# Bounds for Max Consensus in Wireless Networks

*Aida Nowzari*

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

August 2015

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.



2015/08/05

## Abstract

Reaching consensus in distributed systems is a fundamental problem. In consensus problems, we would like all of the nodes of a network to converge to a certain quantity or a function of their values using only local communications. In the maximum value consensus problem, the objective of these communications is that all the nodes converge to the maximum of their initial values. There are two existing algorithms for the maximum value consensus problem in asynchronous networks: *random pairwise max* and *random broadcast max*. In the literature, bounds on the mean convergence time of these algorithms have been derived. In this thesis, we find tighter bounds on the expected convergence time of the random pairwise max and random broadcast max algorithms. Furthermore, we modify the proposed bounds for the cases in which the messages are successfully received probabilistically; e.g., to model packet drops. Performance analysis indicates that the bounds proposed in this thesis are significantly tighter than the previous state-of-the-art bounds. Simulation results show improvements of up to 95%.

## Abrégé

Parvenir à un consensus en utilisant des systèmes distribués est un problème fondamental. Lors d'un problème de consensus, l'idéal serait que tous les nœuds d'un réseau ce convergent, soit vers une certain quantité ou une fonction de sa valeur en utilisant seulement une communication locale. Lors d'un problème de consensus de valeur maximale, l'objectif de ces communications est la convergence de tous les nœuds vers leur valeur initiale maximale. Il y a deux algorithmes qui parviennent à résoudre le problème de consensus de valeur maximale: *random pairwise max* et *random broadcast max*. Une limite superieure de la moyenne de temps de convergence de ces deux algorithmes a déjà été dérivée et ces calculs se retrouvent dans la littérature académique. Cette thèse propose une limite plus stricte pour la durée de convergence des algorithmes *random pairwise max* et *random broadcast max*. De plus, nous modifions les contraintes suggérées pour les cas où les messages sont reçus avec succès (par example, si la réception d'un paquet est interrompue). Une analyse de performance indique que les contraintes suggérées dans cette thèse sont notablement plus restreintes que dans d'autres contraintes de dernier cri. Les résultats des simulations effectuées indiquent une amélioration allant jusqu'à 95%.

## Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Michael Rabbat, for giving me the opportunity of experiencing an innovative research in computer networks field and for his support and constructive feedbacks during the course of the research.

I am thankful to Professor Mark Coates for teaching the Telecommunication Network Analysis course in an interesting fashion that motivated me to dig into this field.

I am deeply grateful to my wonderful husband, Salim, for his endless love and care which kept me motivated during my studies and research.

I wish to express my warm thanks to my family, my mother Tahereh, my brothers Ali and Adel, and my cousin Armaghan. Thank you for your love, support, and unwavering belief in me.

I am grateful to my lab mates who provided a friendly environment. Special thanks to Milad, and thanks to Shohreh, Naghmeh, Babak, Ioannis, Guillaume, Yunpeng, Zhe, Santosh, Sean, Jun, and Syamantak.

# Contents


**1 Introduction** **1**

1.1 Consensus Problems . . . 1
1.2 Motivation . . . 2
1.3 Thesis Contribution . . . 2
1.4 Thesis Organization . . . 3

**2 Problem Formulation and Notation** **4**

2.1 Overview . . . 4
2.2 Graph Notations . . . 4
2.3 Asynchronous Time Model . . . 6
2.4 Maximum Value Consensus Problem . . . 6
2.5 Maximum Value Spreading Versus Rumor Spreading . . . 7
2.6 Graphs of Interest . . . 8
2.6.1 Grid networks . . . 8
2.6.2 Random geometric graphs . . . 8
2.6.3 Torus networks . . . 9
2.7 Summary . . . 10

**3 Previous Work** **11**

3.1 Overview . . . 11
3.2 General Survey on the Consensus Problems . . . 11
3.3 Max Consensus in a Max-Plus Algebraic Setting . . . 12
3.4 Distributed Algorithms for Reaching Consensus on General Functions . . . 13
3.5 Analysis of Max-Consensus Algorithms in Wireless Channels . . . 13
3.5.1 The Random-Walk-Max Algorithm . . . 13

3.5.2 The Random-Pairwise-Max Algorithm . . . . . . . . . . . . . . . . 14
3.5.3 The Random-Broadcast-Max Algorithm . . . . . . . . . . . . . . . 16
3.5.4 Random-Pairwise-Max Versus Random-Broadcast-Max . . . . . . . 18
3.6 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

