcasting gossip algorithm satisfies the sufficiency condition required achieve consensus in the second moment.

Proposition 2 *The broadcast gossip algorithms satisfies the fact that $\lambda_1(\mathbb{E}\{W(t)^T(I - J)W(t)\}) < 1$.*

Proof First, note that the eigenvalue of interest is the maximum eigenvalue of expectation over positive semidefinite matrices since $(W^{(i)})^T(I - J)W^{(i)} = ((I - J)W^{(i)})^T((I - J)W^{(i)})$. This indicates that $\lambda_1(\mathbb{E}\{W(t)^T(I - J)W(t)\}) \geq 0$. Moreover, let $W' = \mathbb{E}\{W(t)^T W(t)\}$ and $W'' = \mathbb{E}\{W(t)^T J W(t)\}$, and observe the following

$$\lambda_1(W' - W'') = \max_{\|u\|_2=1} u^T W' u - u^T W'' u \quad (4.20)$$

where the above follows from the variational definition of eigenvalues (Note that $W' - W''$ is a symmetric matrix). Recall that $\max_{\|u\|_2} u^T W' u = 1$ for $u =$

$u_1 = 1/\sqrt{N}\mathbf{1}$ which is the eigenvector corresponding to the unit eigenvalue. Of note is that for all $\{u : u \in \mathbb{R}^N, \|u\|_2^2 = 1, u \neq u_1\}$, we have $u^T W' u < 1$ which implies that $\lambda_1(\mathbb{E}\{W(t)^T(I - J)W(t)\}) < 1$ since $u^T W'' u \geq 0$ for all $u \in \mathbb{R}^N$ (Note that the expectation is taken over positive semidefinite matrices). Thus, the task reduces to show that for $u = u_1$, we still have $\lambda_1(W' - W'') < 1$. For $u = u_1$, equation (4.20) reduces to

$$u_1^T W' u_1 - u_1^T W'' u_1 = 1 - u_1^T W'' u_1 < 1 \quad (4.21)$$

where the last inequality follows from the fact that $u_1^T W'' u_1 > 0$ since all entries of the W'' matrix is nonnegative (note that the expectation is taken over nonnegative entry matrices). Thus, $u^T W' u - u^T W'' u < 1$ for all $\{u : u \in \mathbb{R}^N, \|u\|_2^2 = 1\}$, indicating that $\max_{\|u\|_2^2=1} u^T W' u - u^T W'' u < 1$, which, in turn, yields $\lambda_1(\mathbb{E}\{W(t)^T(I - J)W(t)\}) < 1$.

4.4.3 Almost Sure Convergence to Consensus

Given the results of Section 4.4.1 and 4.4.2, we are now in the position of stating our main result.

Theorem 2 *The broadcast gossip algorithm converges to a consensus almost surely.*

That is,

$$\Pr \left\{ \lim_{t \rightarrow \infty} x(t) = c\mathbf{1} \right\} = 1 \quad (4.22)$$

for some random variable $c \in \mathbb{R}$ where

$$\mathbb{E}\{c\} = \frac{1}{N}\mathbf{1}^T x(0). \quad (4.23)$$

Proof See Appendix C.2.

The theorem indicates that the broadcasting gossip algorithms achieve consensus with probability one, and the consensus value is, in expectation, equal to the desired value, *i.e.*, average of initial nodes measurements.

4.5 Optimal Mixing Parameter

The sufficient condition in Lemma 18 depends on $\lambda_1(\mathbb{E}\{W(t)^T(I - J)W(t)\})$, which is the rate of convergence of an upper bound for the mean square deviation. Minimizing this parameter is a meaningful criterion of optimality when trying to approach convergence rapidly. In this section, we derive the optimal mixing parameter defined in (4.2) by minimizing $\lambda_1(\mathbb{E}\{W(t)^T(I - J)W(t)\})$ with respect to the mixing parameter in the broadcast gossip algorithm. At the same time, this allows to study the effect of the graph connectivity and network size on the optimal mixing parameter. The following Lemma gives formulae which will be useful in the remaining analysis of the broadcast gossip algorithm.

Lemma 19 *The following two formulas hold:*

(i) Let $W' \triangleq \mathbb{E}\{W(t)^T W(t)\}$, then

$$W' = I - \frac{2\gamma(1 - \gamma)}{N} (D - \Phi). \quad (4.24)$$

(ii) Let $W'' \triangleq \mathbb{E}\{W(t)^T J W(t)\}$, then

$$W'' = \frac{(1 - \gamma)^2}{N^2} (D - \Phi)^2 + J \quad (4.25)$$

where $D = \text{diag}(\Phi \mathbf{1})$ is the diagonal matrix of node degrees.

Proof See Appendix C.3.

Note that $\lambda_1(\mathbb{E}\{W(t)^T(I - J)W(t)\}) = \lambda_1(W' - W'')$ and recall that $\lambda_1(W' - W'')$ gives the worst-case convergence characteristic of the broadcast gossip algorithms, Lemma 18. Now, consider the matrix $W' - W''$ which, after using Lemma 19, reduces to:

$$W' - W'' = I - J - \frac{2\gamma(1 - \gamma)}{N}(D - \Phi) - \frac{(1 - \gamma)^2}{N^2}(D - \Phi)^2. \quad (4.26)$$

First note that the vector $\mathbf{1}$ is an eigenvector of $W' - W''$ with eigenvalue 0. The vector $\mathbf{1}$ corresponds to the only nonzero eigenvalue of the matrix J and the only zero eigenvalue for the Laplacian matrix $L = D - \Phi$. Therefore the eigenvectors of $W' - W''$ are exactly the eigenvectors of $D - \Phi$, and the k -th eigenvalue of $W' - W''$ for $k = 1, 2, \dots, N - 1$ is:

$$\lambda_k(W' - W'') = 1 - \frac{2\gamma(1 - \gamma)}{N}\lambda_{N-k}(L) - \frac{(1 - \gamma)^2}{N^2}\lambda_{N-k}(L)^2 \quad (4.27)$$

Thus, the eigenvalue of interest can now be written as:

$$\lambda_1(W' - W'') = 1 - \frac{2\gamma(1 - \gamma)}{N}\lambda_{N-1}(L) - \frac{(1 - \gamma)^2}{N^2}\lambda_{N-1}(L)^2 \quad (4.28)$$

where $L = D - \Phi$ denotes the Laplacian matrix of the graph. Of note is that $\lambda_{N-1}(L)$ is referred as the *algebraic connectivity* of the graph [7].

In the following, we investigate the effect of the mixing parameter on the eigenvalue of interest which, as seen by Lemma 18, bounds the rate of convergence of the broadcast gossip algorithms.

Corollary 3 *Let us introduce $\lambda_1(W' - W''; \gamma)$ to show the dependency of the eigenvalue of interest to the mixing parameter γ . Then the following statements hold:*

- (i) $\lambda_1(W' - W''; \gamma)$ is convex in γ .

(ii) The optimal mixing parameter, minimizing a worst-case convergence rate, is given by

$$\gamma^* = \frac{N - \lambda_{N-1}(L)}{2N - \lambda_{N-1}(L)}. \quad (4.29)$$

Proof Let us first consider convexity of $\lambda_1(W' - W''; \gamma)$ of w.r.t. to γ . One can show that the first and second derivatives of $\lambda_1(W' - W''; \gamma)$ are given by

$$\frac{\partial \lambda_1(W' - W''; \gamma)}{\partial \gamma} = \gamma \left(\frac{4}{N} \lambda_{N-1}(L) - \frac{2}{N^2} \lambda_{N-1}(L^2) \right) - \frac{2}{N} \lambda_{N-1}(L) + \frac{2}{N^2} \lambda_{N-1}(L^2), \quad (4.30)$$

and

$$\frac{\partial^2 \lambda_1(W' - W''; \gamma)}{\partial \gamma^2} = \frac{4}{N} \lambda_{N-1}(L) - \frac{2}{N^2} \lambda_{N-1}(L^2), \quad (4.31)$$

respectively. Recall that $\lambda_1(W' - W''; \gamma)$ is convex in γ if $\partial^2 \lambda_1(W' - W''; \gamma) / \partial \gamma^2 \geq 0$ for $\gamma \in (0, 1)$. Moreover, the eigenvalues of the Laplacian are non-negative indicating that $\lambda_{N-1}(L) \geq 0$ and $\lambda_{N-1}(L^2) = \lambda_{N-1}^2(L)$. Then, from (4.31), it is easy to see that, $\partial^2 \lambda_{N-1}(L) / \partial \gamma^2 \geq 0$ if $0 \leq \lambda_{N-1}(L) \leq 2N$. But recall that $\lambda(L) \leq \min\{2 \max_i |\mathcal{N}_i|, N\}$.

Consider next the second claim of the corollary. The optimal γ is clearly given by

$$\gamma^* = \arg \min_{\gamma} \lambda_1(W' - W''; \gamma). \quad (4.32)$$

which, since $\lambda_1(W' - W''; \gamma)$ is convex, is simply found by setting (4.30) to zero and solving it for γ .

Interestingly, the above Corollary indicates that the optimal mixing parameter depends on the graph for finite N . For large N we have the following result, whose proof is trivial.

Corollary 4 For graphs such that $\lambda_{N-1}(L) = \Theta(f(N))$ for some function $f(\cdot)$, with $\lim_{N \rightarrow \infty} f(N)/N = 0$, the optimal mixing parameter is given by

$$\lim_{N \rightarrow \infty} \gamma^* = \frac{1}{2}. \quad (4.33)$$

Hence, for large enough N and standard radius connectivity considerations for random geometric graphs (e.g., $R = \Theta(\sqrt{\log N/N})$ and $\lambda_{N-1}(L) = \Theta(\log N)$), the eigenvalue $\lambda_1(W' - W'')$ increases as $|\gamma - 1/2|$ increases. Therefore the worst-case convergence rate, characterized by $\lambda_1(W' - W'')$, decreases. In words, γ values that are closer to $1/2$ yield a faster worst-case convergence rate compared to the γ values closer to its boundaries, i.e., zero and one.

In the following, we investigate the effect of the graph Laplacian on the optimal mixing parameter.

Corollary 5 Let us introduce $\gamma^*(L) \triangleq \gamma^*$ to denote the dependency of the optimal γ on the graph Laplacian. Then, $\gamma^*(L)$ is monotonically decreasing function of $\lambda_{N-1}(L)$.

Proof Proof simply follows by analyzing the first derivative of the optimal mixing parameter w.r.t. the parameter of interest, denoted as $\partial\gamma^*(L)/\partial\lambda_{N-1}(L)$, and showing that $\partial\gamma^*(L)/\partial\lambda_{N-1}(L) < 0$ for $\lambda_{N-1}(L) \leq \min\{2 \max_i |\mathcal{N}_i|, N\}$.

Thus, the above corollary indicates that, as the graph connectivity increases, i.e., the eigenvalues of the Laplacian increases, the optimal mixing parameter tends to zero. This result matches the intuition. In fact, in a fully connected graph clearly $\gamma = 0$ would result in a *consensus* at the first iteration.

4.6 Performance Analysis of Broadcast Gossip Algorithms

While broadcast gossip algorithms do not preserve the network sum, they do compute a linear combination of the network states. We can define as *error* the deviation of the states from the average of the initial states and use the mean square error as a metric to evaluate the algorithm performance. Even though the average displacement does not give the complete probabilistic picture but lends insight to the average MSE performance of the algorithm. Probabilistic concentration results on a general class of such random consensus algorithms can be found in [38] and some results reported here to make the chapter self-contained can be derived as special cases.

This section is dedicated to the derivation of the mean-square error performance of the broadcast gossip algorithm and to studying the mixing parameter effect on the mean-square error performance as well as the convergence. In particular, we prove an upper bound on the discrete time (or equivalently, number of clock ticks) required to get within ϵ of the consensus $c\mathbf{1}$, $c \in \mathbb{R}$. We also derive an upper bound on the limiting mean-square error performance. Finally, we examine the communication complexity of the broadcast gossip algorithms to achieve a certain distance to consensus.

4.6.1 Mean Square Error

Since, in general, the broadcast gossip algorithm does not converge to the initial node measurements average $(N)^{-1}\mathbf{1}^T x(0)$, it is of interest to consider the

distance of the consensus value to $\bar{x}(0)$. In the remaining, we use

$$\alpha(t) = x(t) - Jx(0). \quad (4.34)$$

to denote the difference between the state vector at time step t and the average of initial node measurements.

Lemma 20 *Let $\mathbb{E}\{\|\alpha(t)\|_2^2\}$ denote the mean square error at time step t . The following two statements hold:*

(i) *The mean square error iteration obeys a recursion given as:*

$$\mathbb{E}\{\|\alpha(t+1)\|_2^2\} \leq (1 - \lambda_2(W'))\mathbb{E}\{\|J\alpha(t)\|_2^2\} + \lambda_2(W')\mathbb{E}\{\|\alpha(t)\|_2^2\}. \quad (4.35)$$

(ii) *If $\nexists c \in \mathbb{R}$ such that $x(t) = c\mathbf{1}$ almost surely, then*

$$\mathbb{E}\{\|\alpha(t+1)\|_2^2 | \alpha(t)\} < \|\alpha(t)\|_2^2. \quad (4.36)$$

Proof See Appendix C.4.

The above Lemma reveals that the mean square error (MSE) conditioned on the current state is a *strictly decreasing* function of time and the strict inequality becomes equality when the nodes converge to consensus. In the following, we consider the limiting MSE behavior of the broadcast gossip algorithms.

Proposition 3 *The limiting MSE of the broadcast gossip algorithms is upper bounded by*

$$\lim_{t \rightarrow \infty} \mathbb{E}\{\|\alpha(t)\|_2^2\} \leq \|\alpha(0)\|_2^2 \left(1 - \frac{1 - \lambda_2(W')}{1 - \lambda_{N-1}^2(W - J)} \right). \quad (4.37)$$

Proof See Appendix C.5.

As in the worst-case convergence-rate case, it is of interest to characterize the effect of the mixing parameter γ on the limiting MSE performance. This is considered in the following Corollary.

Corollary 6 *Let $U_\infty(\gamma)$ be the upper-bound on the limiting MSE of the broadcast gossip iterations, given in Proposition 3, as a function of the mixing parameter γ . Then, the following statements hold:*

(i) *The boundary cases, i.e., $\gamma \rightarrow 0$ and $\gamma \rightarrow 1$, are given by*

$$\lim_{\gamma \rightarrow 0} U_\infty(\gamma) = \|\alpha(0)\|_2^2 \quad (4.38)$$

and

$$\lim_{\gamma \rightarrow 1} U_\infty(\gamma) = \|\alpha(0)\|_2^2 \left(1 - \frac{\lambda_{N-2}(L)}{\lambda_1(L)} \right), \quad (4.39)$$

respectively.

(ii) *$U_\infty(\gamma)$ is a monotonically decreasing function of γ .*

(iii) *$U_\infty(\gamma)$, for $\gamma = \gamma^*$, is given by*

$$U_\infty(\gamma^*) = \|\alpha(0)\|_2^2 \left(1 - C(L) \frac{\lambda_{N-2}(L)}{\lambda_1(L)} \right), \quad (4.40)$$

where

$$C(L) = \frac{2N - 2\lambda_{N-1}(L)}{4N - 2\lambda_{N-1}(L) - \lambda_1(L)}. \quad (4.41)$$

Proof See Appendix C.6.

The Corollary indicates that the limiting MSE performance of the broadcast gossip algorithm decreases when γ is increasing. This is due to the fact that as γ approaches zero, the broadcasting nodes create a local dominance shifting away from the desired mean a multitude of nodes, whereas, for γ values closer to unity, the nodes receiving the broadcasted value adjust their own state only slightly, thereby changing minimally the network mean.

4.6.2 Communication Cost to Achieve Consensus

Generalizing the analysis done for standard sum preserving gossip-based averaging algorithms, we define the ϵ -converging time in the following.

Definition 3 Given $\epsilon > 0$, the ϵ -converging time is the earliest time at which the vector $x(k)$ is ϵ close to the normalized initial deviation with probability greater than $1 - \epsilon$:

$$T(N, \epsilon) = \sup_{x(0)} \inf \left\{ t : \Pr \left\{ \frac{\|x(t) - Jx(t)\|_2}{\|x(0) - Jx(0)\|_2} \geq \epsilon \right\} \leq \epsilon \right\} \quad (4.42)$$

where $\|\cdot\|_2$ denotes the l_2 norm of its argument.

Before we move on to the main result of this section, we need the following Lemma giving the order of the eigenvalue of interest.

Lemma 21 For the broadcast gossip algorithm,

$$1 - O\left(\frac{\log^4 N}{N^2}\right) \leq \lambda_1(\mathbb{E}\{W^T(I - J)W\}) \leq 1 - \Omega\left(\frac{\sqrt{\log N}}{N^{5/2}}\right). \quad (4.43)$$

Proof See Appendix C.7.

Unfortunately, the upper and lower bounds do not coincide – they differ (ignoring logarithmic terms) by a \sqrt{N} factor. It may be possible to tighten the upper bound by exploiting the fact that for a communication radius slightly larger than the threshold the random geometric graph is regular in an order sense with degree $\Theta(\log N)$ [16]. However, we do not pursue this here.

Given the convergence rate definition, we have the following rate of convergence to a consensus for the broadcast based average consensus.

Proposition 4 *The ϵ -converging time of the asynchronous broadcast gossip algorithms is bounded by*

$$\Pr \left\{ \frac{\|x(t) - Jx(t)\|_2}{\|x(0) - Jx(0)\|_2} \geq \epsilon \right\} \leq \epsilon \quad (4.44)$$

where

$$\Omega \left(\frac{N^2 \log \epsilon^{-1}}{\log^4 N} \right) = T(N, \epsilon) = O \left(\frac{N^{5/2} \log \epsilon^{-1}}{\sqrt{\log N}} \right). \quad (4.45)$$

Proof At this stage of development, we just have to put the pieces together. Given the Definition 3 and the results of [20], using the Markov inequality and noting that:

$$\frac{0.5 \log \epsilon^{-1}}{\log \lambda_1^{-1}(\mathbb{E}\{W(t)^T(I - J)W(t)\})} \leq T(N, \epsilon) \leq \frac{3 \log \epsilon^{-1}}{\log \lambda_1^{-1}(\mathbb{E}\{W(t)^T(I - J)W(t)\})}. \quad (4.46)$$

Now, we have upper and lower bounds on λ_1 of the form $1 - \alpha$, so

$$\Omega \left(\frac{\log \epsilon^{-1}}{1/2(1 - \lambda_1)} \right) = T(N, \epsilon) = O \left(\frac{\log \epsilon^{-1}}{1 - \lambda_1} \right). \quad (4.47)$$

Substituting the bounds in Lemma 21 yields the result.

Moreover, note that if we set $\epsilon = 1/N^\alpha$ in the above equation, then we obtain $T(N, 1/N^\alpha) = \Omega(N^2/\log^3 N)$. Since the number of transmissions per iteration is one in the broadcast gossip algorithms, this result matches also the communication complexity¹. One can observe that broadcast gossip algorithms improve upon randomized gossip algorithms ($\Theta(N^2 \log N)$), but appears to be worse than the geographic gossip which has communication complexity in the order of $O(N^{3/2}\sqrt{\log N})$. As we will see very shortly through numerical examples, broadcast gossip significantly outperforms both algorithms for practical

¹Note that larger connectivity radius implies, at the expense of larger broadcasting power, smaller λ_1 ($\mathbb{E}\{W^T(I - J)W\}$) and, in turn, better convergence rate.

network sizes. This is interesting, because it illustrates how the constraint of maintaining the sum of the states constant does come with some performance penalty as well.

4.6.3 Performance Analysis: Numerical Examples

In the following, as in [33], we compare the number of *radio transmissions* to achieve a certain distance from consensus of broadcast gossiping. We choose $\gamma = 1/2$, since this value is the optimal value in terms of convergence speed and provides a trade-off for the MSE. To simulate the random geometric graph, we consider nodes that are uniformly distributed over a unit square. Their initial values are initialized as uniformly distributed random values with unit variance and zero mean. The connectivity radius is chosen as $R = \sqrt{\log(N)/N}$. Of note is that each iteration requires one, two and *the number of hops many* radio transmissions, respectively, for broadcast, standard and geographic gossiping.

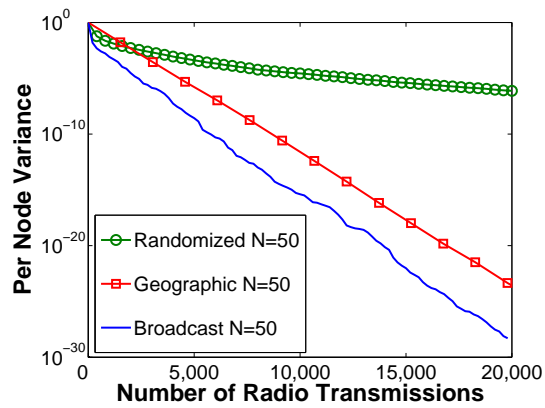
The plots show the standard gossip algorithm [20], geographic gossip algorithm [33], and the broadcast gossip algorithm. Figure 4.1 (a–c) depict per-node variance versus the number of radio transmissions for different network sizes (each data point is an ensemble average of 25 trials). Recall that the transmissions per iterations of randomized, geographic and broadcast gossip algorithms are, respectively equal to two, to the number of hops and to one. The simulation results suggest that broadcast gossiping reaches consensus faster than both competing protocols for an equal communication cost (cost that does not account for the extra complexity of routing two ways in geographic gossiping protocol). Indeed, this comparison is not entirely fair since the methods in [20]

and [33] do meet an extra constraint; the comparison is useful since it highlights the non negligible penalty in speed that results from the extra constraint of maintaining the sum of states constant.

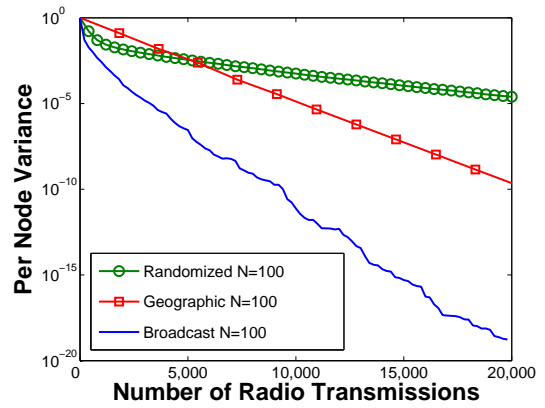
In Fig. 4.2 (a–c), we initialize the node values with zero except one in a single node following [33]. This is a field where computing the average over the network is a harder task than the one considered before as in the previous case all the nodes are somewhat closer to the average. However, in this case, the information of the node containing the spike value needs to diffuse over the whole network. The network simulation setup is the same as above. The plot again illustrate the faster diffusion of consensus in broadcast gossiping compared to geographic and randomized gossiping protocols.

Next, we consider the MSE performance of the broadcast gossip algorithm versus the number of iterations and compare them to those of randomized and geographic gossip algorithms. Recall that the MSE of randomized and geographic gossip algorithms is zero in the limit, whereas the MSE of the broadcast algorithm saturates to a non-zero value as the algorithm converges to a consensus. The random geometric graph is simulated exactly as specified for the previous comparisons.

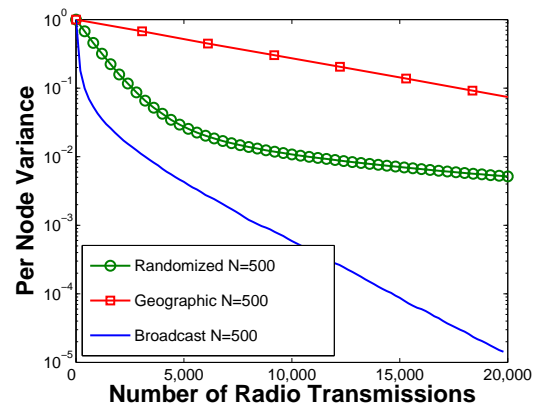
Figure 4.3 depicts the MSE performance of the randomized, geographic and broadcast gossip algorithms versus the number of *radio transmissions* (two, number of hops many and one per iteration for randomized, geographic and broadcast gossip) for $N = \{500\}$, respectively. An interesting observation is that, for a reasonable number of radio transmissions, the MSE performance of the broadcast gossip algorithm is better than the randomized and geographic algorithms. However, as the number of radio transmissions increase, the random-



(a)

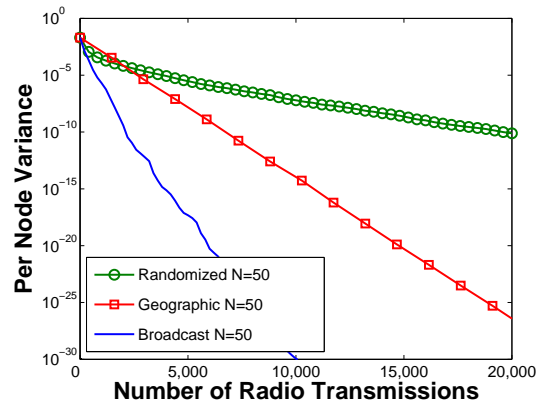


(b)

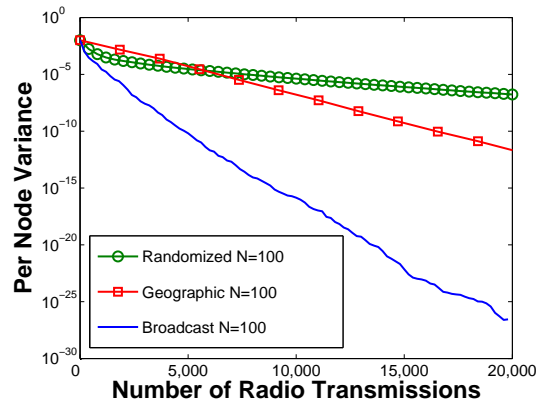


(c)

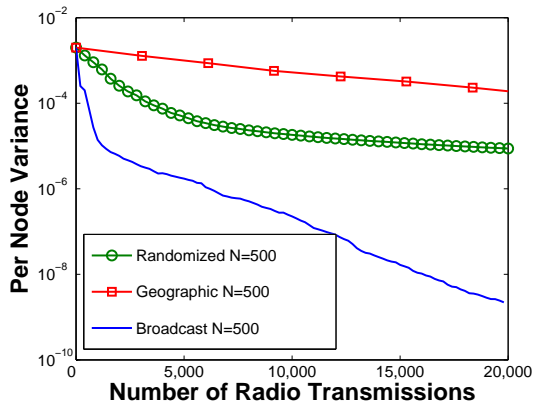
Figure 4.1: Number of radio transmissions required to achieve a given distance (per node variance) from the consensus for $N \in \{50, 100, 500\}$ with initial node values uniformly distributed.



(a)



(b)



(c)

Figure 4.2: Number of radio transmissions required to achieve a given distance (per node variance) from the consensus for $N \in \{50, 100, 500\}$ with initial node values zero except one in a single node.

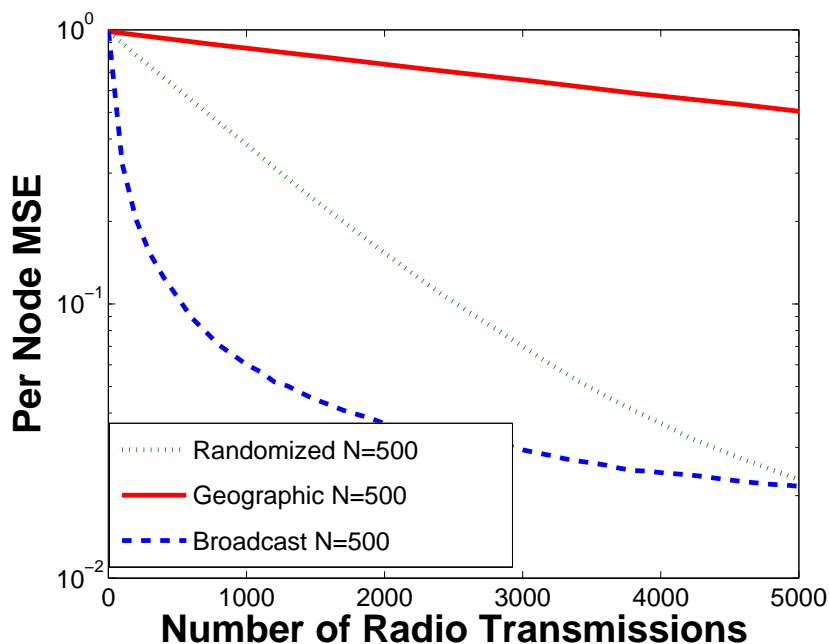


Figure 4.3: The MSE performance of the randomized, geographic and broadcast gossip algorithms with respect to the number of radio transmissions for $N = 500$ with initial node values uniformly distributed.

ized and geographic gossip, outperforms the broadcasting one, as they tend to zero whereas the performance of the broadcast gossip saturates to a non-zero value. Of note is that, the cross-over point where the randomized and geographic gossip starts to outperform the broadcast gossip increases with increasing number of nodes in the network.

These simulation results corroborate the theoretical analysis since they indicate that the MSE strictly decreases as long as consensus is not achieved. They also show that the faster convergence of broadcast gossip can be exploited to approximate quickly the desired average. In applications that are severely constrained in communication cost broadcast consensus may represent a good practical alternative to the competing methods that preserve the sum of the states.

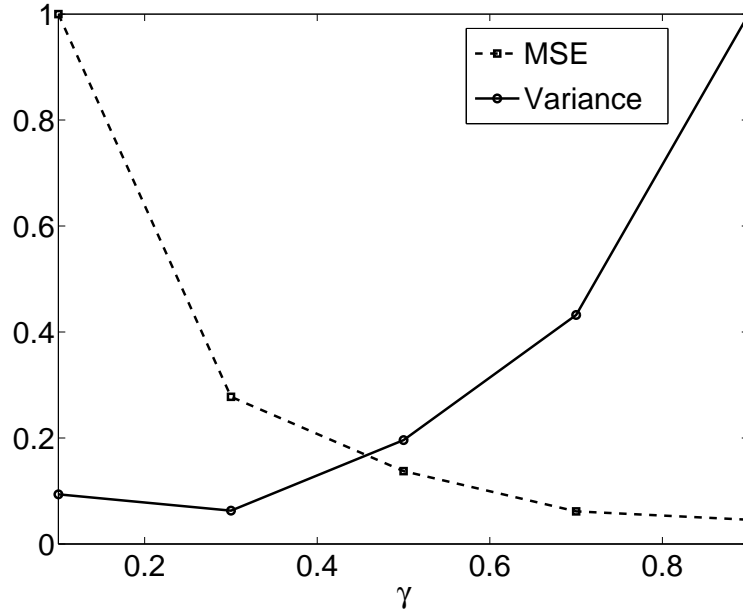


Figure 4.4: The MSE and Variance performances of the broadcast gossip algorithms with respect to γ for $N = 500$ with initial node values uniformly distributed.

In Figure 4.4, we simulate the MSE and the speed of convergence performances of the algorithm with respect to the parameter γ for $N = 500$. Of note is that the simulation results are normalized by the largest corresponding output. Parallel to our theoretical findings in Corollary 6, the MSE of the algorithm monotonically decreases with increasing γ . Moreover, as Corollary 3 suggests, we observe that per-node variance is convex in γ , *i.e.*, the mixing parameter². On the other hand, the theory suggests that the optimal mixing parameter γ (although smaller then) is in the neighborhood of 0.5 where the simulation results indicate that the optimal γ is around 0.3. We note that the theoretical results correspond to the optimization of an upper bound, therefore the value of the practical optimal γ may differ from the theoretical one presented here.

²It is of interest to note that similar trade-off between the convergence rate and MSE performance (larger (smaller) convergence rate yields a larger (smaller) MSE) is also observed in the agreement problems with transmission noise case [51].

4.7 Discussions

In this chapter we used the inherent broadcast nature of the wireless medium to present a simple “one-way” protocol that has good performance in simulations of networks with a modest number of nodes. The protocol simplifies the implementation of random gossiping compared to methods that require a pairwise node exchange. We presented the conditions on the weights matrices that would guarantee convergence to a consensus, and showed that the broadcast gossip algorithm achieves consensus with probability one. Moreover, the random consensus value is, in expectation, equal to the average of the initial node states. Although the network sum is not preserved at each broadcasting time-slot, we provided theoretical and simulation results on the mean square error performance in approximating such average. Finally, we presented theoretical and numerical examples evaluating and comparing the communication cost of gossiping algorithms required to achieve a given distance to consensus.

Even though the broadcast gossip algorithm shows promise in terms of convergence rate and MSE performance, the appropriate distributed averaging algorithm, *e.g.*, randomized, geographic or broadcast, with the appropriate tuning parameters, depends ultimately on the application at hand which determines the relative importance of the implementation simplicity, convergence rate, MSE performance, or cap on the number of radio transmissions. Networks operating under adverse conditions may be prone to packet losses, node failures, and other events that may render pairwise exchange protocols and long-haul routing infeasible. Our analysis and method becomes particularly useful in these situations.