**4 Bounds on the RPM Algorithm in Reliable Networks 21**
4.1 Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
4.2 Random Pairwise Max on Grids . . . . . . . . . . . . . . . . . . . . 21
4.2.1 Expected convergence time upper bound . . . . . . . . . . . . . . 23
4.2.2 Expected convergence time lower bound . . . . . . . . . . . . . . 26
4.3 Random Pairwise Max on Gridded Toruses . . . . . . . . . . . . . . 27
4.3.1 Expected convergence time upper bound . . . . . . . . . . . . . . 28
4.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

**5 Bounds on the RBM Algorithm in Reliable Networks 31**
5.1 Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
5.2 Random Broadcast Max on Random Geometric Graphs . . . . . . . . 31
5.2.1 Expected convergence time upper bound . . . . . . . . . . . . . . 34
5.2.2 Expected convergence time lower bound . . . . . . . . . . . . . . 35
5.3 Random Broadcast Max on Toruses . . . . . . . . . . . . . . . . . . 35
5.3.1 Expected convergence time upper bound . . . . . . . . . . . . . . 36
5.4 Random Broadcast Max on Grids . . . . . . . . . . . . . . . . . . . 36
5.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

**6 Bounds on the Max Consensus Algorithms in Unreliable Networks 38**
6.1 Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
6.2 RPM Algorithm in Networks With Unreliable Links . . . . . . . . . . 38
6.2.1 Modifying the bound in [1] for RGGs . . . . . . . . . . . . . . . . 39
6.2.2 Modifying the bound in Section 4.2 for grids . . . . . . . . . . . . 40
6.3 RBM Algorithm in Networks With Unreliable Links . . . . . . . . . . 41
6.3.1 Modifying the bound in [1] for RGGs . . . . . . . . . . . . . . . . 41
6.3.2 Modifying the bound in Section 5.2 for RGGs . . . . . . . . . . . . 42
6.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

**7 Experimental Results** **45**
7.1 Overview . . . . . . . . . . 45
7.2 Random Pairwise Max Algorithm on Grids . . . . . . . . . . 45
7.3 Random Pairwise Max Algorithm on Toruses . . . . . . . . . . 47
7.4 Random Broadcast Max Algorithm on RGGs . . . . . . . . . . 48
7.5 Random Broadcast Max Algorithm on Toruses . . . . . . . . . . 51
7.6 RBM Algorithm on RGG With Link Failure . . . . . . . . . . 52
7.7 Summary . . . . . . . . . . 52

**8 Conclusion and Future Work** **54**
8.1 Concluding Remarks . . . . . . . . . . 54
8.2 Future Work . . . . . . . . . . 55

**References** **57**

# List of Figures

2.1 Grid network of size 100. . . . . 8
2.2 Construction of a RGG. . . . . 9
2.3 Gridded torus. . . . . 10

3.1 Mean number of communications and associated bound in RPM algorithm 15
3.2 Spanning tree of a graph . . . . . 17
3.3 Mean number of communications and associated bound in RBM algorithm 18
3.4 Percentage of informed nodes vs. the number of iterations . . . . . 19

4.1 General positioning of informed nodes on a grid . . . . . 23
4.2 Spiral growth of the shape. . . . . 24
4.3 Positioning the informed nodes using spiral approach. . . . . 25
4.4 Filling up row by row. Yellow dots influence $|\partial M^{\tau_i}|$. . . . . . 25
4.5 RPM on grid with max $|\partial M^{\tau_i}|$ . . . . . 26
4.6 Sample nodes are placed on the gridded torus. . . . . 27
4.7 Filling up row by row in a torus . . . . . 29
4.8 Building square in a torus . . . . . 29

5.1 Dividing the unit square into bins. . . . . . 32
5.2 Connectivity of RGGs with $r_0 = \sqrt{10}$ . . . . . 33
5.3 $\partial M^{\tau_i}$ in steps . . . . . 36

7.1 Mean of the number of communications for RPM in grids . . . . . 46
7.2 Proposed bounds for the RPM algorithm on grids . . . . . 47
7.3 Random pairwise max on gridded torus . . . . . 49
7.4 Mean of the number of communications for RBM in RGG . . . . . 50

7.5 Random broadcast max on torus . . . . . . . . . . . . . . . . . . . . . . . 51
7.6 RBM on RGG with link failure . . . . . . . . . . . . . . . . . . . . . . . 53

# List of Tables

7.1 Performance of different upper bounds for RPM in grids . . . . . . . . . . . 48
7.2 Performance of different upper bounds for RBM in RGG . . . . . . . . . . . 50

# List of Acronyms

| | |
|---|---|
| WSN | Wireless Sensor Network |
| RGG | Random Geometric Graph |
| RPM | Random Pairwise Max |
| RBM | Random Broadcast Max |
| CPU | Central Processing Unit |

# Chapter 1

# Introduction

## 1.1 Consensus Problems

Reaching consensus in distributed systems is a fundamental issue due to its various applications. In a consensus problem, the objective is that all the agents of a network agree upon a certain quantity or a function of their values using only local interactions (local information exchange).