CHAPTER 5
COMPUTING ALONG THE ROUTES WITH GOSSIPING

5.1 Motivation and Related Work

A fundamental problem in networks is the transportation and distribution of information from one part of the network to another [2, 92, 58, 59]. We focus on a special network computation problem, where a group of destination nodes is interested in a function that can be decomposed as a sum of functions of local variables stored by another set of source-nodes. We refer to this problem as the Computing Along Routes (CAR) problem. This particular problem has a wide range of applications including aggregation queries, distributed detection, content distribution, decentralized traffic monitoring, distributed control and coordination [107, 65, 76, 92].

We focus on the case where the set of source nodes and the set of destination nodes are *disjoint*, and hence our case does not include average consensus gossiping [92] as a special case. This particular assumption has strong practical appeal since, in many cases, the nodes which are responsible for collecting the data and the nodes that are designated to process these data are disjoint and geographically separated. For instance, in the problem of distributed detection, only the fusion centers are interested in the outcome of the sensor decisions. In the case of content distribution, the entity who is interested in the data does not necessarily have access to any part of it before the distribution occurs. In the case of the leader-follower coordination problem [49], several nodes are to follow a group of leaders, and the source and destination sets are clearly disjoint.

One solution to the problem we posed is to unicast between each pair of source-destination nodes via the shortest path joining source and destination[32]. After receiving all the information, each destination node can compute the desired function independently. Except in very special cases, this strategy is inefficient since the exact same information flows in the network several times, and it is unreliable since it is severely affected by link failures. The second approach is multicasting the information, having a single source node transmitting at a time to all destinations, thereby allowing the computation of the sought results independently [29]. Thanks to network coding [2],[35],[57], multicasting can be done using the links efficiently. However, this decomposition of the problem is agnostic to the fact that the nodes do not want the data themselves, but an aggregate result.

In fact, the problem considered in this chapter is closely related to the data aggregation and routing problems studied in the computer science literature, where spatially distributed data is to be collected by a fusion center, utilizing data aggregation and in-network processing techniques [107, 37, 70, 25, 4]. The similarity is obvious if one observes that solutions to the so called *duplicate-sensitive* data aggregation problem (see e.g. [70]) can be generalized to find codes that solve our problem. Duplicate-sensitive data aggregation refers to the case where destinations seek a single copy of data from each source. In the case of a single source and multiple destinations, the duplicate-sensitive aggregation problem has been studied extensively, proposing energy efficient schemes based on spanning trees[64, 96], and algorithms that are robust to link/node failures [70]. In the case of multiple source and multiple destinations, solutions have been proposed including hierarchical structures and minimizing routes costs [4, 45, 25]. These methodologies are not viable for solving the CAR prob-

lem, because one would need to explicitly consider duplicating the data of all sources at the destinations. Moreover, it falls short of explaining the relationship between type of topologies and the feasible queries and it does not incorporate feedback.

Observing the fact that the problem has two parts, *i.e.*, the *computation* of the function and the *routing* of the information required, we are proposing a joint strategy where the computation of the desired function is achieved along the routes via gossiping. Our approach incorporates feedback since gossiping based protocols are iterative, thus nodes continuously exchange their values and create a feedback effect. We focus on separable functions, *i.e.*, function whose synopsis can be written as $\sum_i f_i(x_i)$, where the index i is over the set of nodes. Without loss of generality, we will consider the scenarios where the function of interest is simply the average, since any other separable function can be calculated by initializing the source nodes' values with states $f_i(x_i)$.

As said above (see also Section 5.6), there are algorithms that result in partially directed solutions for the CAR problem. However, one of the main interests of this chapter is exploring the value of feedback.

Feedback is the trademark of *Average Consensus Gossiping* (ACG) protocols, which are considered attractive in wireless sensor applications because the communications among nodes are limited to their immediate neighbors, they are easily mapped into stationary asynchronous policies, while the network topology can be dynamically switching. In ACG protocols each node in the network is both a source and a destination [92],[8],[100],[17]. An ACG protocol can directly be applied to our problem: in fact, non-source nodes could set their initial values to zero, and via ACG the network would converge to a fraction of the

average of the source nodes. When the destination nodes know the number of source nodes and also the network size, the true average can be calculated by rescaling the ACG result. However, this approach has two main drawbacks: 1) Since the desired information has to be distributed to the whole network, the convergence will be slower, 2) Every node will have access to the average at the end of the algorithm, and this is a major weakness since it does not preserve secrecy.

We note that a setup resembling more to our problem is considered in [55], where the authors have applied the gossiping algorithm to solve a sensor localization problem, assuming that each node in the network wants to compute a linear combination of the anchor (source) nodes, to determine their exact locations. Unfortunately, this analysis can not be utilized to solve our problem since all of the non-source nodes are destinations, and destination nodes are not necessarily interested in the same function of the source nodes. A significant work is also due to Mosk-Aoyama et.al. who have considered distributed computation of separable functions as well as information dissemination on arbitrary networks [68, 69]. Moreover, Benezit et.al. have studied average consensus problem via randomized path averaging to achieve increased convergence rates [17]. However, our problem differs from these models in the sense that source and destination sets are disjoint, and there may exist intermediate nodes which are neither sources nor destinations.

In summary, we are proposing a gossiping based algorithm for jointly routing and calculating the desired function at the destinations. The innovative aspect of this procedure is that it incorporates feedback, unlike similar problems that consider a unidirectional flow of data. On the other hand, it does not nec-

essarily distribute the desired value to the whole network which makes our scheme more secure and flexible.

5.1.1 Chapter Organization

We formulate our problem in Section 5.2. In Section 5.3, we introduce necessary and sufficient conditions for the existence of solutions to the proposed problem. In Section 5.4.1, we investigate spectral properties of feasible codes and show what classes of code structures lead to feasible solutions. In Section 5.4.2, we introduce reductions of the network topology which can be employed to simplify the design problem without loss of generality. By focusing on stochastic codes, we provide necessary conditions on the topology for the feasibility and discuss some infeasible cases in Section 5.5. We introduce a formulation for so called *partially directed* solution in Section 5.6, and discuss the complexity and communication cost of the algorithm in Section 5.7. We compare the performance of our solution with the existing solutions in Section 5.8, and provide a simple extension to the asynchronous update scheme in Section 5.9. Finally, we conclude our chapter in Section 5.10.

5.2 Problem Formulation

In this chapter, we consider a connected network (N, E) with N nodes and the corresponding edge set E which consists of ordered node pairs (i, j) . Given the edge (i, j) , i is the tail and j is the head of the edge. We define the neighbor set of node i as $\mathcal{N}_i \triangleq \{j \in \{1, 2, \dots, N\} : (j, i) \in E\}$.

We consider the following problem setup: Each node in the network has an initial scalar measurement denoted by $x_i(0) \in \mathbb{R}$ where $i \in \{1, \dots, N\}$. Let $\mathcal{S} \triangleq \{1, 2, \dots, N\}$. There is a set of nodes (*destination nodes*), denoted as $\mathcal{S}_D \subseteq \mathcal{S}$, which are interested in the average of a set of nodes (*source nodes*), denoted as $\mathcal{S}_S \subseteq \mathcal{S}$. We note that $\mathcal{S}_S \cap \mathcal{S}_D = \emptyset$, *i.e.*, a source node cannot be a destination node. We restrict our attention to a synchronous gossiping protocol, with constant update weights:

$$x_i(t+1) = W_{ii}x_i(t) + \sum_{j \in \mathcal{N}_i} W_{ij}x_j(t), \quad i \in \mathcal{S}, \quad (5.1)$$

where t is the discrete time index, W_{ij} is the link weight corresponding to the edge (j, i) . We note that if $j \notin \mathcal{N}_i$, then $W_{ij} = 0$. At each discrete time instant, each node updates its value as a linear combination of its own value and its neighbors values. This type of analysis is usually the first step to investigate more flexible asynchronous solutions [20]. If we define an $N \times N$ matrix W such that $[W]_{ij} = W_{ij}$ and $x(t) = [x_1(t), x_2(t), \dots, x_N(t)]^T$, then (5.1) can be written in the matrix form as:

$$x(t+1) = Wx(t) = W^{t+1}x(0). \quad (5.2)$$

Parallel to the network coding literature, we refer to the matrix W as the *code* that we are interested in designing. The equation above implies that:

$$\lim_{t \rightarrow \infty} x(t) = \lim_{t \rightarrow \infty} W^t x(0) = W^\infty x(0)$$

where $W^\infty \triangleq \lim_{t \rightarrow \infty} W^t$ (assuming that $\lim_{t \rightarrow \infty} W^t$ exists). One way to solve our problem is assigning zeros as initial state values for non-source nodes, and running an average consensus algorithm on the network. The consensus value will be a rescaled version of the source nodes' average; thus, to determine the desired function, the destination nodes have to multiply the consensus value by a

scaling factor. For this approach, any code W that solves the average consensus can be employed [88, 20, 100, 11]. This approach has two major disadvantages: First, since the whole network has to converge to a consensus, the converge will be slow which will require more resources, *i.e.*, energy, time (c.f. Section 5.8). Second, at the end of the algorithm, every node will have access to the average, and this is a major weakness in the case of secret communications.

The approach which we pursue here is to design a W that produces the desired computation irrespective of what $x(0)$ is, and distributes the value to the destinations only. Since, we are only interested in calculating the average of the nodes values in \mathcal{S}_S at all $j \in \mathcal{S}_D$, we have the following inherent constraints on W^∞ :

$$W_{jk}^\infty = \begin{cases} |\mathcal{S}_S|^{-1} & k \in \mathcal{S}_S \\ 0 & k \notin \mathcal{S}_S \end{cases}, \forall j \in \mathcal{S}_D. \quad (5.3)$$

In other words, the entries of the limiting weight matrix should be chosen such that destination nodes multiply source nodes' values by $|\mathcal{S}_S|^{-1}$ (for averaging), and multiply non-source nodes' values by 0. Therefore, the rows of W^∞ corresponding to destination nodes should have $|\mathcal{S}_S|^{-1}$ for the source node columns, and 0 otherwise.

Let us denote the structure above, imposed by sets \mathcal{S}_D and \mathcal{S}_S , as $F(\mathcal{S}_D, \mathcal{S}_S)$, and as $F(E)$, the structure imposed by the network connectivity, *i.e.*, $W_{ij} = 0$ if $(i, j) \notin E$. We denote a code W , which satisfies (5.3), as an *Average Value Transfer* (AVT) solution to the problem of computing along routes (CAR). The following definitions are in order:

Definition 4 *A code W is a feasible AVT solution for CAR on a given network (N, E) and source-destination sets $(\mathcal{S}_D, \mathcal{S}_S)$, if and only if $W \in F(E)$ and $W^\infty \in F(\mathcal{S}_D, \mathcal{S}_S)$*

with $W^\infty < \infty$.

Definition 5 *CAR is AVT infeasible, if there does not exist any AVT solution.*

We note that $<$ denotes elementwise strict inequality. In the rest of the chapter, unless stated otherwise, we will omit the word *AVT* and refer to Definition 4 as the solution and 5 as the feasibility condition. Therefore, our (in)feasibility definition considers only AVT solutions.

In the rest of the chapter, we limit ourselves to codes in the set of *nonnegative matrices*, i.e., $W \geq 0$ and \geq represents elementwise inequality, in analogy with the ACG policies [20]. In the following section, we will introduce necessary and sufficient conditions for a W matrix to be an AVT solution for the CAR problem.

5.3 Necessary and Sufficient Conditions on Feasible Codes

In the following, we first give necessary conditions on feasible W and W^∞ in addition to the structure given in (5.3). We will then use the resulting equations to introduce necessary and sufficient feasibility conditions.

We first partition N sensors in the network into three *disjoint* classes: M sensors that belong to the *source nodes* set, K sensors that belong to the *destination nodes* set, and L sensors that belong to neither source nodes nor destination nodes (called *intermediate nodes*). In other words, $|\mathcal{S}_S| = M$, $|\mathcal{S}_D| = K$ and $L = N - M - K$ where $|\cdot|$ denotes the cardinality of the set which is its argument. Without loss of generality, we index the set of source nodes as $\{1, \dots, M\}$, the set of destination nodes as $\{M + 1, \dots, M + K\}$, and the set intermediate nodes

as $\{M + K + 1, \dots, N\}$. At this point, we can partition the $N \times N$ matrix W as:

$$W = \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & P \end{bmatrix} \quad (5.4)$$

such that $A \in \mathbb{R}^{M \times M}$, $E \in \mathbb{R}^{K \times K}$ and $P \in \mathbb{R}^{L \times L}$. Assuming that $\lim_{t \rightarrow \infty} W^t$ exists, we denote this limit as W^∞ and its partitions using the superscript $'$, *i.e.*, A' , D' , G' , etc. Since the set \mathcal{S}_D is only interested in the average of the set \mathcal{S}_S , then $B' = 1/|\mathcal{S}_S| \mathbf{1} \mathbf{1}^T$ and $E' = H' = 0$ by (5.3). We note that $\mathbf{1}$ is the all ones vector of the appropriate dimensions. This inherent structure results in the following necessary constraints on W and W' :

Lemma 22 *Given a network $F(E)$ and the sets \mathcal{S}_S and \mathcal{S}_D , if $W \geq 0$ is a feasible solution to CAR, then:*

- (1) $D' = G' = 0$,
- (2) $D = G = 0$,
- (3) $\mathbf{1}^T A = \mathbf{1}^T$.

Proof The proof of the lemma is given in Appendix D.1.

Lemma 22 shows that the information flow between the sources and the rest of the network *must* be one-way, *i.e.*, from the sources to the network. Such a finding is not surprising since the network is only interested in the average of the source nodes, and the average will be biased if the sources mix their state values with values from the non-source nodes, whose initial states are arbitrary. Lemma 22 also shows that the row sums of the matrix A are all equal to 1. Since

the matrix A governs the communication among the source nodes, the sum of the source values must remain constant through the iterations. This result is also intuitive since, otherwise, the average of the source nodes would change and W would not be a solution for the problem. The fact that the flow from the source nodes to the rest of the network is directional, has important consequences on what designs are feasible, as we will see next.

In light of Lemma 22, we will focus on the network codes which have the structure:

$$W = \begin{bmatrix} A & 0 & 0 \\ B & E & H \\ C & G & P \end{bmatrix}.$$

At this point, for mathematical brevity, we repartition the W matrix in four super-blocks recycling the previous symbols as follows:

$$W = \begin{bmatrix} A & 0 \\ B & D \end{bmatrix} \text{ where } B = \begin{bmatrix} B \\ C \end{bmatrix}, D = \begin{bmatrix} E & H \\ G & P \end{bmatrix}. \quad (5.5)$$

Of note is that, the new $B \in \mathbb{R}^{N-M \times M}$ and $D \in \mathbb{R}^{N-M \times N-M}$. Similarly, we can partition the state vector $x(t)$:

$$s(t) = [x_1(t), \dots, x_M(t)]^T, \quad r(t) = [x_{M+1}(t), \dots, x_N(t)]^T.$$

We note that $s(t)$ represents the evolution of the source states while $r(t)$ represents the behavior of the rest of the network including the destination nodes. Similarly, the matrix A governs the communication among the source nodes, and the matrix D determines the communication structure among the rest of nodes. B instead governs the flow from the sources to the rest of the network.

We expand the general form of the update in (5.1) using (5.5) as:

$$\begin{aligned} s(t+1) &= As(t), \\ r(t+1) &= Bs(t) + Dr(t). \end{aligned}$$

The equations above can be rewritten in terms of the initial conditions as:

$$\begin{aligned} s(t+1) &= A^{t+1}s(0), \\ r(t+1) &= D^{t+1}r(0) + \sum_{l=0}^t D^{t-l}BA^l s(0). \end{aligned}$$

The linear system of equations in a compact form is:

$$x(t+1) = \begin{bmatrix} s(t+1) \\ r(t+1) \end{bmatrix} = W^{t+1}x(0) = \begin{bmatrix} A^{t+1} & 0 \\ \sum_{l=0}^t D^{t-l}BA^l & D^{t+1} \end{bmatrix} \begin{bmatrix} s(0) \\ r(0) \end{bmatrix}. \quad (5.6)$$

At this point, we can state the main theorem of the chapter which introduces the necessary and sufficient conditions on the feasible network codes $W \geq 0$:

Theorem 3 *Given a network $F(E)$ and the sets \mathcal{S}_S and \mathcal{S}_D , a $W \geq 0$ matrix of the form (5.5) is an AVT solution to the CAR problem if and only if:*

(1) W is in the form of:

$$W = \begin{bmatrix} A & 0 \\ B & D \end{bmatrix}, \quad (5.7)$$

where $A \in \mathbb{R}^{M \times M}$, $D \in \mathbb{R}^{N-M \times N-M}$ and $M = |\mathcal{S}_S|$,

(2) $\lim_{t \rightarrow \infty} W^t$ exists and is finite,

(3) $\lim_{t \rightarrow \infty} [D^{t+1}]_{1:K} = 0$,

(4) $\lim_{t \rightarrow \infty} [\sum_{l=0}^t D^{t-l}BA^l]_{1:K} = \frac{1}{M} \mathbf{1}\mathbf{1}^T$,

where $[\cdot]_{1:K}$ denotes the first K rows of its argument, and $K = |\mathcal{S}_D|$.

Proof While sufficiency follows from (5.6) and the problem definition given in Section 5.2, these conditions are necessary because of Lemma 22 and equation (5.6).

In the following, we show that one can simplify the fourth constraint in Theorem 3 by exchanging the powers of A with its limit, *i.e.*, the evolution of the source states does not affect the feasibility of the solution.

Lemma 23 *Assuming that conditions (1) – (3) in Theorem 3 hold, then condition (4) holds if and only if*

$$\lim_{t \rightarrow \infty} \left[\sum_{l=0}^t D^{t-l} B A^\infty \right]_{1:K} = \frac{1}{M} \mathbf{1} \mathbf{1}^T, \quad (5.8)$$

where $A^\infty = \lim_{t \rightarrow \infty} A^t$.

Proof The proof is given in Appendix D.2.

Lemma 23 shows that if there exists a limiting matrix A^∞ which satisfies the constraint in (5.8), then any A matrix such that $\lim_{t \rightarrow \infty} A^t = A^\infty$, satisfies the fourth constraint in Theorem 3. Therefore, given the underlying connectivity, one can assume that A has already converged to its limit A^∞ , and then seek a solution for partitions B and D . Unfortunately, even under this simplification, designing a feasible W is difficult, since the fourth constraint in Theorem 3 is non-convex with respect to the elements of the partitions D , B and A^∞ , and also consists of all non-negative powers of D . Moreover, it is not clear from Theorem 3 what kind of topologies do or do not have solutions for the CAR problem we posed.

In the following sections, we will both try to simplify the design of an AVT code and determine topologies where the CAR problem is known to be (in)feasible.

5.4 Spectral Analysis and Topology Reductions

5.4.1 Spectral Analysis of Feasible Codes

We start our discussion by introducing some necessary definitions. Motivated by the theory of Markov Chains, we say that node i has *access* to node j , if for some integer t , $W_{ij}^t > 0$. Two nodes i and j , which have access to the other, are said to be *communicating*. Since communication is an equivalence relation, the set of nodes which communicate forms a *class*. A class which consists of only source nodes is called a *source class*. In the following, we also introduce the graph theoretic definition of a source class.

Definition 6 *A set of source nodes forms a source class if and only if the sub-graph consisting of these source nodes only is irreducible, and including any other source nodes results in a reducible sub-graph.*

We note that two definitions are equivalent. At this point, we are ready to introduce our first condition on the rank of W^∞ .

Lemma 24 *Given a feasible W which satisfies Theorem 3, the following holds true:*

$$\# \text{ of source classes} \leq \text{rank}(W^\infty) \leq \# \text{ of classes.}$$

Proof The proof is given in Appendix D.3.

Let's denote $\text{rank}(W^\infty)$ as r_{W^∞} and \mathcal{U} as the set of linearly independent columns of W^∞ , implying that $|\mathcal{U}| = r_{W^\infty}$. If we denote elements of \mathcal{U} as u_i , $1 \leq i \leq r_{W^\infty}$, then W, W^∞ satisfy:

$$Wu_i = W^\infty u_i = u_i, \quad \forall i \in \{1, \dots, r_{W^\infty}\}. \quad (5.9)$$

In other words, the linearly independent columns of the limiting matrix are the dominant eigenvectors of W matrix, *i.e.*, the eigenvectors corresponding to the eigenvalue one. Therefore, understanding the column structure of the limiting matrix is crucial to determine the spectral properties of a feasible W and vice versa.

We first note that each source class SC will be represented by a single dominant eigenvector u_i in \mathcal{U} , which is equal to $[W^\infty]_i \quad \forall i \in SC$, where $[\cdot]_i$ denotes the i -th column of its argument. We note that W_{ki}^∞ is the weight of node i at node k , in the limit. For a given source node i , if $W_{ki}^\infty > 0$, then node k has access to the weighted value of node i . It is clear that if k is a destination node, then $W_{ki}^\infty = 1/M$, where M is the number of source nodes. The argument follows from Theorem 3, since W is feasible.

Let's assume that r_{W^∞} is strictly greater than the number of source classes. Therefore, there exists at least one class C which is not a source class and $\rho(W_C) = 1$, where W_C is the sub-matrix of rows and columns of W that correspond to the set C . If this is the case, the columns of W^∞ corresponding to the elements of the class must have some non-zero values. Thus, there exists at least one $i \in C$, where $W_{ki}^\infty > 0 \quad \forall k \in C$. In other words, some of the nodes in the network will have access to the weighted values of the nodes in class C in the limit. However, these nodes cannot be utilized in calculating the average of the source nodes at the destinations. In other words, if there exists a node

k which has access to a non-source class C in the limit, the destination nodes cannot have access to the node k in limit, since otherwise the third condition in Theorem 3 would not hold. We will summarize result of the discussion above by the following remark:

Remark 12 *For a feasible code W and a non-source class C , if $\rho(W_C) = 1$, then removing class C from the network does not change the feasibility of the AVT algorithm.*

Another way to interpret this condition is the following: If an AVT solution exists for a given scenario, then there exists at least one feasible solution under the extra constraint that there is no non-source class whose corresponding sub-matrix has spectral radius one. In other words, one can pose the extra constraint that all of the non-source classes have sub-matrices with spectral radii strictly less than 1, and such constraint does not change the feasibility of the problem. We also note that since sub-matrices corresponding to the non-source classes are also sub-matrices of the partition D of W (since D governs communications among non-source nodes), the eigenvalues of non-source classes are also the eigenvalues of D . Therefore, the extra constraint that we posed on non-source classes is equivalent to $\rho(D) < 1$. At this point, we would like to discuss the physical significance of $\rho(D) < 1$. We first note that $\rho(D) < 1$ implies $\lim_{t \rightarrow \infty} D^t = 0$. Moreover, as we have mentioned above, the partition D governs the communication among non-source nodes. Therefore, in this particular case, non-source nodes will have zero information about themselves in the limit. Such a result is not surprising since none of these nodes are interested in the values of non-source nodes. On the contrary, a subset of these nodes (destination nodes) are interested in the average of the source nodes, while the intermediate nodes do not aim at receiving anything at all. We summarize our

result in the following lemma:

Lemma 25 *There exists a feasible code W which satisfies Theorem 3 if and only if there exists a W^* which satisfies both Theorem 3 and $\rho(D) < 1$.*

While adding one more constraint to Theorem 3 may seem like increasing the complexity of an already difficult problem, we observe the following: A feasible W has to satisfy r_{W^∞} linear equations of the form (5.9). Moreover, $\rho(D) < 1$ implies that the rank of W^∞ will be equal to its lower bound given in Lemma 24. This, in return, implicitly shows that a feasible code W has to satisfy fewer equality constraints.

Moreover, since $\rho(D) < 1$, $\lim_{t \rightarrow \infty} D^t = 0$, thus the third condition in Theorem 3 is automatically satisfied. The condition in Lemma 23 becomes:

$$\left[\lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^\infty \right]_{1:K} = [[I - D]^{-1} B A^\infty]_{1:K} = \frac{1}{M} \mathbf{1} \mathbf{1}^T,$$

where I is the identity matrix with appropriate dimensions. Therefore, the fourth condition in Theorem 3 is also simplified.

At this point, we will introduce our last condition on the number of source classes in the network:

Lemma 26 *Consider a CAR problem with the sets $\mathcal{S}_S, \mathcal{S}_D$, a network $F(E)$, and a code W with $\rho(D) < 1$. If W is a feasible solution, then the following must hold:*

$$\# \text{ of source classes} \leq 1 + N - K - \text{rank}(B).$$

Proof The proof is given in Appendix D.4.

We note that K is the number of destination nodes and the matrix partition B governs the communication between source nodes and the rest of the network.

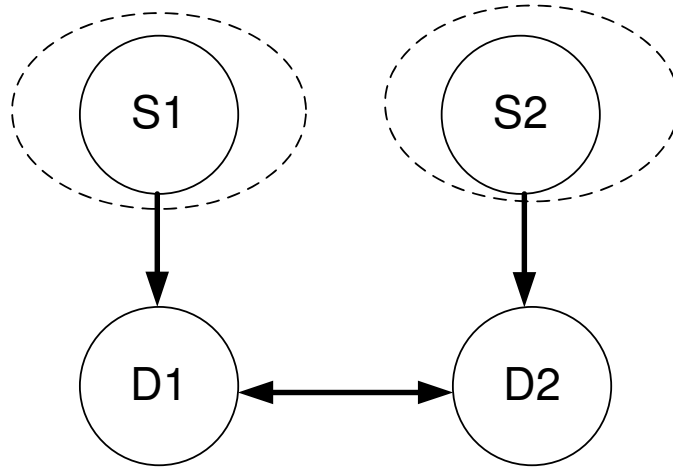


Figure 5.1: An infeasible scenario with two sources and two destinations.

Lemma 26 shows that if a network code W is feasible, then the number of source classes and the rank of the partition B have to balance out each other, *i.e.*, their sum has to be less than or equal to $1 + N - K$. Therefore, network topologies which do not satisfy Lemma 26 will be infeasible. In the following, we will illustrate our point with an example. Fig. 5.1 shows a network with two sources and two destinations. Each source node forms a source class by itself, therefore there are two source classes in the network. By Lemma 26, the partition B of W has to satisfy: $\text{rank}(B) \leq 1$. But, the rank of B can be set to one if and only if one of the two links going from the sources to the destinations is removed. This is not possible since one of the sources will be unconnected to the destinations. Therefore, there does not exist any feasible AVT code for this network.

In the following sections, we will propose several claims under the assumption that $\rho(D) < 1$. In light of Lemma 25, such statements do not change the generality of the claims, since our assumptions do not change feasibility of AVT codes for a given problem.

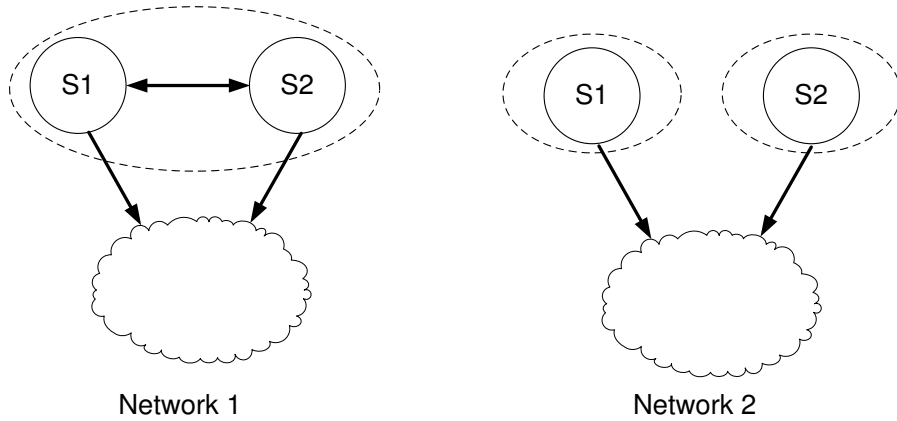


Figure 5.2: Two networks corresponding to the same underlying connectivity but two different network codes, W_1 and W_2 .

5.4.2 Reduction of Topology to Study Feasibility

In this section, we will introduce methods to perform topology reduction, while leaving the feasibility of AVT codes unchanged, by utilizing our discussions in Sections 5.3 and 5.4.1. In return, we will simplify our design problem and develop tools with which we can determine, by inspection, the topologies for which AVT codes are (in)feasible.

In Definition 6, we have given the definition of source class for a given code W . However, for a given network $F(E)$ and source set \mathcal{S}_S , source classes are not unique, *i.e.*, one may change the structure of the source classes by changing the entries of W . An example is given in Fig. 5.2. Both networks have the same underlying connectivity $F(E)$. The first network is induced by a W which assigns non-zero weights to the links between source nodes. Thus, two source nodes form a single source class. On the other hand, the second network corresponds to a W which assigns zero weights to the links between source nodes. Since these nodes cannot communicate, each forms a source class by itself.

In Section 5.4.1, we have also argued that the rank of W^∞ is closely related with the complexity of the problem and, under $\rho(D) < 1$, the rank is equal to the number of source classes in the network. Then, one can also argue that for CAR, one should cluster source nodes as much as possible, thus reduce the complexity of the problem. The definition of the minimal source class follows:

Definition 7 *For a given network and source set, the minimal source class is such that the total number of source classes is minimized.*

This is the case where all of the existing links between source nodes are assigned non-zero weights. The number of source classes in the minimal class can be obtained by removing all but the source nodes from the network $F(E)$, and counting the irreducible sub-graphs that remain.

Next, we will prove that focusing on the minimal class does not change the feasibility of the problem, hence we will conclude that, combined with the design advantages given above, one can focus on minimal source class codes.

Lemma 27 *Consider a problem $F(\mathcal{S}_S, \mathcal{S}_D)$ and a network $F(E)$. If the problem is feasible, then there exists at least one solution W which utilizes the minimum source class.*

Proof The proof is given in Appendix D.5.

The key point of the proof is that activating links among sources without changing an existing solution does not affect the feasibility of AVT codes. From Lemma 27, the first topology reduction is in order:

Reduction 1 *Use the minimal source class and utilize average consensus algorithm within each class.*

We note that utilizing average consensus algorithm within each class does not affect the feasibility of our problem. Such an observation follows from the fact that for a given feasible W , source node i must have equal information about the source nodes in its class, *i.e.*, $W^\infty_{ij} = \alpha_i, \forall j \in \mathcal{S}_{C^*}$, where \mathcal{S}_{C^*} is the class to which node i belongs. This observation is due to Lemma 22 and since each source class converges to a rank one matrix (c.f. Appendix D.3). Therefore one can construct a feasible W^* where each class utilizes an average consensus algorithm, by employing an ACG algorithm among the sources and rescaling some of the entries of the partition B of W matrix, keeping the partition D unchanged.

As a result of this reduction, each node in a given source class converges to the average of the initial source nodes in that particular class.

Reduction 2 *Treat each source class as a single node whose value is equal to the average of the source nodes in that particular class. Connect all non-source nodes, which are adjacent to the source nodes in the class, to this single node.*

We note that the second reduction does not change the feasibility of the AVT code due to Lemma 23.

Reductions 1-2 combined with $\rho(D) < 1$ simplify our design problem significantly, with respect to the conditions given in Theorem 3, without changing the feasibility of the problem.

5.5 A Necessary Condition On the Topology

In this section, we focus on AVT codes which satisfies $W\mathbf{1} = \mathbf{1}$ as well as Theorem 3. Since such condition combined with non-negativity, implies that W is

a stochastic matrix, we will refer these codes as *stochastic* codes. In return, we will provide a non-algebraic necessary condition for the existence of a feasible solution, which is easy to interpret. At the end of the section, we will discuss topologies which are known to have no feasible solutions.

Before introducing the main result of the section, we would like to emphasize the close relationship between stochastic AVT codes and non-homogenous random walks on graphs with absorbing states. A stochastic AVT code W represents the transition matrix of a Markov Chain \mathcal{M} whose state space consists of the nodes in the network (N, E) . The structure of the chain (locations of possible non-zero transition probabilities) is defined by the underlying connectivity $F(E)$, i.e., W_{ij} is the probability of jumping from state i to state j in a single step. Moreover, due to the structure of W in (5.7), each source class forms an absorbing class by itself. For a given non-source node i and a source class \mathcal{S}_{C^*} , the quantity $\sum_{j \in \mathcal{S}_{C^*}} W_{ij}^\infty$ will be equal to the probability that the chain is absorbed by the source class \mathcal{S}_{C^*} given the fact that \mathcal{M} has been initialized at node i [5]. Moreover, for a given source node $i \in \mathcal{S}_{C^*}$, the quantity $W_{ij}^\infty / \sum_{j \in \mathcal{S}_{C^*}} W_{ij}^\infty$ is the frequency that the chain visits node i given the fact that \mathcal{M} has been initialized at node j and it has been absorbed by the source class \mathcal{S}_{C^*} [5]. Given the discussion above, we conclude the following:

Remark 13 *Constructing a stochastic AVT code is equivalent to designing a transition probability matrix W for a Markov chain $\{\mathcal{M}(t)\}_{t=0}^\infty$ on graph (N, E) with state space \mathcal{S} , where each source class forms an absorbing class. Moreover, for each destination node $j \in \mathcal{S}_D$ and each source class \mathcal{S}_{C^*} , absorption probabilities should be chosen such that:*

$$P(\mathcal{M}(\infty) \in \mathcal{S}_{C^*} | \mathcal{M}(0) = j) = \frac{|\mathcal{S}_{C^*}|}{M},$$

where $P(\cdot)$ the probability of its argument. Moreover, for each source node $k \in \mathcal{S}_{C^*}$,

$$P(\mathcal{M}(\infty) \in k | (\mathcal{M}(\infty) \in \mathcal{S}_{C^*} | \mathcal{M}(0) = j)) = \frac{1}{|\mathcal{S}_{C^*}|},$$

where $\mathcal{M} \in k$ denotes the event that \mathcal{M} is in the state k in the limit, i.e, $t \rightarrow \infty$.

Unfortunately, the formulation given above does not simplify our design problem, since constructing transition probability matrices for complex chains with given stationary distributions is an open problem in the literature. However, we believe that pointing out the equivalence relation is necessary as it brings a different perspective to the AVT problem. Moreover, we will make use of the equivalence in the proof of the following lemma:

Lemma 28 *Consider a network $F(E)$ and the sets \mathcal{S}_S and \mathcal{S}_D . Partition the network into two disjoint sets (P, P^c) such that there exists at least one source class-destination pair in both sides of the network. For a given partition, we denote the links going from one set to the other set as cut edges, i.e, an edge (i, j) is a cut edge if $i \in P$ and $j \in P^c$, or $j \in P$ and $i \in P^c$. If there exists a feasible stochastic code, then there exists at least two (cut) edges between P and P^c for all such partitions.*

Proof Proof of the lemma is given in Appendix D.6.

Unlike the results we have proposed in the previous sections, Lemma 28 gives us a topology based method to detect infeasible cases. We note that the condition given in the lemma has to be satisfied by a network where stochastic AVT codes are feasible, besides the connectivity constraint that we imposed. Therefore, one can conclude that the connectivity assumption is not sufficient for the existence of a solution under stochastic codes. At this point, we remind our

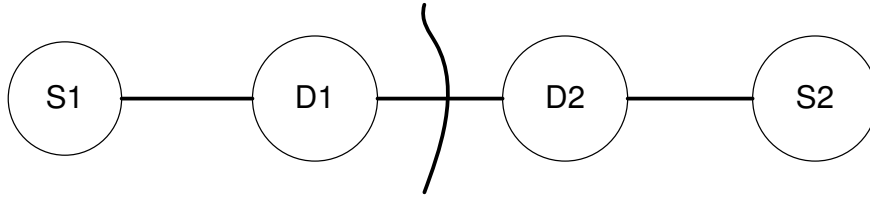


Figure 5.3: Line network topology. S and D denotes source nodes and destinations nodes respectively.

readers that the connectivity assumption is a sufficient condition for the existence of an ACG solution [92]. Thus, we conclude that demands on stochastic AVT codes are stricter than the ones for the consensus problems.

We would like to note that, as stated in the hypothesis, the conditions in Lemma 28 is valid for the cases where there are at least two source classes in the network. For the scenarios where there is only one source class, Lemma 28 is not valid.

In the following, we will give examples of infeasible network topologies for CAR under stochastic AVT codes. We will be considering nontrivial CAR instances with more than one source and destination nodes.

- (1) **Line network topology:** A line network topology is given in Fig. 5.3. The vertical line shows a cut where both partitions have one source-destination pair. By Lemma 28, such a network does not have a solution for CAR, since the cut shown in the figure has a single cut edge.
- (2) **Mesh network topology with a bottleneck link:** The network is shown in Fig. 5.4. It does not have a solution, since there is a single cut edge for the given partition.

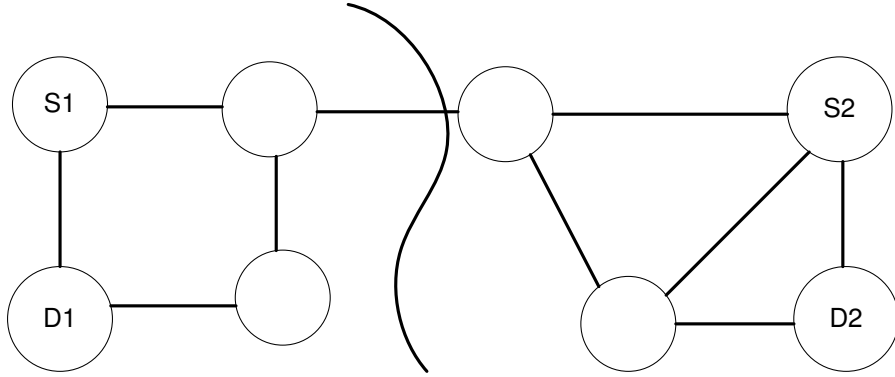


Figure 5.4: Mesh network topology. S and D denotes source nodes and destinations nodes respectively.

5.6 Construction of partially directed AVT solutions

As we have discussed in the preceding sections, it is difficult to construct feasible AVT codes due to the non-linearity of the constraints in Theorem 3. For this reason, we have proposed several simplifications, which do not affect the feasibility of an AVT code in Sections 5.4.1 and 5.4.2. While these simplifications are valuable in determining whether AVT codes are feasible or infeasible, they did not lead to a constructive way for designing AVT solutions.

In the following, we will formulate an integer programming problem whose solution will be utilized to construct so called *partially directed* AVT solutions. These solutions belong to a subset of the AVT solutions given in Theorem 3. We first introduce the definition of a partially directed AVT solution:

Definition 8 Consider a network $F(E)$ and the sets \mathcal{S}_S and \mathcal{S}_D . A code W is a *partially directed AVT solution*, if it satisfies Theorem 3 and each link on the network, except the links among the source nodes, can be utilized only in one direction, i.e., $W_{ij}W_{ji} = 0, \forall i, j \notin \mathcal{S}_S$.

It should be clear from Definition 8 why these codes are called partially directed solutions, *i.e.*, communication is directed only among the non-source nodes. Before proposing our construction, we will introduce some necessary definitions.

Since the communications among source nodes are bidirectional, we can employ Reduction 2 in Section 5.4.2 and assume that each source class has already converged to the average value of its members. We enumerate the source classes (arbitrarily ordered) and we define the set of source classes as \mathcal{S}_C . We also define $E' \subset E$ which contains all edges except the edges among the source nodes. For any given $U \subset N$, we define a set of edges $E(U) = \{(i, j) \in E' \mid i, j \in U\}$. In other words, $E(U)$ is the set of edges whose end points belong to the set U .

In the following lemma, we introduce our method to construct partially directed AVT codes:

Lemma 29 Consider a network $F(E)$ and the sets \mathcal{S}_S and \mathcal{S}_D . For $k \in \mathcal{S}_C$ and $l \in \mathcal{S}_D$, we define a variable b_i^{kl} as:

$$b_i^{kl} = \begin{cases} 1, & \text{if } i = k, \\ -1, & \text{if } i = l, \\ 0, & \text{otherwise.} \end{cases}$$

Consider the following integer programming formulation:

$$\text{mimize } \max_{k \in \mathcal{S}_C, l \in \mathcal{S}_D} \sum_{(i,j) \in E'} z_{ij}^{kl}, \quad (5.10)$$

$$\text{subject to } \sum_{j \mid (i,j) \in E'} z_{ij}^{kl} - \sum_{j \mid (j,i) \in E'} z_{ji}^{kl} = b_i^{kl} \quad i \in \{\mathcal{S}_C \cup \{M+1, \dots, N\}\}, k \in \mathcal{S}_C, l \in \mathcal{S}_D, \quad (5.11)$$

$$z_{ij}^{kl} \leq u_{ij}, \quad (i, j) \in E', k \in \mathcal{S}_C, l \in \mathcal{S}_D, \quad (5.12)$$

$$u_{ij} + u_{ji} \leq 1, \quad (i, j) \in E', \quad (5.13)$$

$$\sum_{(i,j) \in E(U)} u_{ij} \leq |U| - 1, \quad U \subset N, U \neq \emptyset, \quad (5.14)$$

$$z_{ij}^{kl} \in \{0, 1\}, \quad (i, j) \in E', \quad k \in \mathcal{S}_C, \quad l \in \mathcal{S}_D. \quad (5.15)$$

If the integer program given above has a solution, namely, z_{ij}^{*kl} , $(i, j) \in E'$, $k \in \mathcal{S}_C$, $l \in \mathcal{S}_D$, we define y_{ij}^k as follows:

$$y_{ij}^k = \begin{cases} 1, & \text{if } \sum_{l \in \mathcal{S}_D} z_{ij}^{*kl} \geq 1, \\ 0, & \text{otherwise.} \end{cases}$$

Then, a feasible partially directed AVT code can be constructed as follows:

$$W_{ji} = \begin{cases} \frac{\sum_{k \in \mathcal{S}_C} |S_C(k)| y_{ij}^k}{\sum_{l \in \mathcal{N}_j} \sum_{k \in \mathcal{S}_C} |S_C(k)| y_{lj}^k}, & \text{if } \sum_{l \in \mathcal{N}_j} \sum_{k \in \mathcal{S}_C} |S_C(k)| y_{lj}^k \neq 0 \text{ and } (i, j) \in E', \\ \frac{1}{|\mathcal{N}_j|+1}, & \text{if } i, j \in \mathcal{S}_S \text{ and } i \in \mathcal{N}_j, \text{ or } j \in \mathcal{S}_S \text{ and } i = j, \\ 0, & \text{otherwise,} \end{cases} \quad (5.16)$$

where \mathcal{S}_{C_k} is the set of nodes which belongs to the k -th source class.

Proof The proof is given in Appendix D.7.

The integer programming formulation given in the lemma is a directed multi-commodity flow problem with acyclicity constraint [18]. In particular, one can map the variable b_i^{kl} to the net inflow at node i of data with origin k and destination l . The value of the net inflow is positive at the sources, negative at the destinations, and zero otherwise. The variable z_{ij}^{kl} indicates the amount of information with origin k and destination l that flows through link (i, j) . y_{ij}^k is equal to one if there exists at least one flow on (i, j) that is originated from source class k .

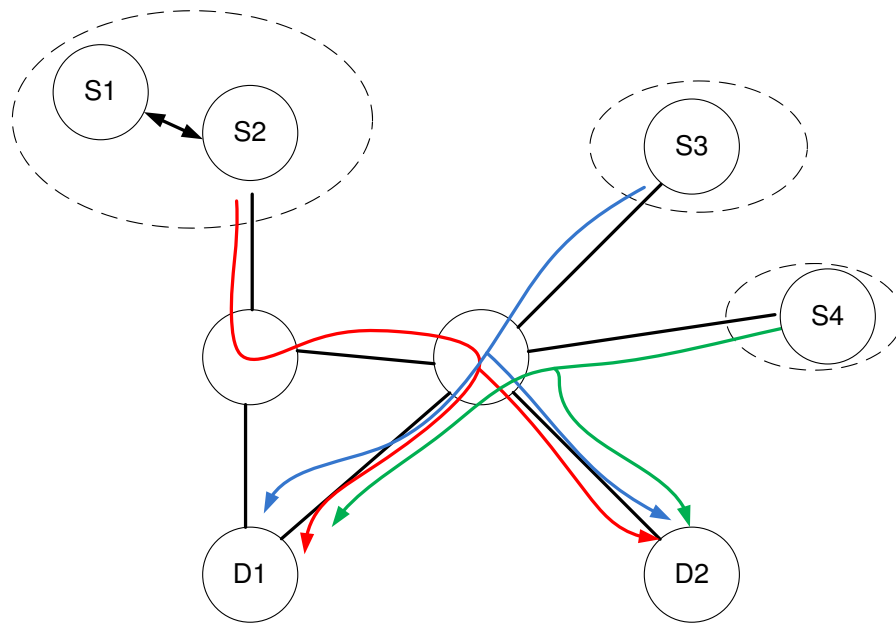
We note that the constraint given in (5.11) guarantees the flow is preserved at each node, *i.e.*, the total inflow to a given node is equal to the total outflow from

this node. Moreover, u_{ij} is a binary variable and equal to one if there exists at least one flow which utilizes the link (i, j) (5.12). Otherwise, it is equal to zero. Hence, (5.13) is the one-way flow constraint. Finally, (5.14) guarantees the flows are acyclic. The objective function of the problem is the maximum of number of paths between all source-destination pairs, thus the problem is minimizing the convergence time of the directed part of the algorithm.

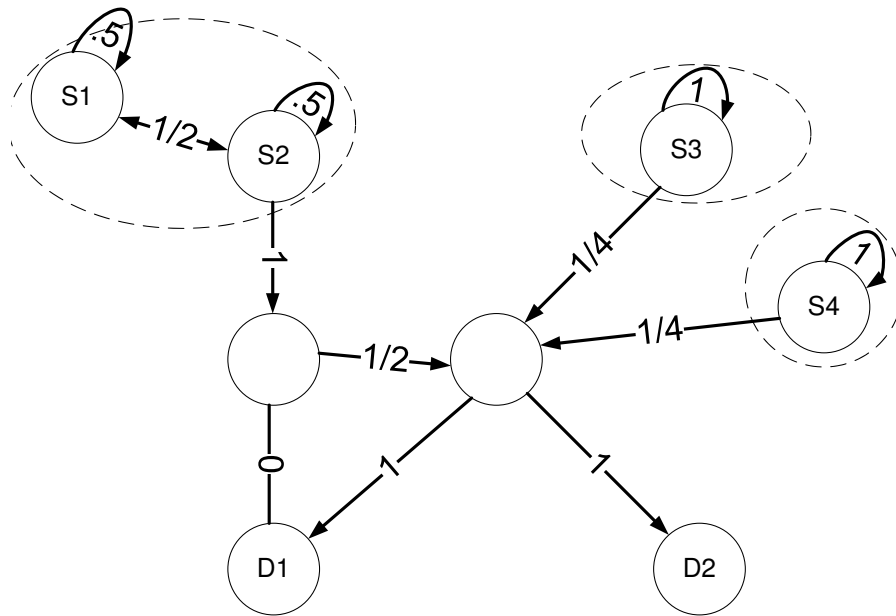
An example is given in Fig. 5.5. There are 4 source nodes with 3 source classes and 2 destination nodes. In Fig. 5.5(a), red, blue and green arrows represent flows from source classes to the destination nodes. The corresponding link weights are shown in Fig. 5.5(b). We note that the weight of the link connecting the flows from nodes 1 and 2 to the central hub is twice as much as the link weights of the other flows. This is because the number of source nodes in that particular flow is twice as much as the size of nodes in other flows.

We would like to conclude the section with the following observation: Existence of a partially directed AVT solution is a sufficient condition for the existence of an AVT solution, since partially directed AVT solutions also satisfy Theorem 3. On the other hand, the reverse argument may not always be true, *i.e.*, existence of an AVT solution does not imply the existence of a partially directed AVT solution. Remarkably, we were not able to find a counter example; we conjecture that the condition is both necessary and sufficient.

We also note that directed solutions are less robust to link/node failures while undirected solutions are more robust due to the presence of feedback. For instance, if a link on the path between a source-destination pair fails in a partially directed solution, the destination node will receive no information about that particular source. In the case of undirected solution, instead destination



(a) Optimal flows.



(b) Corresponding link weights.

Figure 5.5: A directed solution. *S* and *D* represents source and destination nodes respectively.

nodes may not converge to the desired average, but still have partial information about the source.

5.7 Complexity and Communication cost of AVT codes

In this section, we will first briefly discuss the complexity of constructing AVT codes. We, then, analyze the communication cost (number of message exchanges) of AVT codes on random geometric graphs.

5.7.1 Complexity

The problem formulation given in Theorem 3 is non-convex, thus closed form solutions do not exist in general. On the other hand, existing numerical methods for systems of nonlinear equations can be utilized to determine feasible solutions for a given problem [31]. We note that, while these methods perform fairly well in practice, convergence to a true solution is not guaranteed (because of possible singular Jacobians through the iterations, a wrong initialization point, etc). On the other hand, the partially directed AVT solution is an integer programming formulation, thus is guaranteed to converge to a feasible point (if such point exists). However, the formulation is *NP* complete in most cases [18], thus convergence to a feasible solution can be very slow. Since both numerical methods for nonlinear systems and the multicommodity formulation are fundamental questions in their own domains, they are out of the scope of this study.

5.7.2 Communication cost

In the case of undirected AVT codes, convergence rates and communication costs for the CAR problem are difficult to characterize. For this reason, we focus on the partially directed codes and derive corresponding communication cost. We remind that the partial AVT solutions has two time scales in terms of convergence: 1) The time it takes for each source class converge to the average, 2) The *finite time* that it takes for the directed flow from sources to destination to converge.

A 2-D geometric random graph $G^2(N, r)$ consists of N nodes which are uniformly distributed on a unit torus, and two nodes i and j are said to be connected when $d(i, j) \leq r$, where $d(., .)$ denotes the Euclidean distance of its arguments. 2-D random geometric graphs are of particular interest since they have been widely used as simplified models for wireless network topologies and focusing on these graphs makes it possible to compare our algorithm with regular average consensus algorithms in terms of communication cost [42, 19].

For $r = w^1(\sqrt{\log(N)/N})$, it has been shown that the graph is connected with high probability²(w.h.p.)[42], thus we focus on this regime. Moreover, in that particular regime, the underlying graph is regular with w.h.p., *i.e.*, each node has $\Theta(Nr^2)$ neighbors[19, 17]. Therefore, the diameter of the underlying network, *i.e.*, the longest shortest path, can be bounded as $\Theta(r^{-2})$.

We will assume that each source class utilizes the average consensus protocol with constant edge weights[92]. For a given random graph with N nodes and

¹ $w(.)$ denotes asymptotic domination, $\Theta(.)$ denotes asymptotic lower and upper bound and, $O(.)$ denotes asymptotic upper bound, where the asymptotics are with respect to N .

²with probability at least $1 - 1/N^2$

radius $r = w(\sqrt{\log(N)/N})$, the communication cost of the average consensus protocol is $\Theta(N r^{-2})$ [19, 92]. In that particular setting, the communication cost is NT_c , where T_c is the convergence time of the algorithm and it is defined as:

$$T_c = - \left(\log \sup_{x(0) \neq \bar{x}} \lim_{t \rightarrow \infty} \left(\frac{\|x(t) - \bar{x}\|_2}{\|x(0) - \bar{x}\|_2} \right) \right)^{-1},$$

where $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i(0)$. In words, the convergence time T_c gives the (asymptotic) number of steps for the error to decrease by the factor $1/e$, and NT_c is the total number of message exchanges required for that to happen.

Similarly, for a given CAR problem with source set \mathcal{S}_S , and corresponding source classes $\mathcal{SC} = \{\mathcal{S}_{SC_1}, \mathcal{S}_{SC_2}, \dots, \mathcal{S}_{|\mathcal{SC}|}\}$, communication complexity for all classes to converge to their own averages (by $1/e$) is equal to:

$$C = \Theta \left(\sum_{i=1}^{|\mathcal{SC}|} |\mathcal{S}_{SC_i}| r^{-2} \right) = \Theta (M r^{-2}),$$

where each class runs the average consensus algorithm independently. Moreover, once the source classes converge to the average, the algorithm needs, $O(r^{-2})$ more steps for each class to distribute the source information to each destination. We note that $\Theta(r^{-2})$ is equal to the diameter of the underlying network. Therefore, the algorithm needs, $O(|\mathcal{SC}|K r^{-2})$ message exchanges to distribute the information to the destination. We note that as $|\mathcal{SC}|$ and K increases, the paths between source-destination pairs will intersect more and more, thus the bound will become looser.

The communication cost of the algorithm is equal to:

$$C_{AVT} = \Theta(M r^{-2}) + O(|\mathcal{SC}|K r^{-2}) = O([M + |\mathcal{SC}|K] r^{-2}) \quad (5.17)$$

Please note that, in our case, the convergence time is defined as:

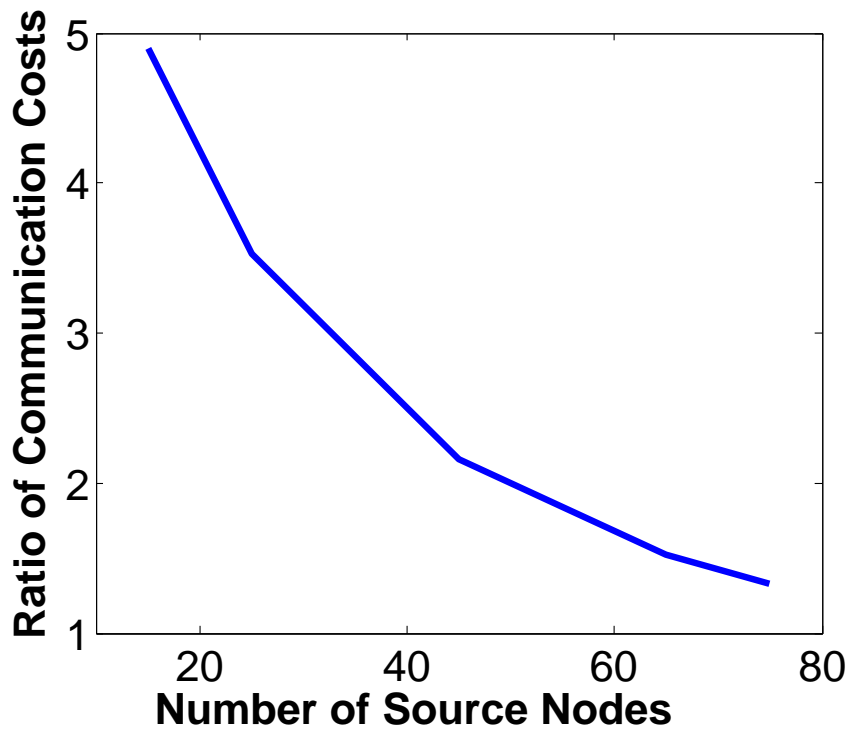
$$T_{AVT} = - \left(\log \sup_{x_{\mathcal{D}}(0) \neq \bar{x}_{\mathcal{S}}} \lim_{t \rightarrow \infty} \left(\frac{\|x_{\mathcal{D}}(t) - \bar{x}_{\mathcal{S}}\|_2}{\|x_{\mathcal{D}}(0) - \bar{x}_{\mathcal{S}}\|_2} \right) \right)^{-1},$$

where x_S and x_D are the states of source and destination nodes respectively.

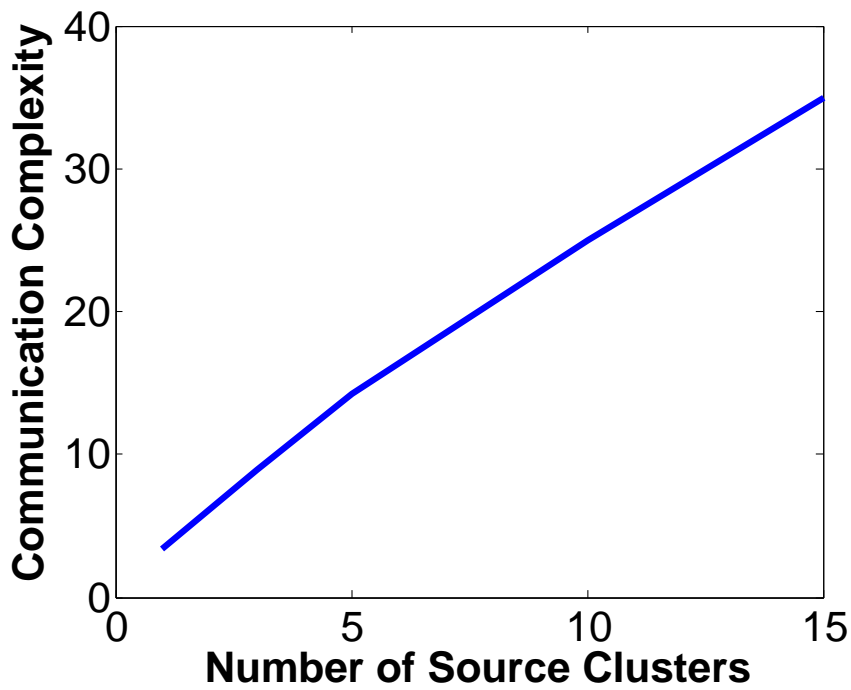
If we were to solve our problem via regular average consensus algorithm as we have discussed in Section 5.2, the communication cost is $\Theta(N r^{-2})$. When, the number of sources, source classes and destination nodes are small, *i.e.*, $M, \mathcal{SC}, K \ll N$, the bound in (5.17) is smaller than $O(N r^{-2})$. We note that this is the regime where (5.17) is tight, thus, we expect that partial AVT solutions will be $N/[M + |\mathcal{SC}|K]$ more efficient than regular average consensus solutions. In the case that $\mathcal{SC}, K \ll N$, our bound is still tight, and AVT will be N/M more efficient. For the regimes where $|\mathcal{SC}|, K$ are large, our bound in (5.17) becomes loose. While it is true that the complexity of the partial AVT solutions will increase, the true complexity is expected to be sub-linear in $|\mathcal{SC}|$ and K .

5.8 Simulations

In this section, we will simulate the behavior of the directed AVT solutions on random graphs, and compare their performances with regular average consensus solutions. Due to the construction complexity of the undirected AVT formulations, they will not be included in our simulations. We first simulate the communication complexity of the directed AVT solutions with respect to the number of source nodes M . To keep the simulation setup as simple as possible, we fix the number of source classes and destinations nodes as one. We choose $N = 100$ and $r = \sqrt{2 \log(N)/N}$. For each value of M , we determine a connected subgraph of the network, and run the average consensus algorithm with constant edge weights [92]. We determine the communication cost as the number of messages required for the error to decrease by the factor $1/e$. We then calculate



(a) Ratio of the communication costs of ACG and directed AVT.



(b) The communication cost of directed AVT.

Figure 5.6: The communication costs of directed AVT.

the number of hops from the source class to the destination node, and add this value to determine the total cost. We generate 100 random geometric graphs for each M , and also averaged our results over 100 random initial node values. Moreover, for each of these initial conditions, we simulate the communication cost of the regular average consensus algorithm, *i.e.*, nodes calculate the average of the whole network. In Fig.5.6(a), we plot the ratio of the communication costs of the regular average consensus and the directed AVT algorithms. As we have mentioned in Section 5.7, AVT requires $M/N < 1$ communication exchanges with respect to the regular average consensus algorithm. The regime is linear with respect to M when $M \leq N/2$, and becomes logarithmic for $M = \Theta(N)$.

In the second part, we simulate the communication complexity of the directed AVT solutions with respect to the number of source classes $|\mathcal{SC}|$. We once again fix the number of destinations as one. We choose $N = 100$ and $r = \sqrt{\log(N)/N}$. For each value of $|\mathcal{SC}|$, we pick $|\mathcal{SC}|$ isolated nodes (in the sense that none of them are neighbors). We then solve the IP programming formulation given in Section 5.6. We, once again, average our results over 100 random graphs. In Fig. 5.6(b), we plot the communication complexity versus the number of source classes. As we have discussed in Section 5.7, the cost increase linearly as the number of source classes increases for $|\mathcal{SC}| \ll N$.

5.9 Extension to Dynamic Networks

In this section, we extend our model into the case where the underlying network is dynamic. One way to integrate the dynamical structure into gossiping algorithms is to consider an asynchronous policy, *i.e.*, nodes wake up at

random times and perform random updates. Unlike *average* consensus algorithms, extension of the AVT codes into asynchronous policies is not straightforward. However, by considering the fact that non-source nodes simply mix and forward the values of the source nodes in the partially directed policies, we propose an asynchronous policy where source nodes keep averaging and non-source nodes keep swapping. Mathematically speaking, we assume that a single node $i \in \mathcal{V}$ is chosen at each discrete time instant $t \geq 0$ with probability $1/N$. The chosen node i selects one of its neighbors uniformly. Then, one of the following arguments hold:

- (1) If $i, j \in \mathcal{S}$, they simply average their values, *i.e.*, $x_i(t+1) = x_j(t+1) = 0.5x_i(t) + 0.5x_j(t)$.
- (2) If $i, j \notin \mathcal{S}$, they swap values, *i.e.*, $x_i(t+1) = x_j(t), x_j(t+1) = x_i(t)$.
- (3) If $i \in \mathcal{S}, j \notin \mathcal{S}$, they calculate a weighted average of their values, *i.e.*, $x_i(t+1) = x_j(t+1) = (1-\gamma)x_i(t) + \gamma x_j(t), \gamma \in [0, 1)$.
- (4) If $j \in \mathcal{S}, i \notin \mathcal{S}$, they calculate a weighted average of their values, *i.e.*, $x_i(t+1) = x_j(t+1) = (1-\gamma)x_j(t) + \gamma x_i(t), \gamma \in [0, 1)$.

We note that for $\gamma > 0$, the average of source nodes will be biased since a given source node i will hear from one of its non-source neighbors j . Since such a setup violates the first constraint in Theorem 3, one can not expect destination nodes to converge to the desired average. In the following, we will show that for $\gamma > 0$, the network will reach to a consensus in expectation.

Lemma 30 *For a connected network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and the asynchronous policy defined above, for $\gamma \in (0, 1)$, the network will reach to a consensus in \mathcal{L}^1 , *i.e.**

$$\lim_{t \rightarrow \infty} E\{x(t)\} = \alpha \mathbf{1}. \quad (5.18)$$

Proof We first note that in the asynchronous case, the network will follow the update rule:

$$x(t+1) = W(t)x(t),$$

where the structure of $W(t)$ depends on the chosen node and its neighbor. For instance, if node i and j are the active ones at time $t \geq 0$, and if $i, j \in \mathcal{S}$, then:

$$W(t) = W_{kl}^{(ij)} = \begin{cases} 0.5 & \text{if } (k, l) = (i, j) \text{ or } (k, l) = (j, i) \\ 1 & \text{if } k = l \text{ and } k \neq i, j \\ 0 & \text{otherwise} \end{cases} \quad (5.19)$$

We note that $W^{(ij)}$ denotes the matrix corresponding to the case where nodes i and j are active. One can construct $W^{(ij)}$ matrices for each of the four cases given above in a similar way. At this point, we define the average matrix W as:

$$W = \frac{1}{N} \sum_{i=1}^N \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} W^{(ij)}. \quad (5.20)$$

W is the average of all possible $W^{(ij)}$ over all nodes in the network and their corresponding neighborhoods. $1/N$ is due to the fact that at each discrete time instant a node i is chosen with probability $1/N$, and node i chooses one of its neighbors uniformly randomly, *i.e.*, with probability $1/|\mathcal{N}_i|$. Since $W^{(ij)}\mathbf{1} = \mathbf{1}$ for all possible (i, j) pairs, W is a stochastic matrix. Moreover, since $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is connected, the average matrix W is irreducible. We also note that diagonals of W are all non-zero.

The expected value of $x(t), t \geq 0$ can be written as:

$$\lim_{t \rightarrow \infty} E \{x(t)\} = \lim_{t \rightarrow \infty} E \left\{ \prod_{k=0}^t W(k)x(0) \right\} \quad (5.21)$$

$$= \lim_{t \rightarrow \infty} E \left\{ \prod_{k=0}^t W(k) \right\} x(0) = \lim_{t \rightarrow \infty} W^t x(0), \quad (5.22)$$

where the last equality follows from the fact that node selection is i.i.d at each discrete index t and W is given in (5.20). Since W is stochastic, irreducible, and has non-zero diagonals, $\lim_{t \rightarrow \infty} W^t$ converges to an agreement matrix [44], i.e., $\beta \mathbf{1}\mathbf{1}^T$ for some $\beta \in \mathbb{R}$. Thus, our result follows.

As we have mentioned, unfortunately, convergence to the true average is not guaranteed unless a very specific chain of wake ups are realized. One way to overcome this difficulty is to set $\gamma = 0$. In this case, source nodes will not hear from the rest of the network, thus the first constraint in Theorem 3 will not be violated.

Lemma 31 *Given a connected network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and the asynchronous policy defined above, $\gamma = 0$, if for each source cluster-destination pair, there exists at least one path in between them which does not go through another source cluster, then, the node values will have a unique stationary distribution: Given there exists $|\mathcal{S}_C|$ source clusters in the network, and denoting their means as $\{\alpha_i\}_{i=1}^{|\mathcal{S}_C|}$ respectively, for a given node $j \notin \mathcal{S}$:*

$$\lim_{t \rightarrow \infty} P(x_j(t) = \alpha_i) > 0 \quad \forall i \in \mathcal{S}_C, \quad (5.23)$$

$$\sum_{i \in \mathcal{S}_C} P(x_j(t) = \alpha_i) = 1. \quad (5.24)$$

Proof Without loss of generality, we will assume that each source cluster consists of a single source node and each non-source node has been initialized with one of the source clusters' value. Due to the fact that nodes are chosen in an i.i.d. fashion at each iteration, the network states $\{x(t)\}_{t=0}^{\infty}$ forms a homogenous Markov Chain \mathcal{M} . In other words, given $x(t)$, $x(t+1)$ and $x(t-1)$ are independent. We denote the set of source clusters as \mathcal{S}_C . The state space of the chain \mathcal{M} has $|\mathcal{S}_C|^{N-|\mathcal{S}_C|}$ elements, since each non-source nodes can assume \mathcal{S}_C distinct

values (due to swapping), and source cluster do not change their values at all (since $\gamma = 0$).

We denote the probability transition matrix of \mathcal{M} as $P_{\mathcal{M}}$. Without loss of generality, let's assume that the chain has non-empty set of inessential states (it may be an empty set depending on the topology). Therefore, by potentially reordering the state space of \mathcal{M} , we can partition $P_{\mathcal{M}}$ as:

$$P_{\mathcal{M}} = \begin{bmatrix} P_1 & 0 \\ R & Q \end{bmatrix}, \quad (5.25)$$

such that $\lim_{k \rightarrow \infty} Q^k = 0$. We note that Q matrix represents the transition probabilities in between nonessential states. The set of indices corresponding to P_1 are the set of essential states. We will denote these indices as \mathcal{M}_1 . In the following, we will prove that the set of essential states forms a single class, *i.e.*, if $y, z \in \mathcal{M}_1$, then $y \leftrightarrow z$. Note that both y and z correspond to an N -dimensional state vector in our model. Let's denote these state vectors as x_y and x_z . Let's define the set $\mathcal{D} = \{m : [x_y]_m \neq [x_z]_m\}$, *i.e.*, the set of indices that are different in the configurations x_y and x_z .

For a given $m \in \mathcal{D}$, choose one of the source clusters with value $[x_y]_m$ or $[x_z]_m$ and determine a path from this particular cluster to m which does not go through any other source clusters. This has to be true, since otherwise, at least one of y and z will be inessential. Then, consider the following chain of events: The first non-source node on the path is chosen with $1/N$ probability and it selects the source node, and performs the update. Then, the second non-source node is chosen and it selects the first non-source node, so on. It should be clear that this particular event has non-zero probability for state vectors x_y and x_z . We repeat this particular argument *number of nodes on the path* times. At this

point, all of the nodes on the path including the node m have values which are equal to the value of the chosen source cluster. Therefore, for a given $m \in \mathcal{D}$, this particular event will transform x_y and x_z into $x_{y'}$ and $x_{z'}$ respectively, where $\mathcal{D}' = \{m : |[x_{y'}]_m \neq [x_{z'}]_m\}$ is a *strict* subset of \mathcal{D} . If we apply the argument above for each $m \in \mathcal{D}$ sequentially, then x_y and x_z will be transformed into $x_{y^*} = x_{z^*}$.

We note that since y is an essential state and $y \rightarrow y^*$, then y^* is also an essential state, *i.e.*, $y \leftrightarrow y^*$. By the same way, one can show that $z \leftrightarrow z^*$. Finally, since $y^* = z^*$ and essentiality is transitive, $y \leftrightarrow z$.

At this point, we have already shown that the chain \mathcal{M} has a single essential class. Moreover, the essential class is aperiodic since there exists at least one aperiodic essential state, *i.e.*, each non-source equals to the same source node's value.

The following lemma is due to Seneta [84]:

Lemma 32 *Let \mathcal{M} a Markov chain which has a single aperiodic essential class. Given its probability matrix $P_{\mathcal{M}}$ in the canonical form as in (5.25), define the stationary distribution to the primitive submatrix P_1 of $P_{\mathcal{M}}$ as v'_1 . Let $v' = (v'_1, 0')$ be an $1 \times N$ vector. Then as $k \rightarrow \infty$:*

$$P_{\mathcal{M}}^k \rightarrow \mathbf{1}v', \quad (5.26)$$

where v' is the unique stationary distribution corresponding to the chain \mathcal{M} and $\mathbf{1}$ is the all ones vector.

Since our \mathcal{M} has a single aperiodic essential class, we can utilize the lemma above and prove that a stationary distribution exists and such a distribution is unique for our model.

Finally, we need to show that for each destination node j , there exists a set of essential states $\{y_i\}_{i=1}^{\mathcal{S}_C}$ where $[x_{y_i}]_j = \alpha_i$, for all $i \in \mathcal{S}_C$. In other words, the probability that a given destination node j being equal to a given source cluster i 's value is non-zero. To see this, we only need to remind our readers that, by the hypothesis, for each destination node-source cluster pair, there exists at least one path which does not go through any other source clusters. Thus, we can find a chain of events that will change destination nodes values to a particular source cluster's value. This concludes our proof.

We note that if there is a single source cluster, the destination nodes will converge to that particular value. On the other hand, if there exists more than one source cluster, then, the network will have a unique stationary distribution which is independent of the initial values of non-source nodes. Moreover, each non-source node will keep switching its value forever, but, the support of the values it can switch is finite, and indeed these values are equal to the averages of each source cluster by (5.23)-(5.24). Therefore, in the long run, destination nodes start observing $|\mathcal{S}_C|$ distinct values, each of which are the averages of individual source clusters. Assuming all of the clusters have the same size, true average can be calculated by taking average of these observed quantities.

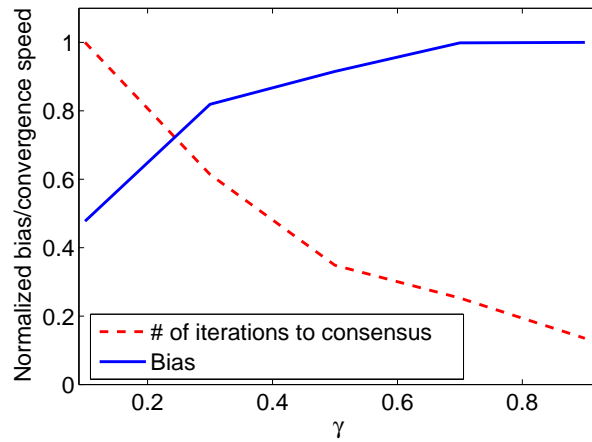
Finally, we would like to note the effect of the mixing parameter γ . As γ increases, the averages of distinct source clusters will mix faster due to increased information flow between the classes through the non-source nodes. But, the bias from the true average will also increase as γ increases. Thus, there is a clear trade-off between convergence speed and bias with respect to the γ parameter. At this point, we will leave the analysis of the convergence speed of the algorithm for both $\gamma > 0$ and $\gamma = 0$ as a future work, as well as the ana-

lytical characterization of the trade off and the complete characterization of the probabilities given in (5.23).

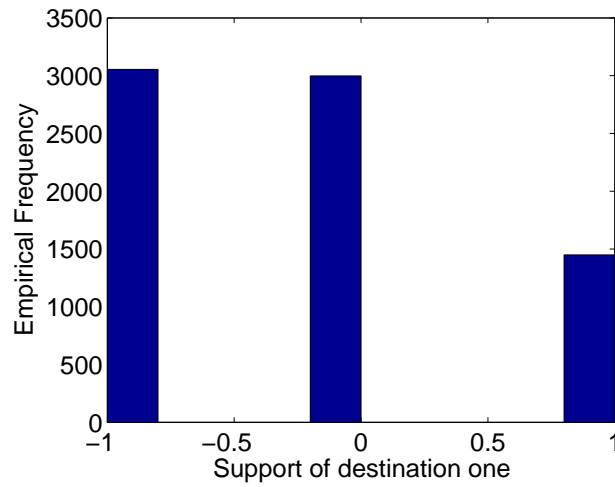
To justify our claims, we have simulated the proposed algorithm on random graphs. We have generated 100 geometric random graphs with $N = 50$ and connectivity radius $\sqrt{2 \log N/N}$. Three source nodes and three destination nodes are chosen randomly. For each graph, the algorithm is run for 5×10^5 steps. In Fig. 5.7(a), we plot the trade-off in between convergence speed/bias and the mixing parameter γ . As we have discussed above, the higher γ is, the faster network reaches an agreement. On the other hand, as γ increases, the bias from the initial mean of the source nodes increases. To validate Lemma 31, we have chosen $\gamma = 0$. Three source nodes are chosen such that all the assumptions in the lemma holds. Source nodes are given the values of $\{-1\}, \{0\}, \{1\}$ respectively. Fig. 5.7(b)-5.7(c) shows the histogram of the two destination nodes in the long run. As we can see from these plots, destination nodes' values only fluctuates between source nodes' values.

5.10 Discussions

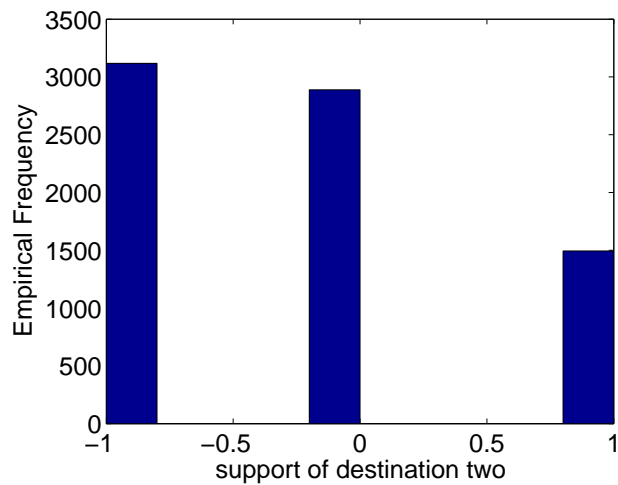
In this chapter, we studied the distributed computation problem, where a set of destination nodes are interested in the average of another set of source nodes. Utilizing gossiping protocols for the information exchange, we provided necessary and sufficient conditions on the feasibility of AVT codes. Moreover, we showed that the feasibility of the problem only depends on the asymptotic behavior of the source states and is independent of the evolution of these states. We analyzed the spectral properties of feasible codes, and proposed several sim-



(a) MSE-Convergence speed versus γ .



(b) Support of destination one.



(c) Support of destination two.

Figure 5.7: Empirical Results.

plifications which reduce the complexity of the problem without affecting the feasibility. By focusing on stochastic updates, we provided necessary condition on the feasibility that are easy to interpret, and then discussed some known infeasible scenarios. Since the feasible region of AVT codes are non-convex, we introduce so called partially directed AVT solutions, and provided integer programming formulation for constructing such solutions. We analyzed the complexity and communication cost of the algorithm and compared the performance of our algorithm with existing solutions. Finally, we provided a simple extension to the asynchronous update scenario which can be implemented on dynamical network structures.

CHAPTER 6
OPINION DIFFUSION IN SOCIAL NETWORKS VIA GOSSIPING

6.1 Introduction

6.1.1 Motivation and Related Work

A social network can be defined as a group of people which are connected with some sort of ties [36]. These ties include friendship, kinship, and economic relationships. Social networks are an integral part of our social and economic lives. For instance, these networks play a major role in the transmission of information about recently introduced products and they are important for all kinds of trading activities. Moreover, they determine how diseases spread, how much and what we learn, what kind of music we prefer and whether we hear about certain job opportunities or not. They even affect our political opinions and how we vote.

In this chapter, we will study how individuals' opinions diffuse in a given social network. Diffusions through social networks, in general, have been studied to some extent in the literature. One of the earliest diffusion models has been introduced by Bass in [15]. While a mathematical model for new product diffusion has been discussed in this particular study, the model itself did not capture the dependence on the structure of the underlying social network. The spread of disease through a network has been originally discussed in [14, 54] under *susceptible, infected, susceptible* (SIS) and *susceptible, infected, removed* (SIR) models respectively. In both SIS and SIR models, the diffusion is explained through

the process of infected agents infecting their susceptible neighbors with certain probabilities. Diffusion has also been discussed under social learning particularly with Bayesian and Observational Learning theory. In this type of work, individuals observe actions of their immediate neighbors (actions of individuals are functions of their beliefs on the true parameter θ) as well as noisy observations of θ . Following these observations, they update their beliefs by Bayesian update rules. A detailed discussion about these models can be found in [46].

Opinion diffusion has been studied under imitation and social influence models. The seminal work on this particular area is due to DeGroot [30] where opinions of individuals are modeled as probabilities that might be thought of the probability that a given statement is true. The interaction patterns are captured through near neighbor based linear updates. We note that the updates of the DeGroot model has the exact the same as the synchronous average consensus updates. Moreover, Kleinberg *et.al.* have considered a collection of probabilistic and game theoretic models for capturing opinion and new product diffusion in social networks [56]. Finally, in their recent work, Acemoglu *et.al.* have studied an extension of the DeGroot model where updates are asynchronous and certain individuals are spreading misinformation by not updating their own beliefs [1].

In this work, we will focus on opinion diffusion through social networks and propose a gossiping based model where individuals randomly meet with their neighbors and probabilistically update their opinions. Our work can be classified under imitation and social influence modeling. However, unlike the DeGroot and misinformation models, we assume that individuals' opinions are discrete rather than continuous variables and there exists so called *stubborn*

individuals who do not change their decisions. Therefore, unlike the models in [30, 1, 56], individuals' opinion do not converge to a particular constant, on the contrary, the opinions keep flipping in the long run. This, in return, will help us to capture correlations among the opinions of different individuals as well as the variations in the collective opinion of the society. We note that unlike Bayesian learning and observation models, we do not assume that there is a true parameter which the society is interested in capturing. We are interested in the propagation of opinions on a certain subject.

6.1.2 Chapter Organization

The remainder of this chapter is organized as follows. In Section 6.2, we will introduce our diffusion model as well as discussing mathematical tools on which we will rely. We derive the mean and the variance of the society's collective opinion in Section 6.3, and discuss how to choose stubborn agents' location in an optimal way in Section 6.4. Finally, we provide end of chapter discussions in Section 6.5.

6.2 The Voter Model and the Dual Approach

6.2.1 The Binary Voter Model with Stubborn Agents

We consider a network $G(\mathcal{V}, \mathcal{E})$ where \mathcal{V} is the set of vertices and \mathcal{E} is the set of undirected edges. The underlying network can be deterministic or random. We define the neighbor set of node $i \in \mathcal{V}$ as $\mathcal{N}_i = \{j | (i, j) \in \mathcal{E}\}$. We note that due

to undirected nature of the edges, $j \in \mathcal{N}_i$ if and only if $i \in \mathcal{N}_j$. The adjacency matrix $A \in \{0, 1\}^{N \times N}$ of a graph G is a binary matrix, where $[A]_{ij} = 1$ whenever $(i, j) \in \mathcal{E}$. The degree matrix D of the graph G is a diagonal matrix whose i -th diagonal element is equal to $|\mathcal{N}_i|$. Finally, $\lambda_k(\cdot)$ denotes the k -th largest eigenvalue (in magnitude) of its argument.

The state of the binary voter model at time $t \geq 0$ is given by a function $x(t) : \mathcal{V} \rightarrow \mathcal{S}^{|\mathcal{V}|}$, where \mathcal{S} the set of possible types (colors, opinions). Due to its *binary* nature, there are only two possible types, *i.e.*, $\mathcal{S} = \{0, 1\}$. We note that one can generalize our model to the case where there exist more than two possible type. However, the binary voter model covers several interesting case and the extension to the latter case will be straightforward. Each voter, $i \in \mathcal{V}$, wakes up according to a rate 1 Poisson process independently, chooses one of its neighbors according to a probability distribution p_i and adapts the decision of the chosen neighbor. Initial node types $x(0)$ can be either deterministic or random.

We will further assume that there exists two disjoint sets of *stubborn* agents, *i.e.*, $\mathcal{V}_0, \mathcal{V}_1 \subset \mathcal{V}$, where:

$$x_i(t) = \begin{cases} 0 \forall t \geq 0, & \text{if } i \in \mathcal{V}_0 \\ 1 \forall t \geq 0, & \text{if } i \in \mathcal{V}_1. \end{cases} \quad (6.1)$$

In words, so called stubborn nodes do not adapt the decisions of their neighbors and always keep their initial opinions. We call this model as the *Binary voter model with stubborn nodes*. Our aim to understand the effects of the stubborn nodes (locations, number of stubborn agents, network structure, etc.) on the opinions of the society and individuals.

The following assumptions are in order:

- A1)** The number of nodes in the network $|\mathcal{V}| = N$ is finite.
- A2)** The underlying topology G is strongly connected, *i.e.*, there exists a path (not necessarily single hop) between any given node pairs.
- A3)** p_i is the uniform distribution on \mathcal{N}_i , *i.e.*, node i chooses its neighbors uniformly.

We note that $A2$ simply means that every individual in the network can reach every other individual by a path. We note that this path may not necessarily be a single hop. $A3$ shows that each node is uniformly influenced by its neighbors.

6.2.2 Effects of Stubborn Agents: From Consensus to Disagreement

In the classical binary voter model where there exists no stubborn agents, Aldous et.al. have shown that, under the assumptions $A1 - A3$, the node states $x(t)$ converge to a complete consensus in the limit [6]. Mathematically speaking,

$$\lim_{t \rightarrow \infty} P(x_i(t) \neq x_j(t)) = 0 \quad \forall i, j \in \mathcal{V}. \quad (6.2)$$

Even a stronger result has been introduced by Liggett where he has relaxed the "finiteness" assumption on the network size. In particular, he has shown that the classical voter model converges to a complete consensus on $d = 1$ and $d = 2$ dimensional infinite lattices [61, 43, 27]. For $d \geq 3$, he has argued that the system $x(t)$ approaches a non-trivial equilibrium where the individual state values are neither independent nor fully correlated with each other.

In our model, *i.e.*, under the existence of stubborn agents, the system can not reach to a consensus on any network. We note that stubborn nodes do not

change their opinions, and since there are two types of stubborn nodes with opposite opinions, consensus can not be achieved. At this point, a natural question to ask is *whether a stationary distribution exists or not?* In the following, we will show that under the assumptions A1 – A3 not only a stationary distribution exists but also such a distribution is unique.

Theorem 4 *Under the assumptions A1 – A3, the voter model with stubborn agents has a stationary distribution which is unique. Mathematically speaking, for given stubborn sets $\mathcal{V}_0, \mathcal{V}_1$:*

$$x(t) \xrightarrow{D} x^*, \text{ for some random variable } x^*, \quad (6.3)$$

where \xrightarrow{D} denotes weak convergence.

Proof We first note that since the node wake-up times are Poisson random variables, the voter model operates in continuous time scale. On the other hand, we can use a discrete index k to identify the sequence of updating events in the network and track the evolution of the states with respect to this index. That is, $k = 1$ corresponds to the first update in the network, $k = 2$ to the second, and so on. Of note is that each of these discrete indexes corresponds to a time interval $[\tau_k, \tau_{k+1})$, where τ_k is the arrival time of the k -th update in the network. In the rest of the proof, we slightly bend our notation and denote $x(\tau_k)$ by $x(k)$.

We can characterize the behavior of $x(k)$ by a homogenous Markov Chain \mathcal{M} with the state space $\mathcal{S}_{\mathcal{M}}$. Each state of the chain is an N dimensional vector, where each dimension represents the value of a particular node in \mathcal{V} . We note that the Markovian property of the chain follows from the fact that $x(k + 1)$ and $x(k - 1)$ are independent conditioned on $x(k)$ ¹. Moreover, the cardinality of the

¹Each node wakes up according to the Poisson processes with rate 1 independently.

state space $\mathcal{S}_{\mathcal{M}}$ is $2^{N-|\mathcal{V}_0 \cup \mathcal{V}_1|}$, since stubborn agents do not change their decisions and non-stubborn agents can only take two possible values, *i.e.*, 0 or 1.

We denote the probability transition matrix of \mathcal{M} as $P_{\mathcal{M}}$. Without loss of generality, let's assume that the chain has non-empty set of inessential states. Therefore, by potentially reordering the state space of \mathcal{M} , we can partition $P_{\mathcal{M}}$ as:

$$P_{\mathcal{M}} = \begin{bmatrix} P_1 & 0 \\ R & Q \end{bmatrix}, \quad (6.4)$$

such that $\lim_{k \rightarrow \infty} Q^k = 0$. We note that Q matrix represents the transition probabilities in between nonessential states. The set of indices corresponding to P_1 are the set of essential states. We will denote these indices as \mathcal{M}_1 . In the following, we will prove that the set of essential states forms a single class, *i.e.*, if $y, z \in \mathcal{M}_1$, then $y \leftrightarrow z$. Note that both y and z correspond to an N -dimensional state vector in the binary voter model. Let's denote these state vectors as x_y and x_z . Let's define the set $\mathcal{D} = \{m : |[x_y]_m \neq [x_z]_m\}$, *i.e.*, the set of indices that are different in the configurations x_y and x_z . We first note that for each $m \in \mathcal{D}$, there has to exist a path from node m to at least one of the agents in the set \mathcal{V}_0 that does not go through \mathcal{V}_1 , and vice versa. We note that if this were not true, then at least one of x_y, x_z would be an inessential state.

For a given $m \in \mathcal{D}$, determine a path from \mathcal{V}_1 to node $m \in \mathcal{V}$ which does not go through \mathcal{V}_0 . Then, consider the following chain of events: The first non-stubborn node along the path chooses a stubborn node in \mathcal{V}_1 and adopts its decision, the second non-stubborn node chooses the first non-stubborn node and adopts its decision, so on. It should be clear that this particular event has non-zero probability for state vectors x_y and x_z . Therefore, for a given $m \in \mathcal{D}$,

this particular event will transform x_y and x_z into $x_{y'}$ and $x_{z'}$ respectively, where $\mathcal{D}' = \{m : |[x_{y'}]_m| \neq |[x_{z'}]_m|\}$ is a *strict* subset of \mathcal{D} . If we apply the argument above for each $m \in \mathcal{D}$ sequentially, then x_y and x_z will be transformed into $x_{y^*} = x_{z^*}$.

We note that since y is an essential state and $y \rightarrow y^*$, then y^* is also an essential state, *i.e.*, $y \leftrightarrow y^*$. By the same way, one can show that $z \leftrightarrow z^*$. Finally, since $y^* = z^*$ and essentiality is transitive, $y \leftrightarrow z$.

At this point, we have already shown that the chain \mathcal{M} has a single essential class. Moreover, each of these essential states is aperiodic since there is a nonzero probability that a stubborn node will wake up, and thus the chain will stay in the same state.

The following lemma is due to Seneta [84]:

Lemma 33 *Let \mathcal{M} a Markov chain which has a single aperiodic essential class. Given its probability matrix $P_{\mathcal{M}}$ in the canonical form as in (6.4), define the stationary distribution to the primitive submatrix P_1 of $P_{\mathcal{M}}$ as v'_1 . Let $v' = (v'_1, 0')$ be an $1 \times N$ vector. Then as $k \rightarrow \infty$:*

$$P_{\mathcal{M}}^k \rightarrow \mathbf{1}v', \quad (6.5)$$

where v' is the unique stationary distribution corresponding to the chain \mathcal{M} and $\mathbf{1}$ is the all ones vector.

Finally, since our \mathcal{M} has a single aperiodic essential class, we can utilize the lemma above and prove that a stationary distribution exists and such a distribution is unique for our model.

We note that the unique stationary distribution is a function of the underlying graph \mathcal{G} and the stubborn sets $\mathcal{V}_0, \mathcal{V}_1$, and is independent of the initial opin-

ions of the non-stubborn agents. This, in return, implies that the initial opinions of the non-stubborn agents does not have any effect on the collective opinion of the society in the long run.

In the following, we will try to characterize some of the properties of the stationary distribution of the system. But, first, we will introduce the dual approach for the voter model with stubborn agent on which we will rely on heavily for future analysis.

6.2.3 The Dual Approach for the Voter Model with Stubborn Nodes

One of the major approaches in the analysis of the original voter model is so called the *dual approach*, where the voting problem is being mapped into *coalescing random walks* on the graph G . We note that the dual approach for the original voter model (without stubborn agents) has been introduced in [61, 43]. Reader may refer those references for detailed discussions and proofs of the duality. In the following, we will first summarize the results for the original voter model for the sake of completeness and, then introduce the dual process for our model.

The Dual Approach for the Original Voter Model

We start our discussion by introducing the definition of *coalescing random walks*. In the coalescing random walks process, there is a single particle at each vertex of the graph \mathcal{G} at time $t = 0$. These particles perform independent continuous random walks on the graph, *i.e.*, each particle jumps to one of its neighbors in-

dependently according to a rate 1 Poisson process. When two or more particles meet on a vertex, they coalesce and form a single particle. Thereafter, these particles perform a single random walk on \mathcal{G} , possibly colliding with other particles.

In the following, we will discuss that coalescing random walks are closely related to the behavior of the original voter model. The idea is based on the following: for each vertex $i \in \mathcal{V}$ and time $T \geq 0$, one can define a dual process $\{y_i^T(s) : 0 \leq s \leq T\}$, which traces the origin of the opinion at node i at time $T \geq 0$. The primal and the dual processes will have the property:

$$x_i(T) = x_{y_i^T(T-s)}(s). \quad (6.6)$$

In other words, the origin of the opinion at node i at time $T \geq 0$, is equal to the opinion at node $y_i^T(T-s)$ at time s . We note that $y_i^T(0) = i$, and the process $\{y_i^T(T-s) : 0 \leq s \leq T\}$ will jump according to the arrival times of the primal process $x(t)$. For instance, if $t_1 \leq T$ is the last time before (and including) T that node i has copied one of its neighbors decision (say neighbor j), then y_i^T will jump to the neighbor j at time $T - t_1$. A sample update path is given in Fig. 6.1. In this example, the x-axis denotes different nodes in the network, and the y-axis denotes continuous time interval. Whenever a node copies its neighbor's decision, it will be denoted by an arrow from the node itself to its neighbor. As we have discussed above, node i has copied node j 's decision at time t_1 , and there is a corresponding arrow in the figure. We note that one can track the origin of the node i 's decision at time T by following arrows backwards in time, *i.e.*, by the dual process. In our example, node i 's decision at time T is node k 's initial decision. We note that since each node copies one of its neighbors decision according to a rate 1 Poisson process, the dual process is also a rate 1 random walk. Moreover, if we want to track the origins of opinions at more than one location,

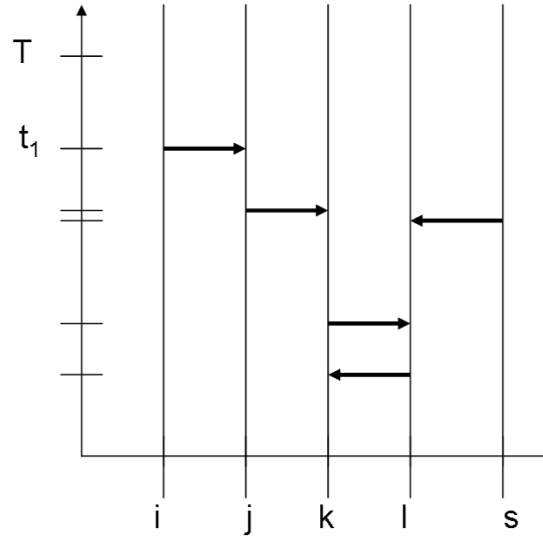


Figure 6.1: A sample update path for the voter algorithm. x-axis denotes node index and y-axis denotes continuous time index.

i.e., at $\mathcal{B} \subset \mathcal{V}$, then we can define an independent dual process for each element $i \in \mathcal{B}$. The crucial point is that if two or more dual processes reside at the same location at the same time, they will collide and move together, since the origin of the decision at that point and time will be exactly the same for both processes. We note that this process is the exact same process as a coalescing random walk. Therefore, the dual process for the original voter model is a coalescing random walks with rate 1.

The Dual Approach for the Voter Model with Stubborn Agents

In our case, where there exists stubborn nodes with fixed decisions, we can still define a dual process in terms of coalescing random walks with slight modification: In the dual processes $y_i^T(T - s)$, the stubborn nodes will be absorbing states. In other words, when a dual particle hits one of the stubborn nodes $i \in \mathcal{V}_0 \cup \mathcal{V}_1$, it will stay in that particular node. Mathematically speaking, if

$y_i^T(T - s^*) = j | j \in \mathcal{V}_0 \cup \mathcal{V}_1$ for some $s^* \leq T$, then $y_i^T(T - s) = j \forall s^* \leq s \leq T$. Such an observation is intuitive since if we track down the opinion of node i at time T via particle jumps over the neighbors, and if such a sample path hits one of the stubborn agents, the process will be stuck on that particular stubborn node since stubborn nodes do not change their decisions, thus the particle can not jump to any other node in the network. Therefore, the dual process for the voter model with stubborn agents is simply the coalescing random walk with absorbing states $\mathcal{V}_0, \mathcal{V}_1$.

6.3 Characterization of the Average Opinion: Mean and Variance

One way to understand the stationary distribution of the agent opinions is to characterize society's average opinion, *i.e.*,

$$\bar{x}(t) = \frac{1}{N} \sum_{i \in \mathcal{V}} x_i(t). \quad (6.7)$$

While such a measure does not uniquely define opinion of each individual, it measures the collective opinion of the society which is, in some cases, more important than individual opinions (for instance political elections). We first introduce the following result:

Corollary 7 *The average opinion of the society has a unique stationary distribution under the assumptions A1 – A3.*

The corollary simply follows from Theorem 4. Unfortunately, characterizing the stationary distribution of $\bar{x}(t)$ is very difficult since the support of the distribution can have up to 2^N distinct values. As identifying this distribution is an

ongoing research, we will, in the following, characterize the mean and the variance of $\lim_{t \rightarrow \infty} \bar{x}(t)$. While the first two moments of the distribution does not characterize it uniquely, we will still have some measure of what is the expected dominant opinion and its variance.

6.3.1 Expected Value of the Average Opinion

In this section, we will characterize the expected value of the average opinion in the network in the long run (stationary distribution). If we take the expectation and limit of both sides in (6.7)

$$\lim_{t \rightarrow \infty} \mathbb{E} \{ \bar{x}(t) \} = \lim_{t \rightarrow \infty} \frac{1}{N} \mathbb{E} \left\{ \sum_{i \in \mathcal{V}} x_i(t) \right\}, \quad (6.8)$$

$$= \frac{1}{N} \sum_{i \in \mathcal{V}} \lim_{t \rightarrow \infty} \mathbb{E} \{ x_i(t) \} \quad (6.9)$$

where the second equality follows from the assumption that the individual limits exists. The following lemma not only shows that the individual limits exist (thus the assumption is true) but also characterizes $\lim_{t \rightarrow \infty} \mathbb{E} \{ x_i(t) \}$ in terms of the absorption probabilities of a random walk over the graph \mathcal{G} which has been initialized at node i .

Lemma 34 *Given a connected network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with stubborn agent sets \mathcal{V}_0 and \mathcal{V}_1 , define a regular random walk on \mathcal{G} where the stubborn sets form two distinct absorbing classes. Then the expected value of the opinion of node i in the limit is equal to the probability that the random walk is absorbed by the set \mathcal{V}_1 given that the walk has been started at node i .*

Proof Without loss of generality, we assume that $\{1, \dots, |\mathcal{V}_0|\}$ nodes are stubborn agents with opinion zero and $\{|\mathcal{V}_0| + 1, \dots, |\mathcal{V}_0| + |\mathcal{V}_1|\}$ nodes are stubborn

agents with opinion one. Similar to the proof of Theorem 4, we will slightly bend our notation and denote $x(\tau_k)$ by $x(k)$, where τ_k is the time when k -th update is performed in the network.

We first note that for any $k \geq 1$ and $i \notin \{\mathcal{V}_0 \cup \mathcal{V}_1\}$,

$$\mathbb{E}\{x_i(k)|x(k-1), \dots, x(0)\} = P(x_i(k) = 1|x(k-1), \dots, x(0)), \quad (6.10)$$

$$= \sum_{s \in \{0,1\}} P(x_i(k) = 1|B_i(k) = s, x(k-1)) P(B_i(k) = s), \quad (6.11)$$

$$= \frac{N-1}{N}x_i(k-1) + \frac{1}{N} \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} x_j(k-1), \quad (6.12)$$

where $B_i(k)$ is the indicator of the event that node i performs an update at the discrete iteration k . (6.10) follows from the fact that $x_i(k)$ is a Bernoulli random variable, (6.11) is since $\{x(k)\}_{k=0}^{\infty}$ forms a Markov Chain and Poisson arrivals are independent from node values, and (6.12) follows from the fact that Poisson arrivals are i.i.d and nodes choose their neighbors uniformly. Moreover, since stubborn nodes do not update their beliefs, for any $k \geq 1$ and $i \in \{\mathcal{V}_0 \cup \mathcal{V}_1\}$, the following holds true,

$$x_i(k) = x_i(k-1). \quad (6.13)$$

Combining (6.12) and (6.13), the formulation given above can be written in the matrix form as:

$$\mathbb{E}\{x(k)|x(k-1), \dots, x(0)\} = Tx(k-1), \quad (6.14)$$

where stochastic matrix T has the structure:

$$T = \begin{bmatrix} I_A & 0 \\ C & D \end{bmatrix}, \quad (6.15)$$

and I_A denotes $(|\mathcal{V}_0| + |\mathcal{V}_1|) \times (|\mathcal{V}_0| + |\mathcal{V}_1|)$ identity matrix. Moreover, the partitions C and D will be in the form of:

$$[C]_{ij} = \begin{cases} \frac{1}{N|\mathcal{N}_{i+|\mathcal{V}_0|+|\mathcal{V}_1|}|} & \text{if } j \in \mathcal{V}_0 \cup \mathcal{V}_1 \cap \mathcal{N}_{i+|\mathcal{V}_0|+|\mathcal{V}_1|}, \\ 0 & \text{otherwise.} \end{cases} \quad (6.16)$$

$$[D]_{ij} = \begin{cases} \frac{N-1}{N} & \text{if } i = j \\ \frac{1}{N|\mathcal{N}_{i+|\mathcal{V}_0|+|\mathcal{V}_1|}|} & \text{if } j \in \mathcal{V}_0 \cup \mathcal{V}_1 \cap \mathcal{N}_{i+|\mathcal{V}_0|+|\mathcal{V}_1|}, \\ 0 & \text{otherwise.} \end{cases} \quad (6.17)$$

We note that the structure given above directly follows from (6.10)-(6.12), and $|\mathcal{V}_0| + |\mathcal{V}_1|$ shift in the indices is due to the fact that the first $|\mathcal{V}_0| + |\mathcal{V}_1|$ nodes are stubborn.

By taking expectations over the past state values, we obtain:

$$\mathbb{E}\{x(k)\} = T^k x(0). \quad (6.18)$$

By taking the limit over k :

$$\lim_{k \rightarrow \infty} \mathbb{E}\{x(k)\} = \lim_{k \rightarrow \infty} T^k x(0), \quad (6.19)$$

$$\lim_{k \rightarrow \infty} T^k = \begin{bmatrix} I_A & 0 \\ \left(\sum_{l=1}^k D^l\right) C & D^k \end{bmatrix} = \begin{bmatrix} I_A & 0 \\ (I_B - D)^{-1} C & 0 \end{bmatrix}, \quad (6.20)$$

where I_B is the $(N - |\mathcal{V}_0| - |\mathcal{V}_1|) \times (N - |\mathcal{V}_0| - |\mathcal{V}_1|)$ identity matrix. We note that $\lim_{k \rightarrow \infty} T^k$ exists since T is stochastic, and $\rho(D) < 1$. Then, $\lim_{k \rightarrow \infty} D^k = 0$ and $(I_B - D)^{-1}$ exist. Therefore, (6.19) and (6.20) are well-defined.

At this point, we focus on the probability transition matrix \tilde{T} of the regular random walk with absorbing classes \mathcal{V}_0 and \mathcal{V}_1 on graph G . In other words, for a given state $i \in \mathcal{V} \setminus \{\mathcal{V}_0 \cup \mathcal{V}_1\}$, process will uniformly jump to one of i 's neighbors on graph G , and if i is an absorbing state, the process will stay at i forever. Thus,

\tilde{T} will be in the form:

$$\tilde{T} = \begin{bmatrix} \tilde{I}_A & 0 \\ \tilde{C} & \tilde{D} \end{bmatrix}, \quad (6.21)$$

where \tilde{C} and \tilde{D} will have the following structure:

$$[\tilde{C}]_{ij} = \begin{cases} \frac{1}{|\mathcal{N}_{i+|\mathcal{V}_0|+|\mathcal{V}_1|}} & \text{if } j \in \mathcal{V}_0 \cup \mathcal{V}_1 \cap \mathcal{N}_{i+|\mathcal{V}_0|+|\mathcal{V}_1|}, \\ 0 & \text{otherwise.} \end{cases} \quad (6.22)$$

$$[\tilde{D}]_{ij} = \begin{cases} \frac{1}{|\mathcal{N}_{i+|\mathcal{V}_0|+|\mathcal{V}_1|}} & \text{if } j \in \mathcal{V}_0 \cup \mathcal{V}_1 \cap \mathcal{N}_{i+|\mathcal{V}_0|+|\mathcal{V}_1|}, \\ 0 & \text{otherwise.} \end{cases} \quad (6.23)$$

Since \mathcal{G} is connected and $N < \infty$, $\lim_{k \rightarrow \infty} \tilde{T}^k$ exists [84], and non-zero columns of this particular matrix corresponds to the absorption probabilities [84]. In the following, we will show that $\lim_{k \rightarrow \infty} T^k = \lim_{k \rightarrow \infty} \tilde{T}^k$.

Parallel to our arguments above, $\lim_{k \rightarrow \infty} \tilde{T}^k$ will have the same structure as in (6.20) in terms of \tilde{I}_A , \tilde{C} and \tilde{D} . Moreover, we note that the following relations hold between T and \tilde{T} :

$$I_A = \tilde{I}_A, \quad C = \frac{1}{N}\tilde{C}, \quad D = \frac{1}{N}\tilde{D} + \frac{N-1}{N}I_B.$$

Therefore,

$$(I_B - D)^{-1}C = (I_B - \frac{1}{N}\tilde{D} - \frac{N-1}{N}I_B)^{-1}\frac{1}{N}\tilde{C} \quad (6.24)$$

$$= (\frac{1}{N}I_B - \frac{1}{N}\tilde{D})^{-1}\frac{1}{N}\tilde{C} = (I_B - \tilde{D})^{-1}\tilde{C}. \quad (6.25)$$

Thus, due to the structure given in (6.20), $\lim_{k \rightarrow \infty} T^k = \lim_{k \rightarrow \infty} \tilde{T}^k$.

By noting the fact that:

$$x_i(0) = \begin{cases} 0, & \text{if } i \in \{1, \dots, |\mathcal{V}_0|\} \\ 1, & \text{if } i \in \{|\mathcal{V}_0| + 1, \dots, |\mathcal{V}_0| + |\mathcal{V}_1|\}. \end{cases} \quad (6.26)$$

one can observe that,

$$\lim_{k \rightarrow \infty} \mathbb{E}\{x_i(k)\} = \sum_{j \in \mathcal{V}_1} [\lim_{k \rightarrow \infty} T^k]_{ij}. \quad (6.27)$$

Since we have already proved that non-zero columns of $\lim_{k \rightarrow \infty} \tilde{T}^k$ are the absorption probabilities of the random walk and $\lim_{k \rightarrow \infty} T^k = \lim_{k \rightarrow \infty} \tilde{T}^k$, (6.27) is the probability that the regular random walk is absorbed by the class \mathcal{V}_1 given that the walk has been initialized at node i .

We note that the random walk introduced in the lemma is over the graph \mathcal{G} , thus the cardinality of the state space of the walk is just N . Moreover, the lemma shows the clear relationship between the location of the stubborn agents and the expected opinion of a given node i . For a given graph \mathcal{G} , and sets $\mathcal{V}_1, \mathcal{V}_0$, the absorption probabilities defined in Lemma 34 are straightforward to calculate.

At this point, the following corollary in order:

Corollary 8 *The expected value of the average node opinion in the limit is equal to:*

$$\lim_{t \rightarrow \infty} \mathbb{E}\{\bar{x}(t)\} = \frac{1}{N} \sum_{i \in \mathcal{V}} p_i, \quad (6.28)$$

where p_i is the probability that the random walk defined in Lemma 34 is absorbed by the class \mathcal{V}_1 given that the walk has been started at node i .

At this point, we note that there is a simple relationship between expected value of the average opinion on \mathcal{G} and the random walk on \mathcal{G} through the absorption probabilities p_i .

6.3.2 Variance of the Average Opinion

If we calculate the variance of the average opinion and take the limit as $t \rightarrow \infty$:

$$\lim_{t \rightarrow \infty} \sigma(\bar{x}(t)) = \frac{1}{N^2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} \lim_{t \rightarrow \infty} (\mathbb{E} \{x_i(t)x_j(t)\} - \mathbb{E} \{x_i(t)\} \mathbb{E} \{x_j(t)\}) \quad (6.29)$$

$$= \frac{1}{N^2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} \left(\lim_{t \rightarrow \infty} \mathbb{E} \{x_i(t)x_j(t)\} - \lim_{t \rightarrow \infty} \mathbb{E} \{x_i(t)\} \mathbb{E} \{x_j(t)\} \right) \quad (6.30)$$

where $\sigma(\cdot)$ denotes the variance of its argument, and the second equality follows from the fact that individual limits exists. We first note that $\lim_{t \rightarrow \infty} \mathbb{E} \{x_i(t)\}$ exists and equal to p_i , $\forall i \in \mathcal{V}$ by Lemma 34. In the following we will show that $\lim_{t \rightarrow \infty} \mathbb{E} \{x_i(t)x_j(t)\}$, $\forall i, j \in \mathcal{V}$ (thus the assumption given above is indeed true), and we will also characterize that particular term.

We note that since $x_i(t) \in \{0, 1\} \forall i \in \mathcal{V}, t \geq 0$, $x_i(t)$ is a binary random variable. For this reason,

$$\lim_{t \rightarrow \infty} \mathbb{E} \{x_i(t)x_j(t)\} = \lim_{t \rightarrow \infty} P(x_i(t) = x_j(t) = 1). \quad (6.31)$$

Therefore, this particular term is equal to the probability that node i 's and node j 's opinions are both equal to one in the limit. Due to the potential high correlation between node i 's and node j 's values (due to the attractive nature of the algorithm), one cannot estimate $\lim_{t \rightarrow \infty} P(x_i(t) = x_j(t) = 1)$ in terms of individual expectations. In the following, we will completely characterize these terms by using the dual approach.

As we have discussed in Section 6.2.3 and (6.6), for a given node pair i, j ,

$$\lim_{t \rightarrow \infty} (x_i(t), x_j(t)) = \lim_{t \rightarrow \infty} (x_{y_i^t(t-s)}(s), x_{y_j^t(t-s)}(s)),$$

where $s \in \mathbb{R}^+ \cup \{0\}$, and $\{y_i^t(t-s), y_j^t(t-s)\}$ is the dual coalescing random walk on \mathcal{G} which has been initialized at node i and node j . We note that by choosing

$s = 0$ and taking probabilities of each side,

$$\lim_{t \rightarrow \infty} P(x_i(t) = x_j(t) = 1) = \lim_{t \rightarrow \infty} P(x_{y_i^t}(0) = 1, x_{y_j^t}(0) = 1), \quad (6.32)$$

In other words, by using our discussion in Section 6.2.3, the probability that node i and node j 's opinions are both equal to one is equal to that the coalescing random walk initiated at node i and node j is absorbed by a stubborn node with opinion one.

To calculate such probability, we first construct a graph $\mathcal{G}'(\mathcal{V}', \mathcal{E}')$, where $\mathcal{V}' = \{(i, j) | i, j \in \mathcal{V}\}$. We denote each element of \mathcal{V}' as l_{ij} where the subscript (i, j) corresponds to the pairs in the original graph \mathcal{G} . We note that the cardinality of the set \mathcal{V}' is N^2 . Moreover, there exists an edge between two nodes l_{ij} and l_{mn} , if one of the followings hold:

- $i = m$, and there exists an edge between nodes j and n in the original graph \mathcal{G} , i.e., $(j, n) \in \mathcal{E}$.
- $j = n$, and there exists an edge between nodes i and m in the original graph \mathcal{G} , i.e., $(i, m) \in \mathcal{E}$.

On this particular graph \mathcal{G}' , we will define a biased random walk $Z'(k)$ as follows: For all $k \geq 0$,

$$P(Z'(k+1) = l_{mn} | Z'(k) = l_{ij})$$

$$= \begin{cases} \frac{1}{2|\mathcal{N}_i|} & \text{if } i \neq j, n = j, m \in \mathcal{N}_i, i \notin \{\mathcal{V}_0 \cup \mathcal{V}_1\}, \\ \frac{1}{2|\mathcal{N}_j|} & \text{if } i \neq j, m = i, n \in \mathcal{N}_j, j \notin \{\mathcal{V}_0 \cup \mathcal{V}_1\}, \\ \frac{1}{2} & \text{if } i \neq j, m = i, n = j, i \in \{\mathcal{V}_1 \cup \mathcal{V}_0\}, j \notin \{\mathcal{V}_1 \cup \mathcal{V}_0\}, \\ \frac{1}{2} & \text{if } i \neq j, m = i, n = j, j \in \{\mathcal{V}_1 \cup \mathcal{V}_0\}, i \notin \{\mathcal{V}_1 \cup \mathcal{V}_0\}, \\ \frac{1}{|\mathcal{N}_i|} & \text{if } m = n, i = j, m \in \mathcal{N}_i, \\ 1 & \text{if } m = i, n = j, (n, m) \in \{\mathcal{V}_1 \cup \mathcal{V}_0\}, \\ 0 & \text{otherwise.} \end{cases}$$

We note that the index k is a discrete index. The biased random walk is the equivalent to the coalescing walks $\lim_{t \rightarrow \infty} \{y_i^t(t), y_j^t(t)\}$. To see this, we observe that the transition probabilities of the process $Z'(k)$ is the simple average² of the transition rates of the individual processes $(y_i^t(t), y_j^t(t))$. Therefore, a jump in the process $Z'(k)$ corresponds to a jump in the original coalescing walk $(y_i^t(t), y_j^t(t))$. We also note that whenever both i, j are stubborn states or $i, j \in \mathcal{V}_0 \cup \mathcal{V}_1$, l_{ij} is an absorbing state. This, in return, implies that once $(y_i^t(t), y_j^t(t))$ is hits absorbing states, then the walk will terminate.

At this point, by combining our discussion above and (6.31), we conclude the following:

Lemma 35 *In the limit as $t, k \rightarrow \infty$,*

$$\lim_{t \rightarrow \infty} \mathbb{E} \{x_i(t)x_j(t)\} = \lim_{t \rightarrow \infty} P(x_i(t) = x_j(t) = 1) \quad (6.33)$$

$$= \sum_{m, n \in \mathcal{V}_1} \lim_{k \rightarrow \infty} P(Z'(k) = l_{mn} | Z'(0) = l_{ij}), \quad (6.34)$$

where $Z'(k)$ is a biased random walk whose transition probabilities are given in (6.33).

We note that by the lemma given above, we have characterized all of the terms

²The reasoning behind the averaging is that the continuous time processes $(y_i^t(t), y_j^t(t))$ have the same parameters.

in $\lim_{t \rightarrow \infty} \sigma(\bar{x}(t))$, thus completely identified variance of the average opinion in the limit.

We would like to conclude the section with the following discussion: While the expected value of the average opinion measures the dominant opinion in the network, it does not measure how much variation one should expect on the collective opinion of the society. Therefore, we believe that, our result in Lemma 35 is highly significant since it captures the variation. Another way to interpret the variance is in terms of the stability of the society's collective opinion. When the variance is small, the collective opinion will be highly stable and the expected value will be a relatively good measure for predicting society's opinion. On the other hand, when the variance is high, the collective opinion will be instable in the sense that it will oscillate, and it will hard to predict the true value of the collective opinion.

6.4 Optimal placement of stubborn agents

In this section, we will focus on a design problem, *i.e.*, optimal stubborn agent replacement problem. In particular, we assume that we are given a network \mathcal{G} and a set of stubborn agents of type zero \mathcal{V}_0 with known locations. Given $k > 0$, we would like to choose k nodes from $\mathcal{V} \setminus \mathcal{V}_0$ which will be assigned type one label, *i.e.*, \mathcal{V}_1 . The question of interest is: *How can one determine the optimal set of k nodes?* To be able answer this question, we first need to choose a measure of optimality. In the following, we will focus on the expected value of the average opinion in the network. In other words, we would like to choose a set of k nodes which will be assigned type 1 label (\mathcal{V}_1) such that $\sum_{i=1}^N \lim_{t \rightarrow \infty} \bar{x}(t)$ is maximized.

In the following we will first focus on the case where $k = 1$, and then introduce the general formulation.

6.4.1 A special case: $k=1$

In this special setting, we are allowed pick a single node as a stubborn agent of type 1 given $\mathcal{V}, \mathcal{V}_0$. We first remind our readers that since the average opinion of a given node i is equal to the probability that the random walk is absorbed by \mathcal{V}_1 given that the walk has been started at node i by Lemma 34. Denoting this particular probability as $p_{\mathcal{V}_1, i}$, our maximization problem can be written as:

$$\max_{m \in \mathcal{V} \setminus \mathcal{V}_0} \sum_{i \in \mathcal{V}} p_{m, i}. \quad (6.35)$$

Therefore, we are interested in m^* which maximizes the formulation given above.

Without loss of generality, we assume that $\{1, \dots, |\mathcal{V}_0|\}$ nodes are stubborn agents with type 0. Then, the probability transition matrix of the random walk on that particular graph can be written as:

$$T = \begin{bmatrix} I_{\mathcal{V}_0} & 0 \\ C & D \end{bmatrix}. \quad (6.36)$$

where $I_{\mathcal{V}_0}$ is $|\mathcal{V}_0| \times |\mathcal{V}_0|$ identity matrix. We note that the matrix $(I - D)^{-1}$ is called the fundamental matrix of the chain and $[(I - D)^{-1}]_{ij}$ is equal to the expected number visits to node j before absorption, given that it has been initiated in node i [84]. We note that $(I - D)^{-1}$ can be determined completely given \mathcal{V} and \mathcal{V}_0 , and is independent of the specific choice of \mathcal{V}_1 .

Without loss of generality, let's assume that we pick node $|\mathcal{V}_0| + 1$ as a candidate for our maximization problem. Assigning the type 1 label to this particular node, the transition probability matrix becomes:

$$\tilde{T} = \begin{bmatrix} I_{\mathcal{V}_0} & 0 & 0 \\ 0 & 1 & 0 \\ \tilde{C} & c & \tilde{D} \end{bmatrix}. \quad (6.37)$$

As we have discussed in the proof of Lemma 34, non-zero columns of $\lim_{k \rightarrow \infty} (\tilde{T})^k$ correspond to the absorption probabilities. In this case,

$$\lim_{k \rightarrow \infty} (\tilde{T})^k = \begin{bmatrix} I_{\mathcal{V}_0} & 0 & 0 \\ 0 & 1 & 0 \\ (I - \tilde{D})^{-1} \tilde{C} & (I - \tilde{D})^{-1} c & 0 \end{bmatrix}. \quad (6.38)$$

It should be clear from (6.38) that the sum of the absorption probabilities by node $|\mathcal{V}_0| + 1$ is simply,

$$1 + \mathbf{1}'(I - \tilde{D})^{-1}c. \quad (6.39)$$

In other words, if node $i = |\mathcal{V}_0| + 1$ is chosen as the stubborn agent with value one, then the objective function in (6.35) is equal to (6.39).

At this point, we would like to emphasize the relationship between D in (6.36) and \tilde{D} in (6.37). Indeed,

$$D = \begin{bmatrix} l & v' \\ c & \tilde{D} \end{bmatrix} \quad (6.40)$$

where l is a scalar quantity, where c and v' are column and row vectors respectively. Thus, by using matrix inversion in block form, the fundamental matrix of T which is equal to $(I - D)^{-1}$ can be written as:

$$(I - D)^{-1} = \begin{bmatrix} 1 - l & -v' \\ -c & I - \tilde{D} \end{bmatrix}^{-1} = \begin{bmatrix} \gamma & \dots \\ (I - \tilde{D})^{-1}c\gamma & \dots \end{bmatrix}, \quad (6.41)$$

where γ is a scalar value and is a function of l, v', c and D . Note that the first column sum (the column corresponding to the specific choice of type 1 agent) of the fundamental matrix is simply $\gamma(1 + (I - \tilde{D})^{-1}c)$. This is exactly γ times the value in (6.39), which is the value of the objective function of the optimization problem when node $i = |\mathcal{V}_0| + 1$ is chosen.

Using the argument above, we can state that, for $m \in \mathcal{V} \setminus \mathcal{V}_0$:

$$\sum_{i \in \mathcal{V}} p_{m,i} = \frac{\mathbf{1}^T [(I - D)^{-1}]_{m-|\mathcal{V}_0|}}{[(I - D)^{-1}]_{m-|\mathcal{V}_0|, m-|\mathcal{V}_0|}}, \quad (6.42)$$

where $[\cdot]_m$ denotes the m -th row of its argument, and $\mathbf{1}$ is the all ones column vector. This particular relation is interesting since the value of the objective function is dependent on the elements of the $(I - D)^{-1}$ matrix in (6.36) whose structure is independent of the particular choice m . However, the m dependence is introduced through the selecting certain rows of the matrix. For practical purposes, the fundamental matrix of the chain can be calculated once, and an exhaustive search can be conducted for determining optimum m . Since $|\mathcal{V}| = N$, the search requires at most N steps.

6.4.2 The general case: $k \geq 2$

For $k \geq 2$, the optimization problem becomes combinatorial. While this particular case is an ongoing research, a naive way to solve the problem will be utilizing a greedy algorithm. In other words, for a given $k = K$, one can solve the problem optimization problem incrementally by adding new nodes one by one. Then, by using sub-modular function theory, one can show that the solution to the greedy algorithm can not be $(1 - 1/e)$ worse than the optimum solution.

6.5 Discussions

In this chapter, we propose a model for capturing opinion propagation through social networks. We have employed modified voter model as well as stubborn agents to capture instability in society's collective opinion and correlations between individuals. We have shown that under mild assumptions society's opinions converge to a unique stationary distribution and we have completely characterized mean and variance of the society's average opinion in the long run. We have briefly posed optimal stubborn agent replacement problem and introduce our preliminary results.

APPENDIX A

APPENDIX OF CHAPTER 2

A.1 Proof of Lemma1

Define the mean-squared deviation from the current mean as:

$$\delta(t) = \sum_{i=1}^N \mathbb{E}\{\| z_i(t) - \frac{1}{N} \mathbf{1}^T z(t) \|^2\}. \quad (\text{A.1})$$

Since our definition of convergence is in the mean of order 2, the nodes reach a consensus if and only if $\delta(t) \rightarrow 0$. Define:

$$m(t) = z(t) - \frac{1}{N} \mathbf{1} \mathbf{1}^T z(t) = z(t) - Jz(t) = (I - J)z(t) \quad (\text{A.2})$$

$m(t)$ is the distances of the node values from their mean. We denote covariance matrix of $m(t)$ as, $\Theta(t) = \mathbb{E}\{m(t)m^T(t)\}$. It is obvious that $\delta(t) = \text{Trace}(\Theta(t))$.

Recursion structure of $m(t)$ can be written as:

$$m(t+1) = (W - J)m(t) + (I - J)v(t)$$

where we define $v(t) = \epsilon Aw(t)$. Then,

$$\Theta(t+1) = (W - J)\Theta(t)(W - J) + (I - J)\mathbb{E}\{v(t)v^T(t)\}(I - J). \quad (\text{A.3})$$

The remaining of the proof follows similarly to [21], but is given here for completeness. Forward: Assume the nodes converge to a consensus. Therefore $\delta(t) \rightarrow 0$. If $\delta(t) \rightarrow 0$, then $\Theta(t+1) \rightarrow \mathbf{0}$ since the diagonal entries of a covariance matrix are non-negative and $[\Theta(t)]_{ii}[\Theta(t)]_{jj} \geq [\Theta(t)]_{ij}^2 \forall \{i, j\}$.¹ By equa-

¹ $\mathbb{E}[X^2]\mathbb{E}[Y^2] \geq \mathbb{E}^2[XY]$.

tion (A.3):

$$\begin{aligned}
\mathbf{0} &= \lim_{t \rightarrow \infty} \Theta(t+1) = \lim_{t \rightarrow \infty} [(W - J)\Theta(t)(W - J) + (I - J)\mathbb{E}\{v(t)v^T(t)\}(I - J)] \\
&= (W - J)[\lim_{t \rightarrow \infty} \Theta(t)](W - J) + (I - J) \\
&+ (I - J)[\lim_{t \rightarrow \infty} \mathbb{E}\{v(t)v^T(t)\}](I - J) \tag{A.4}
\end{aligned}$$

$$= (I - J)[\lim_{t \rightarrow \infty} \mathbb{E}\{v(t)v^T(t)\}](I - J) \tag{A.5}$$

where in (A.4) we have used the fact that $(W - J)$ and $(I - J)$ are bounded, and in (A.5) we have used the fact that $\lim_{t \rightarrow \infty} \Theta(t) = \mathbf{0}$. Therefore, $\mathbb{E}\{v(t)v^T(t)\} = \epsilon^2 A \mathbb{E}\{w(t)w^T(t)\} A \rightarrow \mathbf{0}$. Since $\epsilon > 0$, $a_{ij} \geq 0 \forall i, j \in \{1, \dots, N\}$, and $\mathbb{E}\{w(t)w^T(t)\}$ is a diagonal matrix, each of the diagonal entries approaches 0, *i.e.* $\mathbb{E}\{w_i(t)\}^2 \rightarrow 0$.

Reverse: Assume the quantization noise at each sensor converges to 0. Thus, $\mathbb{E}\{w(t)w^T(t)\} \rightarrow \mathbf{0}$. In other words, given $(\delta > 0)$, \exists a large $(T > 0)$ such that $\forall t \geq T$, $\mathbb{E}\{w(t)w^T(t)\} < \delta I$ where I is the $(N \times N)$ identity matrix. Given $(\delta > 0)$, (A.3) is equal to:

$$\begin{aligned}
\Theta(t+1) &= (W - J)^{(t+1)}\Theta(0)(W - J)^{(t+1)} \\
&+ \sum_{j=0}^t (W - J)^{(t-j)}(I - J)\mathbb{E}\{v(j)v^T(j)\}(I - J)(W - J)^{(t-j)} \\
&= (W - J)^{(t+1)}\Theta(0)(W - J)^{(t+1)} \\
&+ \sum_{j=0}^T (W - J)^{(t-j)}(I - J)\mathbb{E}\{v(j)v^T(j)\}(I - J)(W - J)^{(t-j)} \tag{A.6}
\end{aligned}$$

$$+ \sum_{j=T+1}^t (W - J)^{(t-j)}(I - J)\mathbb{E}\{v(j)v^T(j)\}(I - J)(W - J)^{(t-j)}. \tag{A.7}$$

Since $\rho(W - J) < 1$ and T is finite for a given $\delta > 0$, then the two terms in (A.6) vanishes as $t \rightarrow \infty$. By using the fact that $\forall t \geq T$, $\mathbb{E}\{w(t)w^T(t)\} < \delta I$;

$$\lim_{t \rightarrow \infty} \Theta(t+1) < \lim_{t \rightarrow \infty} \delta \epsilon^2 \sum_{j=T+1}^t (W - J)^{(t-j)}(I - J)AA^T(I - J)(W - J)^{(t-j)}. \tag{A.8}$$

Since $[(I - J)AA^T(I - J)]_{ij}$ is finite $\forall \{i, j\}$ and $\rho(W - J) < 1$, righthand side of the above equation converges to $K < \infty$ where ($<$) denotes entry by entry inequality. Therefore:

$$\lim_{t \rightarrow \infty} \Theta(t + 1) < \delta \epsilon^2 K. \quad (\text{A.9})$$

As $\delta \rightarrow 0$, $\lim_{t \rightarrow \infty} \Theta(t + 1) \rightarrow \mathbf{0}$.

A.2 Calculation of $\mathbb{E}\{\tilde{z}(t - l)\tilde{z}^T(t - m)\}$

$\mathbb{E}\{\tilde{z}(t - l)\tilde{z}^T(t - m)\}$ is written in terms of state and noise covariances as:

$$\begin{aligned} \mathbb{E}\{\tilde{z}(t - l)\tilde{z}^T(t - m)\} &= \mathbb{E}\{z(t - l)z^T(t - m)\} + \mathbb{E}\{z(t - l)w^T(t - m)\} \\ &+ \mathbb{E}\{w(t - l)z^T(t - m)\} \\ &+ \mathbb{E}\{w(t - l)w^T(t - m)\}. \end{aligned} \quad (\text{A.10})$$

We also note that $z(t)$ vector can be written in terms of $z(t - 1 - q)$ as follows:

$$z(t) = W^{q+1}z(t - 1 - q) + \epsilon \sum_{j=0}^q W^j A w(t - 1 - j). \quad (\text{A.11})$$

We evaluate each term in (A.10) independently in order of appearance. W.L.O.G., we assume that $(l \leq m)$. If we focus on the first term in (A.10), by (A.11):

$$\begin{aligned} \mathbb{E}\{z(t - l)z^T(t - m)\} &= W^{m-l}\mathbb{E}\{z(t - m)z^T(t - m)\} \\ &+ \epsilon \sum_{j=0}^{m-l-1} W^j A \mathbb{E}\{w(t - l - 1 - j)z^T(t - m)\} \end{aligned} \quad (\text{A.12})$$

$$= W^{m-l}\mathbb{E}\{z(t - m)z^T(t - m)\}. \quad (\text{A.13})$$

We note that each element of the summation term in (A.12) is zero since the vector $z(t - m)$ is uncorrelated with the future noise vectors. The second term

in (A.10) is:

$$\begin{aligned}
\mathbb{E}\{z(t-l)w^T(t-m)\} &= \mathbb{E}\{z(t-m+(m-l))w^T(t-m)\} \\
&= W^{m-l}\mathbb{E}\{z(t-m)w^T(t-m)\} \\
&+ \epsilon \sum_{j=0}^{m-l-1} W^j A \mathbb{E}\{w(t-l-1-j)w^T(t-m)\} \\
&= \epsilon W^{m-l-1} A \mathbb{E}\{w(t-m)w^T(t-m)\}.
\end{aligned}$$

We use the fact that $z(t-m)$ is uncorrelated of the noise $w(t-m)$, and noise vectors are spatially uncorrelated. By the same token, the third term in (A.10) is:

$$\mathbb{E}\{w(t-l)z^T(t-m)\} = 0.$$

The last term, which is the covariance of the error terms is zero unless $l = m$. Combining each term given above, we verify the results given in Table 2.1.

A.3 Calculation of $\mathbb{E}\{z(t)\tilde{z}^T(t-m)\}$

In this section, we calculate the correlation between the state vector at time (t), and the noise vector at time ($t-m$), $m \geq 1$. By using the equations derived in Appendix A.2 and setting $l = 0$ and $m \geq 1$:

$$\begin{aligned}
\mathbb{E}\{z(t)\tilde{z}^T(t-m)\} &= \mathbb{E}\{z(t)(z^T(t-m) + w^T(t-m))\} \\
&= W^m \mathbb{E}\{z(t-m)z^T(t-m)\} \\
&+ \epsilon W^{m-1} A \mathbb{E}\{w(t-m)w^T(t-m)\}.
\end{aligned}$$

APPENDIX B

APPENDIX OF CHAPTER 3

B.1 Proof of Lemma 9

We first note that by Lemma 7, the network will reach to a consensus if and only if $\mathbb{E}\{w_i^2(t)\} \rightarrow 0$ as $t \rightarrow \infty \forall i \in \mathcal{V}$, *i.e.*, quantization noise variances at each sensor converges to zero. In our special case where the network has a regular structure, we have already discussed that $\mathbb{E}\{w_i^2(t)\} = \mathbb{E}\{w_j^2(t)\}, \forall i, j \in \mathcal{V}$, and these quantities are also equal to $v(t)$ in (3.16). Therefore, the network will reach to a consensus if and only if $v(t)$ approaches zero in the limit.

Without loss of generality, we assume that $\omega_i^2 > 0, \forall i \in \mathcal{V}$. We partition the matrix H (3.16) as:

$$H = \begin{bmatrix} d & b^T \\ c & U \end{bmatrix}, \quad (\text{B.1})$$

where,

$$U = \begin{bmatrix} \omega_2^2 & 0 & \dots & \omega_2^2 \\ 0 & \omega_3^2 & \vdots & \omega_3^2 \\ 0 & \dots & \omega_N^2 & \omega_N^2 \\ \frac{C_K(\omega_2-1)^2}{N2^{2R}} & \dots & \dots & \frac{C_K \sum_{j=2}^N (\omega_j-1)^2}{N2^{2R}} \end{bmatrix}, b^T = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}^T, c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (\text{B.2})$$

$$d = 1. \quad (\text{B.3})$$

We first note that the eigenvalues of H is the union of the eigenvalues of d and U [44]. Moreover, the partition U is an irreducible non-negative matrix. The following lemma is due to Horn *et.al.* [44].

Lemma 36 Given a non-negative square matrix $U \in \mathbb{R}^{N \times N}$, and a non-negative vector $z \in \mathbb{R}^N \geq 0, z \neq 0$, if there exists $\alpha \geq 0$ such that:

$$Uz < \alpha z,$$

then, the largest eigenvalue of U in magnitude is strictly less than α .

In the following, we will try to construct a z vector such that $Uz < z$, and thus show that the largest eigenvalue of U (in magnitude) is strictly less than one.

We construct z as follows:

$$z_i = \begin{cases} \frac{1}{\omega_{i+1}^2} & \text{if } i \in \{1, \dots, N-1\} \\ \min_{j \in \mathcal{V}'} \frac{1-\omega_j^2}{\omega_j^4} & \text{if } i = N, \end{cases} \quad (\text{B.4})$$

where $\mathcal{V}' = \{2, \dots, N\}$. With some straightforward algebra and assuming that

$$R > \log \left(C_K \frac{\max_{i \in \mathcal{V}'} (\omega_i - 1)^2 \max_{i \in \mathcal{V}'} \omega_i^4}{\min_{i \in \mathcal{V}'} \omega_i^2 \min_{i \in \mathcal{V}'} 1 - \omega_i^2} + 4 \right), \quad (\text{B.5})$$

one can show that $Hz < z$. Therefore, by Lemma 36, all of the eigenvalues of the partition U is strictly less than one under a mild condition on the quantization rate R . Therefore, H matrix has a unique eigenvalue with value one and the rest are strictly less than one (B.1).

At this point, we note that the eigenvector corresponding to the eigenvalue one is equal to $[1 \ 0 \ 0 \ \dots \ 0]^T$. Therefore, $\lim_{t \rightarrow \infty} H^t$ has all rows equal to zeros except the first row. Since the last row of the matrix converges to all zeros and $\lim_{t \rightarrow \infty} v(t) = [\lim_{t \rightarrow \infty} H^t]_{N+1} Y(0)$ by (3.15), then $\lim_{t \rightarrow \infty} v(t) = 0$.

Finally, we note that if there exists an index i such that $\omega_i^2 = 0$, z_i can be taken as zero, and node i can be removed from the set \mathcal{V}' .

B.2 Proof Lemma 10

Without loss of generality, we will assume that $\omega_i^2 > 0, \forall i \in \mathcal{V}$. We first note that by (3.15),(3.16) and (3.21):

$$\delta_\infty = \lim_{t \rightarrow \infty} \frac{1}{N} \sum_{l=0}^{t-1} v(t) = \frac{1}{N} \left[\left(\sum_{t=0}^{\infty} H^t \right) Y(0) \right]_{N+1}. \quad (\text{B.6})$$

A careful analysis shows that for a given $t \geq 1$ and $1 \leq j \leq N$, the j -th entry of the last row of H^t is:

$$[H^t]_{N+1,j} = \frac{C_K(1 - \omega_j)^2 \omega_j^{2(t-1)}}{N2^{2R}} + O\left(\frac{1}{(2^{2R}N)^2}\right). \quad (\text{B.7})$$

For sufficiently large N , infinite summation of the noise variances are equal to:

$$\begin{aligned} \sum_{t=0}^{\infty} v(t) = \frac{C_K}{N2^{2R}} & \left\{ \sum_{j=1}^N \left((1 - \omega_j)^2 \sum_{t=0}^{\infty} \omega_j^{2t} \right) [Y(0)]_j \right. \\ & \left. + \left(\sum_{j=1}^N (1 - \omega_j)^2 \sum_{t=0}^{\infty} \omega_j^{2t} + 1 \right) [Y(0)]_{N+1} \right\}. \end{aligned} \quad (\text{B.8})$$

Recall that

$$[Y(0)]_j = \sigma_j(0); \quad [Y(0)]_{N+1} = v(0); \quad v(0) = \frac{C_K}{N2^{2R}} \sum_{j=1}^N \sigma_j(0). \quad (\text{B.9})$$

One can note that the second summation term in (B.8) is $(O(2^{-4R}))$ and negligible with respect to the first term $(O(2^{-2R}))$ for sufficiently large R . Hence, assuming $\omega_j \neq 0, \forall j \geq 2$ and using infinite geometric series equality, (B.8) reduces to:

$$\sum_{t=0}^{\infty} v(t) = \frac{C_K}{N2^{2R}} \left(\sum_{j=2}^N \frac{(1 - \omega_j)^2}{1 - \omega_j^2} \sigma_j(0) \right). \quad (\text{B.10})$$

B.3 Proof of Corollary 2

Note that $\sigma_j(0) \geq 0, \forall j$ and $h(u)/g(u) \leq (\sup h(u))/g(u)$ for positive function $g(\cdot)$ where $u \in \mathcal{U}$ with \mathcal{U} denoting the range of u . Having these in mind, note that

$|\omega_j| < 1 \Rightarrow 1 - \omega_j^2 > 0, \forall j \geq 2$ and observe the following set of inequalities to arrive at the claimed bound

$$\delta_\infty = \frac{C_K}{N^2 2^{2R}} \left(\sum_{j \in \mathcal{V}'} \frac{(1 - \omega_j)^2}{1 - \omega_j^2} \sigma_j(0) \right) \quad (\text{B.11})$$

$$<^{(a)} \frac{4C_K}{N^2 2^{2R}} \left(\sum_{j \in \mathcal{V}'} \frac{1}{1 - \omega_j^2} \sigma_j(0) \right) \quad (\text{B.12})$$

$$\leq^{(b)} \frac{4C_K}{N^2 2^{2R}} \frac{1}{1 - \omega_2^2} \sum_{j \in \mathcal{V}'} \sigma_j(0) \quad (\text{B.13})$$

$$<^{(c)} \frac{4C_K}{N^2 2^R} \frac{1}{1 - \omega_2^2} \max_{j \neq 1} \sigma_j(0) \quad (\text{B.14})$$

where (a) and (b) follow from the fact that $|\omega_j| < 1 \Rightarrow |1 - \omega_j|^2 < 4, \forall j \geq 2$, and that $\omega_2 \geq |\omega_j|$, respectively. Finally, in (c), we simply note that $(N - 1)/N < 1$.

B.4 Proof Lemma 11

First we characterize the second largest eigenvalue of the connectivity matrix of a k -regular graph. Define A_1 as the $(0, 1)$ adjacency matrix of a $1 - D$ regular graph with \sqrt{N} vertices where two vertices are connected if distance between them is less than k/\sqrt{N} . Define $B = I_{\sqrt{N}} + A_1$ where $I_{\sqrt{N}}$ is the $(\sqrt{N} \times \sqrt{N})$ identity matrix. The eigenvalues of B matrix can be represented as a function of the eigenvalues of A_1 matrix as:

$$\lambda_i(B) = \lambda_i(A_1) + 1. \quad (\text{B.15})$$

We can write adjacency matrix of the two-dimensional graph with connectivity rule k/\sqrt{N} as:

$$A_2 = B \otimes B - I. \quad (\text{B.16})$$

The eigenvalues of A_2 are given for $1 \leq i, j \leq \sqrt{N}$:

$$\lambda_{ij}(A_2) = \lambda_i(B)\lambda_j(B) - 1 = (\lambda_i(A_1) + 1)(\lambda_j(A_1) + 1) - 1 \quad (\text{B.17})$$

$$= \left(1 + 2 \sum_{l=1}^k \cos(2\pi l i/\sqrt{N})\right) \left(1 + 2 \sum_{l=1}^k \cos(2\pi l j/\sqrt{N})\right) - 1. \quad (\text{B.18})$$

Given A_2 , the connectivity matrix of the network is equal to:

$$W = I_N - \epsilon(\text{diag}(A_2) - A_2). \quad (\text{B.19})$$

Then, the eigenvalues of the connectivity matrix can be written as:

$$\lambda_{ij}(W) = 1 - \epsilon(4k(k+1) - \lambda_{ij}) \quad (\text{B.20})$$

$$= 1 - \epsilon 4k(k+1) \quad (\text{B.21})$$

$$- \epsilon \left(1 + 2 \sum_{l=1}^k \cos(2\pi l i/\sqrt{N})\right) \left(1 + 2 \sum_{l=1}^k \cos(2\pi l j/\sqrt{N})\right) + \epsilon. \quad (\text{B.22})$$

It is easy to check from the properties of $\cos(\cdot)$ function that the above equation attains its maximum when $(i = \sqrt{N}, j = \sqrt{N})$, moreover the second largest value for $(i = \sqrt{N}, j = \sqrt{N} - 1)$ or $(i = \sqrt{N} - 1, j = \sqrt{N})$. The second largest eigenvalue, thus, is given by

$$\lambda_2(W) = 1 - \epsilon 4k(k+1) \quad (\text{B.23})$$

$$- \epsilon \left(1 + 2 \sum_{l=1}^k \cos(2\pi l)\right) \left(1 + 2 \sum_{l=1}^k \cos(2\pi l (\sqrt{N} - 1)/\sqrt{N})\right) + \epsilon \quad (\text{B.24})$$

$$= 1 - \epsilon \left(4k(k+1) - (1 + 2k) \left(1 + 2 \sum_{l=1}^k \cos(2\pi l (\sqrt{N} - 1)/\sqrt{N})\right) + 1\right) \quad (\text{B.25})$$

$$= 1 - \epsilon \left(4k(k+1) - \left(2k + 2(2k+1) \sum_{l=1}^k \cos\left(2\pi l/\sqrt{N}\right)\right)\right) \quad (\text{B.26})$$

$$= 1 - \epsilon \left[2(2k+1) \left(k + \sum_{l=1}^k \cos\left(2\pi l/\sqrt{N}\right)\right)\right]. \quad (\text{B.27})$$

In the above, $4k(k+1)$ is the number of neighbors for a given $k \geq 1$ and $0 < \epsilon \leq (4k(k+1))^{-1}$. Note that for $k \ll \sqrt{N}$, using Taylor expansion for the $\cos(\cdot)$ function, we find that $\omega_2 = 1 - \mathcal{O}(k^2/N)$ indicating that $\delta_\infty = \mathcal{O}(1/k^2)$ by (3.23).

B.5 Proof of Equation (3.30)

Without loss of generality we will assume that $\omega_j^2 > 0, j \in \mathcal{V}$. Using equations (3.13) and (3.14), we express noise variances in terms of current state variances as:

$$v(t) = \frac{C_K}{N2^{2R(t)}} \sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2}{\omega_j^2} \sigma_j(t). \quad (\text{B.28})$$

Then, with some algebra ratio of two consecutive noise variances is:

$$\beta = \frac{v(t+1)}{v(t)} = 2^{-2R(t+1)} \left(2^{2R(t)} \frac{\sum_{j=2, \omega_j \neq 0}^N (\omega_j - 1)^2 \sigma_j(t)}{\sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2}{\omega_j^2} \sigma_j(t)} + \frac{C_K}{N} \sum_{j=2}^N (\omega_j - 1)^2 \right). \quad (\text{B.29})$$

This implies that quantization rates satisfying bounded convergence follow the non-linear recursion:

$$2^{2R(t+1)} = \frac{1}{\beta} \left(2^{2R(t)} \frac{\sum_{j=2}^N (\omega_j - 1)^2 \sigma_j(t)}{\sum_{j=2}^N \frac{(\omega_j - 1)^2}{\omega_j^2} \sigma_j(t)} + \frac{C_K}{N} \sum_{j=2}^N (\omega_j - 1)^2 \right) \Rightarrow \quad (\text{B.30})$$

$$R(t+1) = \frac{1}{2} \log \left(2^{2R(t)} \frac{\sum_{j=2, \omega_j \neq 0}^N (\omega_j - 1)^2 \sigma_j(t)}{\sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2}{\omega_j^2} \sigma_j(k)} + \frac{C_K}{N} \sum_{j=2}^N (\omega_j - 1)^2 \right) \quad (\text{B.31})$$

$$- \frac{1}{2} \log \beta. \quad (\text{B.32})$$

B.6 Proof Lemma 12

If we take the limit of both sides of (3.30):

$$\lim_{t \rightarrow \infty} R(t+1) = \lim_{t \rightarrow \infty} \left[\frac{1}{2} \log \left(2^{2R(t)} \frac{\sum_{j=2, \omega_j \neq 0}^N (\omega_j - 1)^2 \sigma_j(t)}{\sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2}{\omega_j^2} \sigma_j(t)} + \frac{C_K}{N} \sum_{j=2}^N (\omega_j - 1)^2 \right) \right] \quad (\text{B.33})$$

$$- \frac{1}{2} \log \beta \quad (\text{B.34})$$

$$= \frac{1}{2} \log \left(\lim_{t \rightarrow \infty} \left[2^{2R(t)} \frac{\sum_{j=2, \omega_j \neq 0}^N (\omega_j - 1)^2 \sigma_j(t)}{\sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2}{\omega_j^2} \sigma_j(t)} \right] + \frac{C_K}{N} \sum_{j=2}^N (\omega_j - 1)^2 \right) \quad (\text{B.35})$$

$$- \frac{1}{2} \log \beta. \quad (\text{B.36})$$

Since the rest of the terms are constants and $\log\{\cdot\}$ is continuous function, we focus on:

$$\lim_{t \rightarrow \infty} \left[2^{2R(t)} \frac{\sum_{j=2, \omega_j \neq 0}^N (\omega_j - 1)^2 \sigma_j(t)}{\sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2}{\omega_j^2} \sigma_j(t)} \right] = 2^{\lim_{t \rightarrow \infty} 2R(t)} \lim_{t \rightarrow \infty} \frac{\sum_{j=2, \omega_j \neq 0}^N (\omega_j - 1)^2 \sigma_j(t)}{\sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2}{\omega_j^2} \sigma_j(t)}. \quad (\text{B.37})$$

Assuming both limits exist on the left hand side of the equality, the limit of multiplication can be written as multiplication of the limits. We denote the limit of the rates as R^* , i.e $2^{\lim_{t \rightarrow \infty} 2R(t)} = 2^{2R^*}$. To calculate the second term, we make the following observation in (3.25):

$$\frac{\sigma_j(t)}{\omega_j^2} - \sigma_j(t-1) = \beta^{t-1} v(0) \quad \forall \omega_j \neq 0. \quad (\text{B.38})$$

Then by (B.38):

$$\frac{\sigma_j(t)}{\omega_j^2} - \sigma_j(t-1) = \beta \left(\frac{\sigma_j(t-1)}{\omega_j^2} - \sigma_j(t-2) \right). \quad (\text{B.39})$$

Therefore, state covariance matrix eigenvalues follow a second order differential equation:

$$\sigma_j(t) - (\beta + \omega_j^2)\sigma_j(t-1) + \omega_j^2\beta\sigma_j(t-2) = 0 \quad (\text{B.40})$$

which has the following solution given $(\beta \neq \omega_j^2 \ \forall j \in \{2, \dots, N\})^1$

$$\sigma_j(t) = A_j(\omega_j^2)^t + B_j(\beta)^t. \quad (\text{B.41})$$

A and B are calculated using the initial conditions:

$$A_j = \sigma_j(0) - \frac{\omega_j^2 v(0)}{\beta - \omega_j^2} \quad B_j = \frac{\omega_j^2 v(0)}{\beta - \omega_j^2}. \quad (\text{B.42})$$

We note that the second term in (B.37) has two distinct cases:

$$K^* = \lim_{t \rightarrow \infty} \frac{\sum_{j=2, \omega_j \neq 0}^N (\omega_j - 1)^2 \sigma_j(t)}{\sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2}{\omega_j^2} \sigma_j(t)} = \left\{ \begin{array}{ll} \frac{\sum_{j=2, \omega_j \neq 0}^N (\omega_j - 1)^2 B_j}{\sum_{j=2, \omega_j \neq 0}^N \frac{(\omega_j - 1)^2}{\omega_j^2} B_j} & \beta > \max_j \omega_j^2 \\ \max_j \omega_j^2 & \beta < \max_j \omega_j^2 \end{array} \right\}. \quad (\text{B.43})$$

Then by plugging (B.43) into (B.36), asymptotic rate is:

$$\lim_{t \rightarrow \infty} R(t) = \frac{1}{2} \log \left(\frac{C_K}{N(\beta - K^*)} \sum_{j=2}^N (\omega_j - 1)^2 \right). \quad (\text{B.44})$$

Moreover, $(\beta - K^* > 0)$ is a necessary constraint since $(R(t) > 0 \ \forall t)$. Therefore $(\beta < \max_j \omega_j^2)$ is not a valid choice for β .

B.7 Proof of Lemma 13

To prove that R^* is decreasing in the region of interest, we need to show $\beta - K^*$ is an increasing function of β . In other words, we simply show $Q'(\beta) > 0$ in this

¹If $\beta = \omega_j^2$ for some j , then the solution is simply in the form of $\sigma_j(t) = A_j(\omega_j^2)^t + B_j t(\beta)^t$.

region where $Q(\beta) = \beta - K^*$. To simplify the presentation, let us introduce the following notation:

$$N(\beta, \omega) = \sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2 \omega_j^2}{\beta - \omega_j^2} \quad (\text{B.45})$$

and

$$D(\beta, \omega) = \sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2}{\beta - \omega_j^2}. \quad (\text{B.46})$$

Now, one can check that we have

$$Q'(\beta) = 1 - \frac{N'(\beta, \omega)D(\beta, \omega) - D'(\beta, \omega)N(\beta, \omega)}{D^2(\beta, \omega)} \quad (\text{B.47})$$

where we denote $N'(\beta, \omega) = \partial N(\beta, \omega)/\partial \beta$ and $D'(\beta, \omega) = \partial D(\beta, \omega)/\partial \beta$. Since we need to have $Q'(\beta) > 0$, consider the followings:

$$1 - \frac{N'(\beta, \omega)D(\beta, \omega) - D'(\beta, \omega)N(\beta, \omega)}{D^2(\beta, \omega)} > 0 \quad (\text{B.48})$$

$$\Rightarrow \frac{D^2(\beta, \omega) - N'(\beta, \omega)D(\beta, \omega) + D'(\beta, \omega)N(\beta, \omega)}{D^2(\beta, \omega)} > 0 \quad (\text{B.49})$$

$$\Rightarrow D^2(\beta, \omega) - N'(\beta, \omega)D(\beta, \omega) + D'(\beta, \omega)N(\beta, \omega) > 0 \quad (\text{B.50})$$

where the last line follows from the fact that $D^2(\beta, \omega) > 0$. Rearranging the term, we see that the above is satisfied if $D(\beta, \omega)(D(\beta, \omega) - N'(\beta, \omega)) > -D'(\beta, \omega)N(\beta, \omega)$. Now consider the followings:

$$D(\beta, \omega)(D(\beta, \omega) - N'(\beta, \omega)) = D(\beta, \omega) \left(\sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2}{\beta - \omega_j^2} + \sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2 \omega_j^2}{(\beta - \omega_j^2)^2} \right) \quad (\text{B.51})$$

$$= D(\beta, \omega) \sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2 \beta}{(\beta - \omega_j^2)^2} = -\beta D(\beta, \omega)D'(\beta, \omega). \quad (\text{B.52})$$

Since, $\beta > \omega_j^2$ for all $j \in \{2, \dots, N\}$, it is clear that:

$$\beta D(\beta, \omega) = \sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2 \beta}{\beta - \omega_j^2} > \sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2 \omega_j^2}{\beta - \omega_j^2} = N(\beta, \omega). \quad (\text{B.53})$$

Since $-D'(\beta, \omega) > 0$, it follows:

$$-D(\beta, \omega)D'(\beta, \omega) > -\beta N(\beta, \omega)D'(\beta, \omega). \quad (\text{B.54})$$

Therefore, we have $Q'(\beta) > 0$, as claimed by the Lemma.

B.8 Proof of Lemma 14

Let $\mathcal{R}(\beta_0) = [\omega_2^2 + \epsilon, 1 - \epsilon]$ denote the β_0 range of interest. Note that $Q(\beta)$ is defined and the limit exists for all $\beta_0 \in \mathcal{R}(\beta_0)$, *i.e.* $\lim_{\beta_0 \in \mathcal{R}(\beta_0) \rightarrow \beta} Q(\beta_0) = Q(\beta)$, thus, $Q(\beta_0)$ is continuous in $\mathcal{R}(\beta_0)$. Moreover, $Q(\beta_0)$ is an increasing function in $\mathcal{R}(\beta_0)$ by Lemma 13. Hence, it suffices to find the necessary conditions to satisfy $Q(\beta_0) < 0$ for $\beta_0 = \omega_2 + \epsilon$ and $Q(\beta_0) > 0$ for $\beta_0 = 1 - \epsilon$ to show that there is a β such that $\omega_2^2 + \epsilon < \beta < 1 - \epsilon$ and $Q(\beta) = 0$.

Let us consider $Q(\beta_0|_{\beta_0=\omega_2^2+\epsilon}) = Q(\omega_2^2 + \epsilon)$, for small $\epsilon > 0$,

$$Q(\omega_2^2 + \epsilon) = \omega_2^2 + \epsilon - K(\omega) - \frac{\sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2 \omega_j^2}{\omega_2^2 + \epsilon - \omega_j^2}}{\sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2}{\omega_2^2 + \epsilon - \omega_j^2}}. \quad (\text{B.55})$$

After some algebraic manipulations, the above reduces to

$$Q(\omega_2^2 + \epsilon) = \frac{\sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2}{\omega_2^2 + \epsilon - \omega_j^2} (\omega_2^2 + \epsilon - K(\omega) - \omega_j^2)}{\sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2}{\omega_2^2 + \epsilon - \omega_j^2}}. \quad (\text{B.56})$$

Since $g(u)/h(u) < 0 \Rightarrow g(u) < 0$ for $h(u) > 0$, we need

$$\sum_{j=2, \omega_j \neq 0}^N \frac{(1 - \omega_j)^2}{\omega_2^2 + \epsilon - \omega_j^2} (\omega_2^2 + \epsilon - K(\omega) - \omega_j^2) < 0 \Rightarrow Q(\omega_2^2 + \epsilon) < 0. \quad (\text{B.57})$$

Note that $(1 - \omega_j)^2 / (\omega_2^2 - \omega_j^2 + \epsilon) > 0$, for all $j = 2, 3, \dots, N$ since $|\omega_2| > |\omega_j|$ and $|\omega_j| < 1$ for all $j = 2, 3, \dots, N$. Thus, a sufficient condition for $Q(\omega_2^2 + \epsilon) < 0$ is given by

$$\omega_j^2 > \omega_2^2 + \epsilon - K(\omega) \Rightarrow Q(\omega_2^2 + \epsilon) < 0 \quad (\text{B.58})$$

for $j = 2, 3, \dots, N$. Similarly, considering $Q(\beta_0 |_{\beta_0=1-\epsilon}) = Q(1 - \epsilon)$ and the case $Q(1 - \epsilon) > 0$ shows that a sufficient condition for this case, using steps similar to above, yields

$$\omega_j^2 < 1 - \epsilon - K(\omega) \Rightarrow Q(1 - \epsilon) > 0 \quad (\text{B.59})$$

for $j = 2, 3, \dots, N$. Now, combining (B.58) and (B.59), and recalling that $Q(\beta_0)$ is continuous in $\mathcal{R}(\beta_0)$ gives the sufficient conditions to have at least one root in $\mathcal{R}(\beta_0)$ such that $Q(\beta) = 0$ where $\beta \in \mathcal{R}(\beta_0)$. Since $Q(\beta)$ is an increasing function in $\mathcal{R}(\beta_0)$, the root is unique.

B.9 Proof of Lemma 15

In the sequel, we assume that β is such that $\lim_{k \rightarrow \infty} R(k) = 0$. All the summation are over non-zero ω_j 's, they are omitted from the summation in order not to have a cumbersome notation. In addition, let us denote

$$K(k) \triangleq \frac{\sum_{j=2}^N (\omega_j - 1)^2 \sigma_j(k)}{\sum_{j=2}^N \frac{(\omega_j - 1)^2}{\omega_j^2} \sigma_j(k)} \quad (\text{B.60})$$

and $\epsilon(k) \triangleq K(k) - K^*$. Substituting $\sigma_j(k)$ and K^* with their equivalent terms as given in Appendix B.8, we obtain the following,

$$\epsilon(k) = \frac{\sum_{j=2}^N A'_j (\omega_j^2)^k + B'_j \beta^k}{\sum_{j=2}^N A''_j (\omega_j^2)^k + B''_j \beta^k} - \frac{\sum_{j=2}^N B'_j}{\sum_{j=2}^N B''_j} \quad (\text{B.61})$$

where we denote $A'_j \triangleq (\omega_j - 1)^2 A_j$, $B'_j \triangleq (\omega_j - 1)^2 B_j$, $A''_j \triangleq (\omega_j - 1)^2 / (\omega_j^2) A_j$ and $B''_j \triangleq (\omega_j - 1)^2 / (\omega_j^2) B_j$. Moreover, let $B' \triangleq \sum_{j=2}^N B'_j$ and $B'' \triangleq \sum_{j=2}^N B''_j$. The

above, after some algebraic manipulations, reduces to

$$\epsilon(k) = \frac{\sum_{j=2}^N (B'' A'_j - B' A''_j) (\omega_j^2)^k}{(B'')^2 \beta^k + \sum_{j=2}^N B'' A''_j (\omega_j^2)^k} \quad (\text{B.62})$$

$$\leq \frac{(\omega_2^2)^k \sum_{j=2}^N (B'' A'_j - B' A''_j)}{(B'')^2 \beta^k + \sum_{j=2}^N B'' A''_j (\omega_j^2)^k} \quad (\text{B.63})$$

$$\leq \frac{(\omega_2^2)^k \sum_{j=2}^N (B'' A'_j - B' A''_j)}{(B'')^2 \beta^k + (\omega_N^2)^k \sum_{j=2}^N B'' A''_j}. \quad (\text{B.64})$$

Of note is that, we assume $B'' A'_j - B' A''_j > 0$ and $B'' A''_j > 0$ for all $j \in \{1, \dots, N\}$ in (B.63). Recall that $\beta > \omega_2^2$, thus, for large enough k , the second term in the denominator is negligible. Thus, we have:

$$\epsilon(k) \leq \frac{(\omega_2^2)^k \sum_{j=2}^N (B'' A'_j - B' A''_j)}{(B'')^2 \beta^k} \quad (\text{B.65})$$

$$= C \left(\frac{\omega_2^2}{\beta} \right)^k \quad (\text{B.66})$$

where $C \triangleq \sum_{j=2}^N (B'' A'_j - B' A''_j) / (B'')^2$. Since $\beta > \omega_2^2 \Rightarrow \omega_2^2 / \beta < 1$, $\epsilon(k)$ forms a geometric series. Thus, $K(k)$ achieves its limit K^* , at the worst case, with a geometric rate.

In the following, we utilize asymptotic behavior of $\epsilon(k)$ to prove the sum-rate is finite. For notational purposes, we denote:

$$M(k) \triangleq 2^{2R(k)} \quad (\text{B.67})$$

$$L \triangleq \frac{C}{N} \sum_{j=2}^N (\omega_j - 1)^2. \quad (\text{B.68})$$

Then, rate recursion given in (3.30) becomes:

$$M(k+1) = M(k) \frac{K(k)}{\beta} + \frac{L}{\beta} \quad (\text{B.69})$$

where $k \geq 1$. Since $\sum_{k=0}^{\infty} R(k) < \infty$ if and only if $\prod_{k=0}^{\infty} M(k) < \infty$, we focus on the latter identity. It is clear that $\lim_{k \rightarrow \infty} M(k) = 1$ and $\lim_{k \rightarrow \infty} K(k) = K^*$ (K^*

in (3.32) for the specific value of β . We will upper bound $M(k) - \lim_{k \rightarrow \infty} M(k)$ difference as follows:

$$M(k+1) - \lim_{k \rightarrow \infty} M(k) = M(k) \frac{K(k)}{\beta} + \frac{L}{\beta} - 1 \quad (\text{B.70})$$

$$= M(1) \prod_{j=1}^k \frac{K(j)}{\beta} + \frac{\beta - K^*}{\beta} \left(1 + \sum_{j=0}^{k-2} \prod_{i=0}^j \frac{K(k-i)}{\beta} \right) - 1 \quad (\text{B.71})$$

$$= M(1) \prod_{j=1}^k \frac{K(j)}{\beta} + \frac{\beta - K^*}{\beta} \left(\sum_{j=0}^{k-2} \left(\prod_{i=0}^j \frac{K(k-i)}{\beta} - \frac{K^{*j+1}}{\beta^{j+1}} \right) + \sum_{j=0}^{k-1} \left(\frac{K^*}{\beta} \right)^j \right) - 1 \quad (\text{B.72})$$

$$= M(1) \prod_{j=1}^k \frac{K(j)}{\beta} + \frac{\beta - K^*}{\beta} \left(\frac{1 - \left(\frac{K^*}{\beta} \right)^k}{1 - \frac{K^*}{\beta}} + \sum_{j=0}^{k-2} \left(\prod_{i=0}^j \frac{K(k-i)}{\beta} - \frac{K^{*j+1}}{\beta^{j+1}} \right) \right) - 1 \quad (\text{B.73})$$

$$= M(1) \prod_{j=1}^k \frac{K(j)}{\beta} + \frac{\beta - K^*}{\beta} \sum_{j=0}^{k-2} \left(\prod_{i=0}^j \frac{K(k-i)}{\beta} - \frac{K^{*j+1}}{\beta^{j+1}} \right) - \left(\frac{K^*}{\beta} \right)^k \quad (\text{B.74})$$

$$\leq M(1) \prod_{j=1}^k \frac{K(j)}{\beta} + \frac{\beta - K^*}{\beta} \sum_{j=0}^{k-2} \left(\prod_{i=0}^j \frac{K(k-i)}{\beta} - \frac{K^{*j+1}}{\beta^{j+1}} \right) \quad (\text{B.75})$$

$$\leq M(1) \prod_{j=1}^k \frac{K(j)}{\beta} + \eta \sum_{j=1}^k e^{-k} 2^k + \eta e^{-k} k^{-1/2} \quad (\text{B.76})$$

$$\leq M(1) \prod_{j=1}^k \frac{K(j)}{\beta} + \eta \left(\frac{e}{2} \right)^{-k} k + \eta k e^{k^2 - k/2} \quad (\text{B.77})$$

$$\leq M(1) \kappa \nu^k + \eta k \left(\frac{e}{2} \right)^{-k} + \eta k e^{k^2 - k/2} \quad (\text{B.78})$$

$$< \frac{1}{k} \quad (\text{B.79})$$

where η, κ some positive constants and $\nu \in (0, 1)$. (B.71) follows from the fact that $L = (\beta - K^*)/\beta$ and recursion on $M(k)$. In (B.72), we add and subtract a geometric series, and we utilize the geometric sum formula in (B.73). Of note is that $K^*/\beta < 1$. (B.75) follows the fact that $K^*/\beta > 0$ and, (B.76) follows since each summation term in (B.75) is upper bounded by $\eta \prod_{i=0}^j \epsilon(k-i)/\beta + \eta 2^k e^{-k}$ and $\epsilon(k)/\beta \leq e^{-k}$ for large k . In (B.78), we use the fact that $K^*/\beta < 1$, therefore

$\exists N > 0$ such that $K(k)/\beta < 1 \forall k > N$. It is clear that both terms in (B.78) are strictly less than $1/k$.

Therefore, each $M(k)$ term approaches to 1, its limit, faster than $1/k$ for all $k > N$ where $N > 0$ is a sufficiently large integer. It follows that $\prod_{k=0}^{\infty} M(k)$ is bounded. Thus, $\sum_{k=0}^{\infty} R(k) < \infty$ as claimed by the Lemma.

APPENDIX C

APPENDIX OF CHAPTER 4

C.1 Proof of Lemma 17

First we note that

$$W_{jk} = \frac{1}{N} \sum_{i=1}^N W_{jk}^{(i)}. \quad (\text{C.1})$$

Then, by (4.5), we have

$$W_{jk} = \begin{cases} 1 - \frac{|\mathcal{N}_j|}{N} + \frac{\gamma|\mathcal{N}_j|}{N} & k = j \\ 1 - \frac{1-\gamma}{N} & k \in \mathcal{N}_j \\ 0 & \text{elsewhere} \end{cases}. \quad (\text{C.2})$$

Therefore (4.9) follows. Note that (4.9) is the representation of the weight matrix in terms of graph Laplacian L , i.e., $W = I - \eta L$ where $L = \text{diag}\{\Phi \mathbf{1}\} - \Phi$ and $\eta = (1 - \gamma)/N$. Since $0 < \eta < 1/(N - 1)$ for all $\gamma \in (0, 1)$, W satisfies the conditions given in (4.10).

C.2 Proof of Theorem 2

At this stage of development, we just need to put the pieces together. First, we introduce a Lemma concerning convergence of random sequences that will prove useful to prove the above theorem.

Lemma 37 [77] *Consider a sequence of nonnegative random variables $\{V(t)\}_{t \geq 0}$ with $\mathbb{E}\{V(0)\} < \infty$. Let*

$$\mathbb{E}\{V(t+1)|V(t), \dots, V(1), V(0)\} \leq cV(t) \quad (\text{C.3})$$

where $0 < c < 1$. Then, $V(t)$ almost surely converges to zero, i.e.,

$$\Pr \left\{ \lim_{t \rightarrow \infty} V(t) = 0 \right\} = 1. \quad (\text{C.4})$$

We have almost sure convergence if $\mathbb{E}\{\|\beta(t+1)\|_2^2 | \beta(t), \beta(t-1), \dots, \beta(0)\} \leq c\|\beta(t)\|_2^2$, for some $0 < c < 1$ since

$$\Pr \left\{ \lim_{t \rightarrow \infty} x(t) = c\mathbf{1} \right\} = \Pr \left\{ \lim_{t \rightarrow \infty} \|\beta(t)\|_2^2 = 0 \right\}. \quad (\text{C.5})$$

However, given $\beta(t)$, we have, as can be seen in the proof of Lemma 18, that,

$$\mathbb{E}\{\|\beta(t+1)\|_2^2 | \beta(t), \beta(t-1), \dots, \beta(0)\} = \mathbb{E}\{\|\beta(t+1)\|_2^2 | \beta(t)\}. \quad (\text{C.6})$$

From the proof of Lemma 18 and Proposition 2, we know that

$$\mathbb{E}\{\|\beta(t+1)\|_2^2 | \beta(t)\} \leq \lambda_1(\mathbb{E}\{W(t)(I-J)W(t)\})\|\beta(t)\|_2^2 \quad (\text{C.7})$$

where $0 < \lambda_1(\mathbb{E}\{W(t)(I-J)W(t)\}) < 1$. Thus, using the Lemma regarding the convergence of nonnegative random sequences, we have that $\Pr \{\lim_{t \rightarrow \infty} \|\beta(t)\|_2^2 = 0\} = 1$, completing the proof.

C.3 Proof of Lemma 19

Let $d_j = |\mathcal{N}_j|$ be the degree of node j . From the per-node weight matrices, we obtain

$$\{W^{(i)T}W^{(i)}\}_{jk} = \begin{cases} 1 + d_i(1-\gamma)^2 & k = j = i \\ \gamma(1-\gamma) & k \in \mathcal{N}_i, j = i \\ \gamma^2 & j \in \mathcal{N}_i, k = j \\ \gamma(1-\gamma) & j \in \mathcal{N}_i, k = i \\ 1 & j \notin \mathcal{N}_i, k = j \\ 0 & \text{otherwise} \end{cases}, \quad (\text{C.8})$$

where $W^{(i)}$ denotes the weight matrix corresponding to the case where node i 's clock ticks. Therefore, the average is

$$W'_{jk} = \begin{cases} 1 - \frac{d_j}{N} (1 - (1 - \gamma)^2 - \gamma^2) & k = j \\ \frac{2\gamma(1-\gamma)}{N} & k \in \mathcal{N}_j \\ \frac{2\gamma(1-\gamma)}{N} & j \in \mathcal{N}_k \\ 0 & \text{otherwise} \end{cases} . \quad (\text{C.9})$$

Then, (4.24) follows. As we noted before, (4.24) is the representation of W' in terms of graph Laplacian $D - \Phi$. Since $0 < 2\gamma(1 - \gamma)/N < 1/N - 1$ for all γ , W' satisfies the properties given in (4.10).

Turning now to W'' , we first calculate

$$\{\mathbf{1}^T W^{(i)}\}_j = \begin{cases} 1 & j \notin \mathcal{N}_i \cup \{i\} \\ 1 + d_i(1 - \gamma) & j = i \\ \gamma & j \in \mathcal{N}_i \end{cases} \quad (\text{C.10})$$

Then we find that

$$(W^{(i)T} \mathbf{1} \mathbf{1}^T W^{(i)})_{jk} = \begin{cases} 1 & j, k \notin \mathcal{N}_i \cup \{i\} \\ (1 + d_i(1 - \gamma))^2 & j = k = i \\ \gamma^2 & j, k \in \mathcal{N}_i \\ \gamma & j \in \mathcal{N}_i, k \notin \mathcal{N}_i \text{ or } j \notin \mathcal{N}_i, k \in \mathcal{N}_i \\ \gamma(1 + d_i(1 - \gamma)) & j = i, k \in \mathcal{N}_i \text{ or } k = i, j \in \mathcal{N}_i \\ 1 + d_i(1 - \gamma) & j = i, k \notin \mathcal{N}_i \text{ or } k = i, j \notin \mathcal{N}_i \end{cases} \quad (\text{C.11})$$

Now we can take expectations. For each pair (j, k) we can calculate the expectation over i . Note that $(\Phi^2)_{jk}$ is the number of paths of length 2 from j to k or alternatively the number of i which are in the neighborhood of j and k . Consider first the case where $j \neq k$.

- There are $(\Phi^2)_{jk}$ values for i for which $j, k \in \mathcal{N}_i$.
- There are $d_j + d_k - 2(\Phi^2)_{jk}$ values for which $i \neq j, k$ and exactly one of j and k is in \mathcal{N}_i if j and k are not adjacent. There are $d_j + d_k - 2 - 2(\Phi^2)_{jk}$ values if j and k are adjacent.
- Note that $i = j$ once and $i = k$ once. Of the last two alternatives in (C.11), we have the former if $\Phi_{jk} = 1$ and the latter if $\Phi_{jk} = 0$.
- If j and k are not adjacent, there are $N - 2 - d_j - d_k + (\Phi^2)_{jk}$ values of i for which $j, k \notin \mathcal{N}_i \cup \{i\}$. If they are adjacent, there are $N - d_j - d_k + (\Phi^2)_{jk}$ such values

Then we have for $j \neq k$ that:

$$\begin{aligned}
N^2 W''_{jk} &= \gamma^2 (\Phi^2)_{jk} + \gamma (d_j + d_k - 2(\Phi^2)_{jk})(1 - \Phi_{jk}) + \gamma (d_j + d_k - 2 - 2(\Phi^2)_{jk}) \Phi_{jk} \\
&\quad + \gamma (2 + (1 - \gamma)(d_j + d_k)) \Phi_{jk} + (2 + (1 - \gamma)(d_j + d_k))(1 - \Phi_{jk}) \\
&\quad + (N - 2 - (d_j + d_k) + (\Phi^2)_{jk})(1 - \Phi_{jk}) + (N - d_j - d_k + (\Phi^2)_{jk}) \Phi_{jk} \\
&= ((1 - \gamma)^2 \Phi^2 + N^2 J - (1 - \gamma)^2 (D\Phi + \Phi D))_{jk} \tag{C.12}
\end{aligned}$$

The matrix above has the following values on the diagonal:

$$(1 - \gamma)^2 d_j + N = (NI + (1 - \gamma)^2 D)_{jj} . \tag{C.13}$$

If $j = k$ then for 1 value of i we have $i = j = k$, for d_j values of i we have $j, k \in \mathcal{N}_i$, and for $N - d_j - 1$ values of i we have $j, k \notin \mathcal{N}_i \cup \{i\}$. Thus on the diagonal we should have

$$N^2 W''_{jj} = (1 + (1 - \gamma)d_j)^2 + \gamma^2 d_j + (N - 1 - d_j) \tag{C.14}$$

$$= (NI + (1 - \gamma)^2 D^2 + (1 - \gamma)^2 D)_{jj} . \tag{C.15}$$

So we must add the correction term $(1 - \gamma)^2 D^2$ to (C.12) to get the correct matrix:

$$\begin{aligned} N^2 \mathbb{E} [W^T J W] &= (1 - \gamma)^2 \Phi^2 + N^2 J - (1 - \gamma)^2 (D\Phi + \Phi D) + (1 - \gamma)^2 D^2 \\ &= (1 - \gamma)^2 (D - \Phi)^2 + N^2 J. \end{aligned} \quad (\text{C.16})$$

C.4 Proof of Lemma 20

It is easy to see that $\alpha(t+1) = W(t)\alpha(t)$, yielding the following recursion for the second moment:

$$\mathbb{E}\{\alpha(t+1)^T \alpha(t+1) | \alpha(t)\} = \alpha(t)^T \mathbb{E}\{W(t)^T W(t)\} \alpha(t) \quad (\text{C.17})$$

$$= \alpha(t)^T W' \alpha(t) \quad (\text{C.18})$$

$$= y(t)^T \Lambda y(t), \quad (\text{C.19})$$

where we utilize the eigendecomposition of $W' = V \Lambda V^T$ and define $y(t) = V^T \alpha(t)$. Given $\alpha(t)$, we can find $y(t)$, so we have the following:

$$\mathbb{E}\{\alpha(t+1)^T \alpha(t+1) | \alpha(t)\} = \sum_{i=1}^N \lambda_i(W') |y_i(t)|^2 \quad (\text{C.20})$$

$$= |y_1(t)|^2 + \sum_{i=2}^N \lambda_i(W') |y_i(t)|^2 \quad (\text{C.21})$$

$$\begin{aligned} &= (1 - \lambda_2(W')) |y_1(t)|^2 \\ &+ \lambda_2(W') |y_1(t)|^2 + \sum_{i=2}^N \lambda_i(W') |y_i(t)|^2 \end{aligned} \quad (\text{C.22})$$

$$\leq (1 - \lambda_2(W')) |y_1(t)|^2 + \lambda_2(W') \sum_{i=1}^N |y_i(t)|^2 \quad (\text{C.23})$$

$$= (1 - \lambda_2(W')) |y_1(t)|^2 + \lambda_2(W') \|y(t)\|_2^2 \quad (\text{C.24})$$

$$= (1 - \lambda_2(W')) \|J\alpha(t)\|_2^2 + \lambda_2(W') \|\alpha(t)\|_2^2, \quad (\text{C.25})$$

where the last line follows from the facts that $\|y(t)\|_2^2 = y(t)^T y(t) = \alpha(t)^T V V^T \alpha(t) = \alpha(t)^T \alpha(t) = \|\alpha(t)\|_2^2$ due to unitary decomposition and $|y_1(t)|^2 = |v_1^T \alpha(t)|^2 = (N)^{-1} \alpha(t)^T \mathbf{1} \mathbf{1}^T \alpha(t) = \alpha(t)^T J \alpha(t) = \|J \alpha(t)\|_2^2$ due to the relation $v_1 = (\sqrt{N})^{-1} \mathbf{1}$. This concludes the proof of the first item.

Let us now consider the second item. Note that J is a paracontracting matrix with respect to ℓ_2 norm since its symmetric and all its eigenvalues are in $(-1, 1]$. Thus, we have

$$Jx \neq x \Leftrightarrow \|Jx\|_2^2 < \|x\|_2^2. \quad (\text{C.26})$$

Thus, if we can show that $J\alpha(t) = \alpha(t)$ if and only if $x(t) = c\mathbf{1}$ for some $c \in \mathbb{R}$, we are done. If $x(t) = c\mathbf{1}$, then,

$$J\alpha(t) = Jx(t) - Jx(0) = x(t) - Jx(0) = \alpha(t), \quad (\text{C.27})$$

where we used the facts that $J^2 = J$ and $Jx(t)|_{x(t)=c\mathbf{1}} = x(t)$. Now, if $\alpha(t) = J\alpha(t)$, then $\alpha_i(t) = (N)^{-1} \sum_{i=1}^N \alpha_i(t)$. Thus, $\alpha(t) = \bar{\alpha}(t)\mathbf{1}$. Since

$$x(t) = \alpha(t) + Jx(0) = \bar{\alpha}(t)\mathbf{1} + \bar{x}(0)\mathbf{1} = (\bar{\alpha}(t) + \bar{x}(0))\mathbf{1}, \quad (\text{C.28})$$

we are done. Therefore, the proof of the second item is complete.

C.5 Proof of Proposition 3

One can check that the following holds:

$$\mathbb{E}\{\|J\alpha(t)\|_2^2\} = \mathbb{E}\{\|\alpha(t)\|_2^2\} - \mathbb{E}\{\|\beta(t)\|_2^2\}. \quad (\text{C.29})$$

Substituting the above into the claim (i) of Lemma 20, we obtain

$$\mathbb{E}\{\|\alpha(t+1)\|_2^2\} \leq \mathbb{E}\{\|\alpha(t)\|_2^2\} - (1 - \lambda_2(W'))\mathbb{E}\{\|\beta(t)\|_2^2\}. \quad (\text{C.30})$$

To upper bound the above, we need to lower bound $\mathbb{E}\{\|\beta(t)\|_2^2\}$ term. Utilizing the Jensen's inequality, we have

$$\mathbb{E}\{\|\beta(t)\|_2^2\} \geq \|\mathbb{E}\{\beta(t)\}\|_2^2 \quad (\text{C.31})$$

$$= \|\mathbb{E}\{x(t) - Jx(t)\}\|_2^2 \quad (\text{C.32})$$

$$= \|(W^t - J)x(0)\|_2^2, \quad (\text{C.33})$$

where the last line follows from Lemma 17. Moreover, utilizing the properties of W , the above reduces to

$$\mathbb{E}\{\|\beta(t)\|_2^2\} \geq \|(W^t - J)\alpha(0)\|_2^2 \quad (\text{C.34})$$

$$= \alpha(0)^T (W^t - J)^T (W^t - J) \alpha(0) \quad (\text{C.35})$$

$$= \alpha(0)^T (W^{2t} - J) \alpha(0) \quad (\text{C.36})$$

$$= \alpha(0)^T (W - J)^{2t} \alpha(0). \quad (\text{C.37})$$

Note that $(W - J)$ is symmetric since W and J are symmetric. Thus, utilizing the unitary eigendecomposition, $(W - J) = U\Lambda U^T$, and defining $z(0) = U^T \alpha(0)$, we obtain the following:

$$\mathbb{E}\{\|\beta(t)\|_2^2\} \geq \alpha(0)^T U \Lambda^{2t} U^T \alpha(0) \quad (\text{C.38})$$

$$= z(0)^T \Lambda^{2t} z(0) \quad (\text{C.39})$$

$$= \sum_{i=1}^N \lambda_i^{2t} (W - J) |z_i(0)|^2. \quad (\text{C.40})$$

We need the following lemma before we continue the proof.

Lemma 38 *All the eigenvalues of $W - J$ except zero, is lower bounded by γ , i.e.,*

$$\lambda_i(W - J) \geq \gamma \quad (\text{C.41})$$

for all $i \in \{1, 2, \dots, N - 1\}$.

Proof Note that the vector $\mathbf{1}$ is an eigenvector of $W - J = I - (1 - \gamma)/NL - J$ with eigenvalue 0. The vector $\mathbf{1}$ corresponds to the only nonzero eigenvalue of the matrix J and the only zero eigenvalue for the Laplacian matrix L . Therefore the eigenvectors of $W - J$ are exactly the eigenvectors of L , and the k -th eigenvalue of $W - J$ for $k = 1, 2, \dots, N - 1$ is:

$$\lambda_k(W - J) = 1 - \frac{1 - \gamma}{N} \lambda_{N-k}(L) \geq \gamma \quad (\text{C.42})$$

where the inequality follows from the fact that $\lambda(L) \leq N$. Thus, the proof is complete.

Since $\lambda(W - J) \geq 0$ by the above lemma, $\lambda_N(W - J) = 0$ and $z_N(0) = \mathbf{1}^T \alpha(0) = 0$ by construction, the above reduces to

$$\mathbb{E}\{\|\beta(t)\|_2^2\} \geq \sum_{i=1}^{N-1} \lambda_i^{2t}(W - J) |z_i(0)|^2 \quad (\text{C.43})$$

$$= \sum_{i=1}^N \lambda_i^{2t}(W - J) |z_i(0)|^2 \quad (\text{C.44})$$

$$\geq \lambda_{N-1}^{2t}(W - J) \sum_{i=1}^N |z_i(0)|^2 \quad (\text{C.45})$$

$$= \lambda_{N-1}^{2t}(W - J) \|\alpha(0)\|_2^2. \quad (\text{C.46})$$

Substituting (C.46) into (C.30) yields

$$\mathbb{E}\{\|\alpha(t+1)\|_2^2\} \leq \mathbb{E}\{\|\alpha(t)\|_2^2\} - (1 - \lambda_2(W')) \lambda_{N-1}^{2t}(W - J) \|\alpha(0)\|_2^2. \quad (\text{C.47})$$

Repeatedly utilizing the above, we obtain

$$\mathbb{E}\{\|\alpha(t)\|_2^2\} \leq \|\alpha(0)\|_2^2 \left[1 - \left((1 - \lambda_2(W')) \sum_{j=0}^{t-1} \lambda_{N-1}^{2j}(W - J) \right) \right] \quad (\text{C.48})$$

$$= \|\alpha(0)\|_2^2 \left[1 - \left((1 - \lambda_2(W')) \frac{1 - \lambda_{N-1}^{2t+2}(W - J)}{1 - \lambda_{N-1}^2(W - J)} \right) \right], \quad (\text{C.49})$$

where the second line uses the geometric series. Now, taking the limit of the above as t tends to infinity, we obtain the result of the proposition.

C.6 Proof of Corollary 6

Let us first solve for the boundary cases. We consider the $\gamma \rightarrow 1$ case since $\gamma = 0$ simply follows by replacing $\gamma = 0$ in the $U_\infty(\gamma)$ expression (as is clear in the remainder of the proof). Thus, we have

$$\lim_{\gamma \rightarrow 1} U_\infty(\gamma) = \|\alpha(0)\|_2^2 \lim_{\gamma \rightarrow 1} \left(1 - \frac{1 - \lambda_2(W')}{1 - \lambda_{N-1}^2(W - J)} \right) \quad (\text{C.50})$$

$$= \|\alpha(0)\|_2^2 \left(1 - \lim_{\gamma \rightarrow 1} \frac{\frac{2\gamma(1-\gamma)}{N} \lambda_{N-2}(L)}{1 - \left(1 - \frac{1-\gamma}{N} \lambda_1(L)\right)^2} \right) \quad (\text{C.51})$$

$$= \|\alpha(0)\|_2^2 \left(1 - \lim_{\gamma \rightarrow 1} \frac{2\gamma \lambda_{N-2}(L)}{2\lambda_1(L) - \frac{1-\gamma}{N} \lambda_1^2(L)} \right), \quad (\text{C.52})$$

where the second line follows from the facts that $\lambda_2(W') = 1 - 2\gamma(1 - \gamma)/N\lambda_{N-2}(L)$ and $\lambda_{N-1}(W - J) = 1 - (1 - \gamma)/N\lambda_1(L)$ (See Appendix C.7) and the last line follows from expanding the square. Of note is that $\gamma = 0$ follows straight from (C.52) by replacing γ with zero. Note that for $\gamma = 1$, the limit of interest reduces to, after substituting $\gamma = 1$ in the limit expression, the expression stated in the Corollary.

Consider next the monotonicity of the upper-bound w.r.t. the mixing parameter. To prove this claim, we simply show that $\partial U_\infty(\gamma)/\partial\gamma < 0$ for $\gamma \in (0, 1)$. Differentiating $U_\infty(\gamma)$ w.r.t. γ and focusing on the numerator of the $\partial U_\infty(\gamma)/\partial\gamma$ expression (since denominator is always positive and does not effect the sign), after tedious algebraic steps, gives:

$$\text{sgn} \left\{ \frac{\partial U_\infty(\gamma)}{\partial\gamma} \right\} = \text{sgn} \left\{ \frac{\lambda_1(L)}{N} - 2 \right\}. \quad (\text{C.53})$$

where $\text{sgn}\{\cdot\}$ denotes the sign operator. Since $\lambda(L) < 2N$ for non-superconnected graphs, it is easy to see that $\partial U_\infty(\gamma)/\partial\gamma < 0$ for $\gamma \in (0, 1)$.

The last claim simply follows by substituting the optimal mixing parameter

expression into the upper bound expression.

C.7 Proof of Lemma 21

We would like to calculate the eigenvalue $\lambda_1(W' - W'')$. The matrix $W' - W''$ is given by:

$$W' - W'' = I - J - \frac{2\gamma(1-\gamma)}{N}(D - \Phi) - \frac{(1-\gamma)^2}{N^2}(D - \Phi)^2. \quad (\text{C.54})$$

First note that the vector $\mathbf{1}$ is an eigenvector of $W' - W''$ with eigenvalue 0. The vector $\mathbf{1}$ corresponds to the only nonzero eigenvalue of the matrix J and the only zero eigenvalue for the Laplacian matrix $L = D - \Phi$. Therefore the eigenvectors of $W' - W''$ are exactly the eigenvectors of $D - \Phi$, and the k -th eigenvalue of $W' - W''$ for $k = 1, 2, \dots, N - 1$ is:

$$\lambda_k(W' - W'') = 1 - \frac{2\gamma(1-\gamma)}{N}\lambda_{N-k}(L) - \frac{(1-\gamma)^2}{N^2}\lambda_{N-k}(L)^2 \quad (\text{C.55})$$

Thus to characterize $\lambda_1(W' - W'')$ we must characterize the second-smallest eigenvalue of the Laplacian matrix L . The number $\lambda_{N-1}(L)$ is sometimes called the algebraic connectivity of the graph.

An upper bound on $\lambda_{N-1}(L)$ will yield a lower bound on the largest eigenvalue of $W' - W''$. A result of Alon and Milman[7, Theorem 2.7] shows that $\lambda_{N-1}(L)$ is upper bounded by the following function of the diameter $\text{diam}(G)$ of the graph:

$$\lambda_{N-1}(L) \leq \frac{8d_{\max}}{\text{diam}(G)^2} \log_2^2 N \quad (\text{C.56})$$

If the communication radius is chosen large enough, for the random geometric graph with standard connectivity assumptions, $d_{\max} = \Theta(\log N)$ (see [16]). The

diameter can be found as the number of hops to get from one corner to the diagonally opposite corner, so it is $\Theta(\sqrt{N/\log N})$. Thus the whole bound is:

$$\lambda_{N-1}(L) = O\left(\frac{\log^4 N}{N}\right). \quad (\text{C.57})$$

This gives the bound

$$\lambda_1(W - W'') = 1 - O\left(\frac{\log^4 N}{N^2}\right). \quad (\text{C.58})$$

To upper bound $\lambda_1(W - W'')$ we need a nontrivial lower bound on $\lambda_{N-1}(L)$. A result of Mohar [67] gives this lower bound in terms of the diameter of the graph. Mohar's lower bound is:

$$\lambda_{N-1}(L) \geq \frac{4}{N \cdot \text{diam}(G)}. \quad (\text{C.59})$$

Therefore

$$\lambda_{N-1}(L) = \Omega\left(\frac{\sqrt{\log N}}{N^{3/2}}\right), \quad (\text{C.60})$$

and

$$\lambda_1(W - W'') = 1 - \Omega\left(\frac{\sqrt{\log N}}{N^{5/2}}\right). \quad (\text{C.61})$$

APPENDIX D

APPENDIX OF CHAPTER 5

D.1 Proof of Lemma 22

- (1) We first note that W^∞ is an idempotent matrix. Thus by using (5.3) and (5.4):

$$W^\infty W^\infty = W^\infty \Rightarrow B'D' = E' = 0.$$

Since $B' = 1/|\mathcal{S}_S|\mathbf{1}\mathbf{1}^T > 0$ and $D' \geq 0$, the above equality holds if and only if $D' = 0$. The proof that $G' = 0$ follows easily and therefore it is omitted.

- (2) We first note that by the first part of the lemma, W^∞ has the structure:

$$W^\infty = \begin{bmatrix} A' & 0 & 0 \\ 1/|\mathcal{S}_S|\mathbf{1}\mathbf{1}^T & 0 & 0 \\ C' & F' & P' \end{bmatrix}.$$

Since $W^\infty W = W^\infty$, then $1/|\mathcal{S}_S|\mathbf{1}\mathbf{1}^T D = 0$. Since D is nonnegative, then $D = 0$. Similarly, $1/|\mathcal{S}_S|\mathbf{1}\mathbf{1}^T G = 0$ and $G \geq 0$, thus $G = 0$.

- (3) By noting once again $W^\infty W = W^\infty$, $1/|\mathcal{S}_S|\mathbf{1}\mathbf{1}^T A = 1/|\mathcal{S}_S|\mathbf{1}\mathbf{1}^T$. Therefore, $\mathbf{1}^T A = \mathbf{1}^T$.

D.2 Proof of Lemma 23

We first prove that that if

$$\lim_{t \rightarrow \infty} \left[\sum_{l=0}^t D^{t-l} B A^l \right]_{1:K} = \frac{1}{M} \mathbf{1}\mathbf{1}^T,$$

then,

$$\lim_{t \rightarrow \infty} \left[\sum_{l=0}^t D^{t-l} B A^\infty \right]_{1:K} = \frac{1}{M} \mathbf{1} \mathbf{1}^T.$$

Let's denote by $B' = \lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^l$, and recall that $B' < \infty$ exists by $W^\infty < \infty$ assumption. If we multiply both sides of the equality by A^∞

$$\begin{aligned} B' A^\infty &= \left(\lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^l \right) A^\infty \\ &= \left(\lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^l A^\infty \right) = \lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^\infty, \end{aligned}$$

where the second equality follows from the fact that A^∞ is a fixed matrix and $B' < \infty$ (therefore one can take A^∞ inside the limit), and the third equality is due to the fact that $A^l A^\infty = A^\infty \forall l \geq 0$. Since the column sum of A is constant and equal to 1 (by Lemma 22) and A is a non-negative matrix, then $\lambda = 1$ is an eigenvalue of A corresponding to eigenvector $\mathbf{1}^T$ [40]. Therefore, $\mathbf{1}^T A^\infty = \mathbf{1}^T$. Keeping that in mind, $[B' A^\infty]_{1:K} = 1/|\mathcal{S}_S| \mathbf{1}^T A^\infty = 1/|\mathcal{S}_S| \mathbf{1}^T = [B']_{1:K}$. This concludes the first part of the proof.

Now, we show that, if $\lim_{t \rightarrow \infty} \left[\sum_{l=0}^t D^{t-l} B A^\infty \right]_{1:K} = \frac{1}{M} \mathbf{1} \mathbf{1}^T$, then $\lim_{t \rightarrow \infty} \left[\sum_{l=0}^t D^{t-l} B A^l \right]_{1:K} = \frac{1}{M} \mathbf{1} \mathbf{1}^T$. Let's denote $B' = \lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^\infty$, and $B' < \infty$ exists by assumption. Then,

$$\begin{aligned} B' &= \lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^\infty \\ &= \left(\lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^l A^\infty \right) = \left(\lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^l \right) A^\infty, \end{aligned}$$

where the second equality follows from the fact that $A^l A^\infty = A^\infty$. To prove the third equality, we first note that $\lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^l$ exists (although its value may not be finite) since, each term in the expression is non-negative. Since A^∞ does not have all zeros rows, each term in the matrix $(\lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^l)$

is present in B' with a strictly positive coefficient. Therefore, one may take A^∞ outside of the limit. We also note that nonexistence of all zeros row in A^∞ matrix can be shown by considering the equality $AA^\infty = A^\infty$.

Let's define $L' = (\lim_{t \rightarrow \infty} \sum_{l=0}^t D^{t-l} B A^l)$. Since $W^\infty W^\infty = W^\infty$, then $L' A^\infty + D' L' = L'$. Noting that $L' A^\infty = B'$ and the first K rows D' matrix are all zeros by the third condition of Theorem 3, the first K rows of L' is equal to the first K rows of B' . Therefore, $\lim_{t \rightarrow \infty} [\sum_{l=0}^t D^{t-l} B A^l]_{1:K} = \frac{1}{M} \mathbf{1} \mathbf{1}^T$. This concludes the proof.

D.3 Proof of Lemma 24

We first note that the rank of a limiting matrix W^∞ is equal to both the number of eigenvalues of W that are equal to 1 and the number of eigenvalues of W^∞ that are equal to 1. Moreover, eigenvalues of W^∞ are the union of the eigenvalues of A^∞ and D^∞ since W^∞ is in the block lower triangular form [44]. Since any given source class SC_i forms an irreducible sub-network, the largest eigenvalue of A_{SC_i} is equal to 1 (since column sum is equal to 1 by Lemma 22) and it is unique in magnitude (due to irreducibility) [44]. Such an argument holds for all source classes. Thus, the number of non-zero eigenvalues of $A^\infty = \lim_{t \rightarrow \infty} A^t$ is equal to the number of source classes. Therefore, this is a lower bound for the rank of W^∞ .

To prove the upper bound, we first note that we can partition the network into classes. Moreover, these classes are irreducible and disjoint by the definition. For a given class, the spectral radius upper bounded by 1 and the largest eigenvalue is unique and equal to the spectral radius [44]. Therefore, W^∞ can

have at most $\#$ of classes non-zero eigenvalues.

D.4 Proof of Lemma 26

Given $\rho(D) < 1$, $\lim_{t \rightarrow \infty} D^t = 0$ [40], and the fourth condition in the Theorem 3 simplifies to:

$$\begin{aligned} \lim_{t \rightarrow \infty} \sum_{l=0}^t ([D^l]_{1:K} B A^\infty) &= \lim_{t \rightarrow \infty} \left(\sum_{l=0}^t [D^l]_{1:K} \right) B A^\infty \\ &= [(I - D)^{-1}]_{1:K} B A^\infty = \frac{1}{M} \mathbf{1} \mathbf{1}^T, \end{aligned} \quad (\text{D.1})$$

where the third equality follows from the fact that $\lim_{t \rightarrow \infty} \sum_{l=0}^t [D^l] = [(I - D)^{-1}]$ when $\rho(D) < 1$ [44]. (D.1) can be rewritten as:

$$R(I - D)^{-1} B A^\infty = \frac{1}{M} \mathbf{1} \mathbf{1}^T, \quad (\text{D.2})$$

where $R \in \mathbb{R}^{K \times N}$ with diagonal elements equal to one, and the rest are all equal to zero. Since the rank of $1/M \mathbf{1} \mathbf{1}^T$ is one, then the rank of the multiplication on the left hand side of the equality has to be equal to one. At this point, we remind our readers the following well-known result regarding the rank of a matrix multiplication:

Lemma 39 [66] *If T is $m \times N$ and Q is $N \times p$, then $\text{rank}(TQ) \geq \text{rank}(T) + \text{rank}(Q) - N$.*

By noting that the ranks of $R, (I - D)^{-1}, A^\infty, \mathbf{1} \mathbf{1}^T$ are equal to $K, N, \#$ of source classes and 1 respectively, our result follows from (D.2).

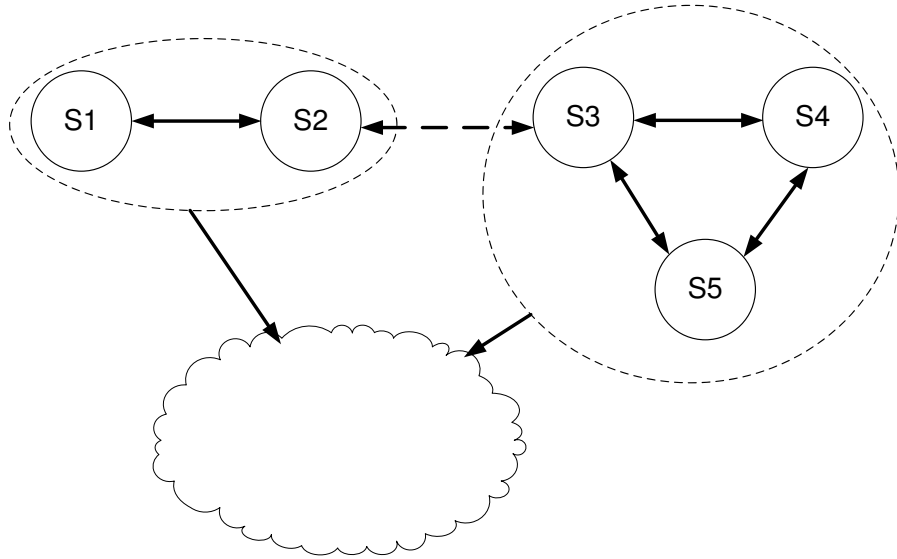


Figure D.1: Source clustering due to W^* . Dashed line between S_2 and S_3 represents zero weight link. Cloud represents rest of the network.

D.5 Proof of Lemma 27

Consider CAR $F(\mathcal{S}_S, \mathcal{S}_D)$ and a network $F(E)$. By hypothesis, the problem has at least one feasible solution, namely W^* . If W^* utilizes the minimal source class, then the lemma is satisfied. Let us assume that W^* does not utilize the minimal class, *i.e.*, it assigns zero weights to some of the existing links in between source nodes. Without loss of generality, we assume that under minimal class the network has only one class, and W^* has two source classes since it assigns zero weight to a link between a pair of source nodes. An example network is given in Fig. D.1. We further assume that the source nodes have already converged to their limiting states. We note that this assumption does not change the feasibility of the problem due to Lemma 23. Without loss of generality, we also assume that each source's limiting state value is the average of the initial values of the source nodes in that particular class.

Let there be M_1 source nodes in Cluster 1 and M_2 source nodes in Cluster 2 and $M_1 + M_2 = M$. Source nodes in class 1 and 2 have already converged to $1/M_1 \sum_{i \in C_1} x_i(0)$ and $1/M_2 \sum_{i \in C_2} x_i(0)$ respectively, where C_i is the set of nodes in class i . Since W^* is a feasible code, all of the destinations in the network converge to the average of the all source nodes. But, this is only possible if destination nodes have access to:

$$\frac{M_1}{M} \left(\frac{1}{M_1} \sum_{i \in C_1} x_i(0) \right) + \frac{M_2}{M} \left(\frac{1}{M_2} \sum_{i \in C_2} x_i(0) \right) = \frac{1}{M} \sum_{i \in \mathcal{S}_S} x_i(0), \quad (\text{D.3})$$

in the limit. At this point, we design the minimal class code \bar{W} from W^* as follows: We assign non-zero weights to all of the links between source nodes, and choose these weights such that source nodes converge to the average of the initial source values. Since all of the links in between source nodes are assigned non-zero weight, \bar{W} corresponds to the minimal class. We keep B and D partitions of W^* unchanged, utilize these in \bar{W} . Since, the link weights governing the communication among non-source nodes (partition D) and also between sources and non-source nodes (partition B) have not changed, the destination nodes will converge to:

$$\frac{M_1}{M} \left(\frac{1}{M} \sum_{i \in \mathcal{S}_S} x_i(0) \right) + \frac{M_2}{M} \left(\frac{1}{M} \sum_{i \in \mathcal{S}_S} x_i(0) \right) = \frac{1}{M} \sum_{i \in \mathcal{S}_S} x_i(0). \quad (\text{D.4})$$

We note that the equation given above is very similar to (D.3). In particular, the coefficients in front of the first and the second terms in the summations are exactly the same. As we have mentioned above, this is true since B and D partitions of W^* remains unchanged. Moreover, the terms in the parenthesis have changed, since each source node now has access to the global average.

Since (D.4) is equal to the average the source nodes, \bar{W} is a feasible minimal class solutions and hence the lemma is proved.

The proof can be easily generalized to the case of several source classes and to the case where all sources do not converge to the average of the initial source node values in that particular class, by accounting more than two terms in (D.3)-(D.4).

D.6 Proof of Lemma 28

Consider a network $F(E)$ and the source and destinations sets, S_S and S_D , respectively. We partition the network into two disjoint sets (P, P^c) such that there exists at least one source class-destination pair on both sides of the network. We consider the case where there exists at least one partition such that the number of cut edges is strictly less than two. The graphical representation of the network is given in Fig. D.2. The curve represents the boundary between two disjoint partitions. $SS1$ represents all of the sources nodes in partition P , and $D1$ represents all of the destination nodes in partition P . Similarly, $SS2$ and $D2$ represents the source and destination nodes in partition P^c . We note that the number of cut edges cannot be zero since otherwise the network would be disconnected. Thus the number of cut edges between P and P^c has to be equal to one. We will assume that there exists a feasible W for this problem, and prove our claim by finding a contradiction.

In the network topology given in Fig. D.2, we denote $I1$ and $I2$ as bottleneck nodes, *i.e.*, the edge between $I1$ and $I2$ is the only cut edge for the partition. We index the source nodes in the network such that the first M_1 source nodes are in P and the last M_2 source nodes are in P^c . We note that $M_1 + M_2 = M$. We refer to W_{ij}^∞ as the weight of the node j at node i .

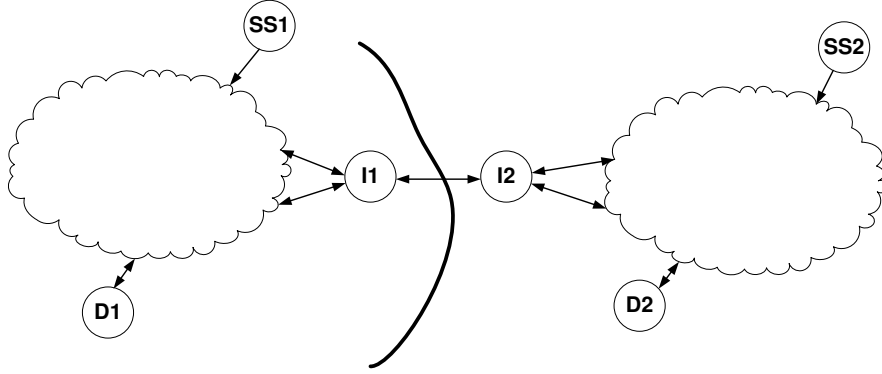


Figure D.2: A network with two partitions and a single cut edge.

Since, there is only one cut edge between partition P and P^c , information from $SS2$ to $D1$ has to flow through the edge $(I2, I1)$. As we have discussed in Section 5.5, W matrix is indeed a transition probability matrix for the chain \mathcal{M} , W^∞ consists of corresponding absorption probabilities by the source classes. It is clear that $SS1$ and $SS2$ are absorbing classes in our case. The probability of being absorbed by $SS2$ when the chain has been initialized at state $I2$ is larger than the probability of being absorbed by $SS2$ when the chain has been initialized at state $I1$, since, in the latter case, the chain has to visit $I2$ before being absorbed by $SS2$. Therefore, for all $k \in SS2$, the frequency of visiting k in the long run starting from $I2$ is larger than the frequency of visiting k in the long run starting from $I1$. Since these probabilities are measured by the entries of W^∞ matrix, the argument follows. Mathematically speaking, our result is:

$$W_{I1,k}^\infty \leq W_{I2,k}^\infty \quad \forall k \in \{M_1 + 1, \dots, M\}. \quad (\text{D.5})$$

Similarly, the weight of $SS1$ at $I1$ has to be greater than or equal to the weight of $SS1$ at $I2$, i.e.,

$$W_{I2,k}^\infty \leq W_{I1,k}^\infty \quad \forall k \in \{1, \dots, M_1\}. \quad (\text{D.6})$$

Similarly, the weight of $SS2$ at $D1$ is less than or equal to the weight of $SS2$

at $I1$ and, the weight of the $SS1$ information to $D2$ is less than or equal to that of $SS1$ at $I2$. Moreover, since, W is feasible, sources' weights at destination nodes $D1$ and $D2$ are equal to $1/M$. Combining these facts with (D.5)-(D.6):

$$\frac{1}{M} \leq W_{I1,k}^\infty \leq W_{I2,k}^\infty \quad \forall k \in \{1, \dots, M_1\}, \quad (\text{D.7})$$

$$\frac{1}{M} \leq W_{I2,k}^\infty \leq W_{I1,k}^\infty \quad \forall k \in \{M_1 + 1, \dots, M\}. \quad (\text{D.8})$$

Since $W\mathbf{1} = \mathbf{1}$, then $W^\infty\mathbf{1} = \mathbf{1}$:

$$\sum_{k=1}^M W_{I1,k}^\infty = 1 \text{ and } \sum_{k=1}^M W_{I2,k}^\infty = 1. \quad (\text{D.9})$$

Since all of the terms in both summations are lower bounded by $1/M$, and there are M terms in both summations, the equality in (D.9) holds if and only if $W_{I1,k}^\infty = W_{I2,k}^\infty = 1/M \quad \forall k \in \{1, \dots, M\}$. Such an equality implies that $I1$ and $I2$ also converge to the average.

This observation is particularly important for the following reason: Since we are focusing on the limiting values of the node states, they are invariant with respect to the transformation with respect to W matrix, *i.e.*, $WW^\infty = W^\infty$. In particular, the weight of $SS2$ at node $I1$ has to be equal to a convex combination of the weights of $SS2$ at node $I1$ and at $I1$'s neighbors. We have already shown that the weight of $SS2$ at $I1$ and $I2$ (which is a neighbor of $I1$) are equal to $1/M$. Since all the neighbors of $I1$ except $I2$ belong to the partition P , the weight of $SS2$ at each of these nodes is upper bounded by $1/M$. Then the weight of $SS2$ at all of the neighbors of $I1$ has to be equal to the $1/M$. Otherwise, this would not be a stable point.

One can now focus on the neighbor set of node $I1$ (denoted by \mathcal{N}_{I1}) and argue that the neighbors of \mathcal{N}_{I1} also have $1/M$ as the weight of node $SS2$ since all of them belong to the partition P . One can utilize the argument above itera-

tively to show that the weight of $SS2$ at all of the nodes in P partition converge to $1/M$.

However, this is a contradiction since the weight of $SS2$ at $SS1$ cannot be equal to $1/M$, since since these nodes can not communicate with $SS2$. Therefore, there does not exist any feasible code for such a topology.

D.7 Proof of Lemma 29

We first to note that the integer programming formulation given in the lemma is a directed multicommodity flow problem with acyclicity constraint. In particular, one can map the variable b_i^{kl} to net inflow at node i of data with origin k and destination l . z_{ij}^{kl} indicates the amount of information with origin k and destination l that flows through link (i, j) . y_{ij}^k is equal to one if there exists at least one flow on (i, j) that is originated from source class k .

We assume that the integer programming formulation has a feasible solution. We will show that W , which is constructed as in (5.16), is a feasible AVT solution.

We first note that for a given source node i , we assign equal weights to all of its source neighbors and itself. Therefore, each source node in a given class will converge to the average of the nodes in that particular class (c.f.[92]). Utilizing Lemma 23, we can assume that source nodes have already converged to the averages.

Moreover, W in (5.16) is stochastic by construction, therefore it has a limit. By Theorem 3.1 of [81], the rank of W^∞ is equal to the number of source classes

in the network. Similar to our discussion in Section 5.4.1, we denote the columns of W^∞ as u_m , $1 \leq m \leq r_{W^\infty}$. These columns are the eigenvectors of W corresponding to the eigenvalue 1.

We will prove that m -th eigenvector u_m will have the following structure:

$$[u_m]_j = \begin{cases} \frac{1}{|\mathcal{S}_{C_m}|}, & \text{if } j \in \mathcal{S}_{C_m}, \\ 0, & \text{if } j \in \mathcal{S}_S \text{ and } j \notin \mathcal{S}_{C_m}, \\ \frac{1}{\sum_{l \in \mathcal{N}_j} \sum_{k \in \mathcal{S}_C} |\mathcal{S}_C(k)| y_{lj}^k}, & \text{if } \sum_{l \in \mathcal{N}_j} y_{lj}^m \neq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (\text{D.10})$$

where \mathcal{S}_{C_m} is the set of source nodes that belong to the source class m and $[\cdot]_j$ denotes the j -th element of its argument. If u_m is an eigenvector of W corresponding to the eigenvalue 1, then $W u_m = u_m$ must hold, *i.e.*:

$$[W u_m]_j = \sum_{i=1}^N W_{ji} [u_m]_i \quad \forall j \in \{1, 2, \dots, N\}.$$

If node j is a source node which is in the class m , then:

$$[W u_m]_j = \sum_{i=1}^N W_{ji} [u_m]_i = \frac{1}{|\mathcal{S}_{C_m}|} \sum_{i \in \{\mathcal{N}_j \cup j\}} \frac{1}{|\mathcal{N}_j| + 1} = \frac{1}{|\mathcal{S}_{C_m}|},$$

where the second equality follows from the fact that all of the neighbors of node j belongs to the set \mathcal{S}_{C_m} , and the third equality is due to the fact that each of the incoming links has the same weight, *i.e.* $1/|\mathcal{N}_j| + 1$, by (5.16). Therefore, $[W u_m]_j = [u_m]_j$, $\forall j \in \mathcal{S}_{C_m}$.

If j is a source node, but does not belong to the class m , then:

$$\begin{aligned} [W u_m]_j &= \sum_{i=1}^N W_{ji} [u_m]_i = \sum_{i \in \mathcal{S}_S \setminus \mathcal{S}_{C_m}} W_{ji} [u_m]_i + \sum_{i \in \mathcal{S} \setminus (\mathcal{S}_S \setminus \mathcal{S}_{C_m})} W_{ji} [u_m]_i \\ &= \sum_{i \in \mathcal{S}_S \setminus \mathcal{S}_{C_m}} W_{ji} 0 + \sum_{i \in \mathcal{S} \setminus (\mathcal{S}_S \setminus \mathcal{S}_{C_m})} 0 [u_m]_i = 0 = [u_m]_j, \end{aligned}$$

where \setminus denotes the "set difference". The third equality follows from the construction of u_m in (D.10) and the fact that source nodes, which do not belong to the source class m , cannot hear from the class m .

If j is not a source, then:

$$[Wu_m]_j = \sum_{i=1}^N W_{ji}[u_m]_i = \sum_{i \in \mathcal{S}_{C_m}} W_{ji}[u_m]_i + \sum_{i \in \{\mathcal{S}_S \setminus \mathcal{S}_{C_m}\}} W_{ji}[u_m]_i + \sum_{i \notin \mathcal{S}_S} W_{ji}[u_m]_i. \quad (\text{D.11})$$

We note that the first term above represents the summation over the source nodes which are in class m , the second term is the summation over the rest of the source nodes, and the last summation is over the non-source nodes. The second summation in (D.11) is zero, since $[u_m]_i = 0$ for all source nodes i which are not included in the source class m by the construction in (D.10). Moreover, of all the terms in the first and the third summation, only a single term can be non-zero. This is true due to the acyclicity constraint in (5.14). If there were two non-zero terms in the summation, this would mean that node j has two neighbors both of which carry information from the source class m . But, this would result in a cycle in the flow graph. Thus, such flow would not be a feasible solution to the integer programming formulation given in the lemma.

We first assume that there exists a non-zero term, the term is in the first summation in (D.11) and, we denote the index of this term as i . Then by (5.16) and (D.10):

$$[Wu_m]_j = \frac{|\mathcal{S}_{C_m}| y_{ij}^m}{\sum_{l \in \mathcal{N}_j} \sum_{k \in \mathcal{S}_C} |\mathcal{S}_C(k)| y_{lj}^k} \frac{1}{|\mathcal{S}_{C_m}|} = \frac{1}{\sum_{l \in \mathcal{N}_j} \sum_{k \in \mathcal{S}_C} |\mathcal{S}_C(k)| y_{lj}^k} = [u_m]_j.$$

Note that $y_{ij}^m = 1$, since we assumed that there is a flow from node i to node j which is originated from class m .

Next, we assume that the non-zero term is in the third summation in (D.11) and denote the index of this term as i . Then by (5.16) and (D.10):

$$[Wu_m]_j = \frac{\sum_{k \in \mathcal{S}_C} |S_C(k)| y_{ij}^k}{\sum_{l \in \mathcal{N}_j} \sum_{k \in \mathcal{S}_C} |S_C(k)| y_{lj}^k} \frac{1}{\sum_{l \in \mathcal{N}_i} \sum_{k \in \mathcal{S}_C} |S_C(k)| y_{li}^k}. \quad (\text{D.12})$$

We remind the reader that if the term above is non-zero, then $\sum_{l \in \mathcal{N}_i} \sum_{k \in \mathcal{S}_C} |S_C(k)| y_{li}^k = \sum_{k \in \mathcal{S}_C} |S_C(k)| y_{ij}^k$. One can think of $\sum_{l \in \mathcal{N}_i} \sum_{k \in \mathcal{S}_C} |S_C(k)| y_{li}^k$ as the total number of flows from distinct classes (rescaled by the corresponding class sizes) that are going into the node i . Moreover, if there exists at least one flow from node i to node j , then there has to be at least one flow from node i to node j from each distinct class whose information is present at the node i . Otherwise, acyclicity constraint would not hold. Therefore, the equality holds. Then, we can rewrite (D.12) as:

$$[Wu_m]_j = \frac{1}{\sum_{l \in \mathcal{N}_j} \sum_{k \in \mathcal{S}_C} |S_C(k)| y_{lj}^k} = [u_m]_j.$$

Now, we assume that all of the terms in (D.11) are zero. But due to the construction of W and u_m , this is only possible if there is no flow that is going into node j from source class m . If this is the case, $[Wu_m]_j = [u_m]_j = 0$. Since we covered all possible cases, this proves that $\{u_m\}_{m=1}^{r_{W^\infty}}$ are the eigenvectors of W corresponding to the eigenvalue 1. Moreover, due to their constructions, these eigenvectors are linearly independent.

We construct the left eigenvectors corresponding to the eigenvalue 1 as follows:

$$[c_m]_j = \begin{cases} 1, & \text{if } j \in \mathcal{S}_{C_m}, \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, it is easy to check that $c_m^T u_n = 0$ if $m \neq n$. Then by [74]:

$$\lim_{t \rightarrow \infty} W^t = \sum_{m=1}^{r_{W^\infty}} \frac{u_m c_m^T}{c_m^T u_m} = \sum_{m=1}^{r_{W^\infty}} u_m c_m^T,$$

where the second equality follows from the fact that $c_m^T u_m = |\mathcal{S}_{C_m}|/|\mathcal{S}_{C_m}| = 1$ for all m . At this point, we need to observe that $[u_m]_j = 1/|\mathcal{S}_S|$ if $j \in \mathcal{S}_D$, due to the integer programming formulation in the lemma and the eigenvector construction in (D.10). This concludes the proof.

BIBLIOGRAPHY

- [1] Daron Acemoglu, Asuman Ozdaglar, and Ali Parandehgheibi. Spread of misinformation in social networks. Jun 2009.
- [2] Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li, and Raymond W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, pages 1204–1216, 2000.
- [3] Lan F. Akyildiz, Welljan Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks, 2002.
- [4] Jamal N. Al-Karaki, Raza Ul-Mustafa, and Ahmed E. Kamal. Data aggregation and routing in wireless sensor networks: Optimal and heuristic algorithms. *Computer Networks*, 53(7):945–960, 2009.
- [5] David Aldous and Jim Fill. Reversible markov chains and random walks on graphs. <http://www.stat.berkeley.edu/~aldous/RWG/book.html>.
- [6] David Aldous and Jim Fill. Reversible markov chains and random walks on graphs. <http://stat-www.berkeley.edu/~aldous/RWG/book.html>, 1994.
- [7] N. Alon and V.D. Milman. λ_1 , isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38:73–88, 1985.
- [8] T. C. Aysal, M. E. Yildiz, Anand D. Sarwate, and A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal Processing*, July 2009.
- [9] T. C. Aysal, M. E. Yildiz, and A. Scaglione. Broadcast gossip algorithms. In *Proceedings of the 2008 IEEE Information Theory Workshop*, Porto, Portugal, May 2008.
- [10] T.C. Aysal, M.J. Coates, and M.G. Rabbat. Distributed average consensus using probabilistic quantization. In *Proc. IEEE Statistical Signal Processing Workshop*, Madison, WI, Aug. 2007.
- [11] T.C. Aysal, M.J. Coates, and M.G. Rabbat. Probabilistically quantized distributed averaging. *IEEE Transactions Signal Processing*, October 2008.

- [12] Tuncer C. Aysal, Mark J. Coates, and Michael G. Rabbat. Rates of convergence of distributed average consensus with probabilistic quantization. In *Proceedings of the Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 2007.
- [13] Tuncer C. Aysal, Mehmet E. Yildiz, Anand D. Sarwate, and Anna Scaglione. Broadcast gossip algorithms: Design and analysis for consensus. In *CDC*, pages 4843–4848, 2008.
- [14] N. T. J. Bailey. *The Mathematical Theory of Infectious Diseases*. Hafner, New York, second edition, 1975.
- [15] Frank M. Bass. A new product growth for model consumer durables. *Manage. Sci.*, 15, 1969.
- [16] F. Benezit, A. G. Dimakis, P. Thiran, and M. Vetterli. Gossip along the way: Order-optimal consensus through randomized path averaging. In *Proceedings of the Allerton Conference on Communication, Control, and Computing*, Allerton, IL, Sep. 2007.
- [17] Florence Benezit, Alexandros Dimakis, Patrick Thiran, and Martin Vetterli. Gossip along the way: order-optimal consensus through randomized path averaging. In *Allerton*, 2007.
- [18] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.
- [19] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Mixing times for random walks on geometric random graphs. In *Proceedings of the Workshop on Analytic Algorithms and Combinatorics*, Vancouver, BC, January 2005.
- [20] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Info. Theory*, 52(6):2508–2530, June 2006.
- [21] S. Boyd, A. Gosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, June 2007.
- [22] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge UK: Cambridge University Press, 2004.

- [23] A. Browder. *Mathematical Analysis: An Introduction*. New York: Springer-Verlag, 1996.
- [24] Ruggero Carli, Fabio Fagnani, Alberto Speranzon, and Sandro Zampieri. Communication constraints in the average consensus problem. *Automatica*, 44(3):671–684, 2008.
- [25] Pietro Ciciriello, Luca Mottola, and Gian Pietro Picco. Efficient routing from multiple sources to multiple sinks in wireless sensor networks. *Lecture Notes in Computer Science*, 4373, April 2007.
- [26] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.
- [27] J. Theodore Cox and David Griffeath. Diffusive clustering in the two dimensional voter model. *The Annals of Probability*, 14(2):347–379, 1986.
- [28] P. J. Davis. *Circulant Matrices*. Wiley, 1979.
- [29] Stephen Edward Deering. *Multicast routing in a datagram internetwork*. PhD thesis, Stanford University, 1992.
- [30] M.H. DeGroot. Reaching a consensus. *Journal of American Statistical Association*, 1974.
- [31] J. E. Dennis, Jr. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Classics in Applied Mathematics, 16)*. Soc for Industrial & Applied Math, 1996.
- [32] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269–271, 1959.
- [33] Alexandros G. Dimakis, Anand D. Sarwate, and Martin J. Wainwright. Geographic gossip: Efficient aggregation for sensor networks. In *Proceedings of the Information Processing in Sensor Networks*, Nashville, TN, April 2006.
- [34] Alexandros G. Dimakis, Anand D. Sarwate, and Martin J. Wainwright. Geographic gossip: Efficient averaging for sensor networks. *IEEE Transaction on Signal Processing*, 56(3), March 2008.

- [35] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. *IEEE Transactions on Information Theory*, pages 2745–2759, 2005.
- [36] David Easley and Jon Kleinberg. *Networks, Crowds and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 1 edition, 2010.
- [37] Mihaela Enachescu, Ashish Goel, Ramesh Govindan, and Rajeev Motwani. Scale-free aggregation in sensor networks. *Theor. Comput. Sci.*, pages 15–29, 2005.
- [38] F. Fagnani and S. Zampieri. Randomized consensus algorithms over large scale networks. *IEEE Journal on Selected Areas in Communications*, 26:634–649, 2008.
- [39] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, Sep. 2004.
- [40] F.R. Gantmacher. *Matrix Theory*, volume 2. Chelsea Publishing, 1959.
- [41] A. Giridhar and P. R. Kumar. Towards a theory of in-network computation in wireless sensor networks. *IEEE Communications Magazine*, 44(4):98–107, Apr. 2006.
- [42] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46:388–404, March 2000.
- [43] Richard A. Holley and Thomas M. Liggett. Ergodic theorems for weakly interacting infinite systems and the voter model. *The Annals of Probability*, 3(4):643–663, 1975.
- [44] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1987.
- [45] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, 2003.
- [46] Matthew O. Jackson. *Social and Economic Networks*. Princeton University Press, 1 edition, 2008.

- [47] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. on Automatic Control*, 48:988–1001, 2003.
- [48] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *AC-48(6)*:988–1001, June 2003.
- [49] Meng Ji, A. Muhammad, and M. Egerstedt. Leader-based multi-agent coordination: controllability and optimal control. In *American Control Conference, 2006*, June 2006.
- [50] S. Kar and J. M. F. Moura. Distributed average consensus in sensor networks with random link failures and communication channel noise. In *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2007.
- [51] S. Kar and J.M.F. Moura. Distributed consensus algorithms in sensor networks: Quantized data and random link failures. *Accepted to IEEE Transactions on Signal Processing*, 2009 2009.
- [52] Soumya Kar, S. A. Aldosari, and J. M. F. Moura. Topology for distributed inference on graphs. *IEEE Transactions on Signal Processing*, 56(6):2609–2613, June 2008.
- [53] A. Kashyap, T. Basar, and R.Srikant. Quantized consensus. *Automatica*, 43:1192–1203, July 2007.
- [54] W. Kermack and A. McKendrick. Contributions to the mathematical theory of epidemics: Further studies of the problem of endemicity. *Bulletin of Mathematical Biology*, 53(1):89–118, March 1991.
- [55] U. A. Khan, S. Kar, and J. M. F. Moura. Distributed sensor localization in random environments using minimal number of anchor nodes. *IEEE Transactions on Signal Processing*, *Accepted*, 2008.
- [56] Jon Kleinberg. *Cascading Behavior in Networks: Algorithmic and Economic Issues*. Cambridge University Press, 2007.
- [57] Ralf Koetter and Muriel Medard. An algebraic approach to network coding. *IEEE Transaction on Networking*, pages 782–795, 2003.

- [58] Ralf Koetter and Muriel Medard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11:782–795, 2003.
- [59] Bhaskar Krishnamachari, Deborah Estrin, and Stephen B. Wicker. The impact of data aggregation in wireless sensor networks. In *Proceedings of the 22nd International Conference on Distributed Computing Systems*, pages 575–578, Washington, DC, USA, 2002.
- [60] A. Leon-Garcia and I. Widjaja. *Communication Networks*. McGrawHill, U.S, 2000.
- [61] Thomas M. Liggett. *Interacting Particle Systems*. Springer Berlin Heidelberg, 2005.
- [62] J. Lin and A. S. Morse. The multi-agent rendezvous problem. In *IEEE Conf. on Decision and Control Proc.*, pages 1508–1513, December 2003.
- [63] Z. Lin, S. Cheng, A. D. Liveris, and Z. Xiong. Slepian-wolf coded nested quantization (swc-nq) for wyner-ziv coding: Performance analysis and code design. In *Data Compression Conference Proc.*, pages 322–331, 2004.
- [64] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. The design of an acquisitional query processor for sensor networks. In *ACM SIGMOD*, pages 491–502, 2003.
- [65] Samuel Madden, Robert Szewczyk, Michael J. Franklin, and David Culler. Supporting aggregate queries over ad-hoc wireless sensor networks. In *IEEE Workshop on Mobile Computing and Systems Applications*, pages 49–58, 2002.
- [66] Carl Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 2001.
- [67] B. Mohar. *Graph Theory, Combinatorics, and Applications*, volume 2, chapter The Laplacian spectrum of graphs, pages 871–898. Wiley, 1991.
- [68] D. Mosk-Aoyama and D. Shah. Information dissemination via network coding. In *Information Theory, 2006 IEEE International Symposium on*, pages 1748–1752, July 2006.
- [69] D. Mosk-Aoyama and D. Shah. Fast distributed algorithms for computing

- separable functions. *Information Theory, IEEE Transactions on*, 54(7):2997–3007, July 2008.
- [70] Suman Nath, Phillip B. Gibbons, Srinivasan Seshan, and Zachary R. Anderson. Synopsis diffusion for robust aggregation in sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 250–262, 2004.
- [71] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis. On distributed averaging algorithms and quantization effects. Technical report, LIDS Report 2274, MIT, November 2007.
- [72] A. Nedić and A. Ozdaglar. Convergence rate for consensus with delays. Technical report, LIDS Report 2774, MIT, October 2007.
- [73] Angelia Nedic, Alexander Olshevsky, Asuman E. Ozdaglar, and John N. Tsitsiklis. On distributed averaging algorithms and quantization effects. In *CDC*, pages 4825–4830. IEEE, 2008.
- [74] Rufus Oldenburger. Infinite powers of matrices and characteristic roots. *Duke Mathematics Journal*, 12:357–361, 1940.
- [75] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Automat. Contr.*, 49, no. 9:1520–1533, Sept. 2004.
- [76] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.
- [77] B. Polyak. *Introduction to Optimization*. Optimization Software Inc., New York, 1987.
- [78] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (discus): design and construction. In *Data Compression Conference Proc.*, pages 158–167, March 1999.
- [79] M. Rabbat, R. Nowak, and J. Bucklew. Generalized consensus computation in networked systems with erasure links. In *IEEE Workshop on Signal Processing Advances in Wireless Communications Proc.*, pages 1088–1092, June 2005.

- [80] W. Ren, R. W. Beard, and D. B. Kingston. Multi-agent kalman consensus with relative uncertainty. In *American Control Conference Proc.*, pages 1865 – 1870, OR, USA, June 2005.
- [81] Uriel G. Rothblum. Algebraic eigenspaces of nonnegative matrices. *Linear Algebra and Its Applications*, 12:281–292, 1975.
- [82] R. O. Saber and R. M. Murray. Agreement problems in networks with directed graphs and switching topology. *California Institute of Technology Technical Report*, CIT-CDS 03-005, 2003.
- [83] V. Saligrama and D. Castanon. Reliable distributed estimation with intermittent communications. In *45th IEEE Conference on Decision and Control*, pages 6763–6768, San Diego, CA, Dec. 2006.
- [84] E. Seneta. *Non-negative Matrices and Markov Chains*. Springer Series in Statistics, 1981.
- [85] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory*, IT-19:471–480, July 1973.
- [86] A. B. Sripad and D. L. Snyder. A necessary and sufficient condition for quantization errors to be uniform and white. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 25:442–448, 1977.
- [87] Internet World Stats. Internet user statistics. <http://www.internetworldstats.com/stats.htm>, 2009.
- [88] J.N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Dept. of Electrical Engineering and Computer Science, M.I.T., Boston, MA, 1984.
- [89] A. Vosughi and A. Scaglione. Precoding and decoding paradigms for distributed data compression. *IEEE Transactions on Signal Processing*, 75:1445–1460, April 2007.
- [90] A. D. Wyner. The rate-distortion function for source coding with side information at the decoder-ii: General sources. *Information and Control*, 38:60–80, 1979.
- [91] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with

- side information at the decoder. *IEEE Trans. Inform. Theory*, IT-22:1–10, Jan 1976.
- [92] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53:65–78, 2004.
- [93] L. Xiao, S. Boyd, and S. Kim. Distributed average consensus with least-mean-square deviation. In *Mathematical Theory of Networks and Systems Proc.*, pages 158–167, Kyoto, Japan, July 2006.
- [94] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the Information Processing in Sensor Networks*, Los Angeles, CA, April 2005.
- [95] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the IEEE/ACM Int. Symp. on Inf. Proc. in Sens. Netw.*, Los Angeles, CA, April 2005.
- [96] Yong Yao and Johannes Gehrke. Query processing for sensor networks, 2003.
- [97] M. E. Yildiz and Acaglione. The limiting rate behavior and rate allocation strategies for average consensus problems with bounded convergence. In *IEEE ICASSP*, Las Vegas, NV, April 2008.
- [98] M. E. Yildiz and A. Scaglione. Supplement of coding with side information for rate constrained consensus. <http://www.people.cornell.edu/pages/mey7/supplementCSIRCC>.
- [99] M. E. Yildiz and A. Scaglione. On the asymptotic scalability of the consensus algorithm. In *IEEE Statistical Signal Processing Workshop Proc.*, pages 645–649, August 2007.
- [100] M. E. Yildiz and A. Scaglione. Coding with side information for rate constrained consensus. *IEEE Transactions on Signal Processing*, July 2008.
- [101] M. E. Yildiz and A. Scaglione. Computing along routes via gossiping. *To Appear in IEEE Transactions on Signal Processing*, 2010.
- [102] Mehmet E. Yildiz and Anna Scaglione. Differential nested lattice encoding for consensus problems. In *Proceedings of the Information Processing in Sensor Networks*, Cambridge, MA, April 2007.

- [103] Mehmet E. Yildiz and Anna Scaglione. Directed gossiping for distributed data aggregation. In *IEEE CAMSAP*, December 2009.
- [104] Mehmet E. Yildiz, Anna Scaglione, and Tuncer C. Aysal. Computing along routes via gossiping. In *Information Theory Workshop*, October 2009.
- [105] R. Zamir and M. Feder. On lattice quantization noise. *IEEE Trans. on Information Theory*, 42:1152–1159, 1996.
- [106] R. Zamir and S. Shamai. Nested linear/lattice codes for wyner-ziv encoding. In *IEEE Inform. Theory Workshop Proc.*, pages 158–167, Killarney, Ireland, 1998.
- [107] Feng Zhao, Jie Liu, Juan Liu, Leonidas Guibas, and James Reich. Collaborative signal and information processing: An information directed approach. In *Proceedings of the IEEE*, 2003.