In the maximum value consensus problem, the network nodes have initial values, and the network needs to reach consensus on the maximum of the initial values. This type of consensus problem has been studied more recently in asynchronous networks [1, 2]. The focus of this thesis is on the bounds for the expected convergence time of maximum value consensus algorithms in connected, undirected, and asynchronous networks. Being asynchronous means that the nodes of a network do not have a common clock. Instead, each node has its own clock and can start a communication with its neighbors at its own clock ticks [3, 4].

There are several algorithms in the literature proposed for the maximum value consensus problem [1]. The performances of these algorithms are evaluated via three criteria: the mean convergence time, the convergence time dispersion, and the number of communications.[1] Since the network reaches consensus via local information exchanges, it is expected that the performance of these algorithms depends on the size of the network, the diameter of the network, and the degrees of the nodes of the network.

[1] Formal definitions for these metrics will be given in Chapter 2.

## 1.2 Motivation

The maximum value consensus problem is important to study because of its practical uses. As an example, consider a network in which the nodes use the same medium to send information. Hence, the network should decide which agent has to transmit information. To solve this problem, the nodes are asked to draw a number in a common window, and then reach consensus on the greatest drawn number. Then the node with the maximal value can use the medium [1].

The previous example is analogous to the *leader election* problem, which also uses the max consensus algorithm. The network has to decide who takes the role of the leader. For example, if each agent has a performance index, then all the nodes will compare their indices and the node with the greatest performance index will be selected as the leader [5].

Another practical use is when the wireless network has to transmit data on a regular basis. An intuitive method to achieve this without possible node failure is to choose the node with the maximum remaining power to perform the communications. Therefore, the network should be able to identify the agent that has the maximum remaining power. This will be feasible if the network reaches consensus on the maximal value of the remaining power of the agents [1].

*Minimum time rendezvous* is another application of the max consensus problem. Assume that a group of agents are supposed to meet each other at a specific time and each agent needs a *minimum time* to get to the meeting place. Therefore, the group should reach consensus on the maximum of the minimum times, in order to arrange the meeting time [5–7].

## 1.3 Thesis Contribution

We study algorithms for the maximum value consensus problem in wireless networks. We derive tight bounds for the expected convergence times of these algorithms. Specifically, various novel bounds for the mean convergence time of the maximum value consensus algorithms in asynchronous networks are developed. These bounds are significantly tighter than the bounds proposed in [1]. We then extend our work to address the possible presence of link failures. These results are validated via simulations.

## 1.4 Thesis Organization

The organization of this thesis is summarized below.

In Chapter 2, we will summarize the essential elements from graph theory that are employed throughout this document. Then, the asynchronous time model for the distributed systems is presented. Moreover, the maximum value consensus problem is discussed formally along with its performance measures. The graphs (grids, toruses, and random geometric graphs) on which we study the maximum value consensus problem are also described in Chapter 2.

In Chapter 3, a general survey of consensus and gossip algorithms is reported. The chapter is then followed by a review of the existing work in the literature concerning the maximum value consensus problem.

Novel bounds for the expected convergence time of the random pairwise max algorithm in reliable networks are presented in Chapter 4.

In Chapter 5, novel bounds for the mean convergence time of the random broadcast max algorithm in reliable networks are developed.

The probability of link failure is introduced to the analysis in Chapter 6, and the proposed bounds in Chapters 4 and 5 are generalized to account for the possibility of messages being dropped.

The performance comparison of our proposed bounds for the mean convergence time of the maximum value consensus algorithms and the previous state-of-the-art bounds is presented in Chapter 7.

This thesis is summarized by the concluding remarks and suggestions for future work in Chapter 8.

# Chapter 2

# Problem Formulation and Notation

## 2.1 Overview

In this chapter, the notation that is required to state the maximum value consensus problem is introduced. Then, the asynchronous time model for the distributed systems is presented. We then state the problem formulation and the underlying performance measures. Furthermore, the difference between the *maximum value spreading* and *rumor spreading* problems is summarized. In the last section of this chapter, the different types of graphs that are used in our study are described.

## 2.2 Graph Notations

Let $G = (V, E)$ be an undirected graph with $N$ nodes, where $V$ is the set of vertices and $E$ is the set of edges of the graph. The edges of the graph are represented as an ordered pair $(v_i, v_j)$ if there exists a path of length one from node $v_i$ to node $v_j$, where $v_i$, $v_j \in V$. In undirected graphs:

$$(v_i, v_j) \in E \Longleftrightarrow (v_j, v_i) \in E. \tag{2.1}$$

The set of neighbors of the node $v_i$, denoted by $L_{v_i}$, consists of the nodes that vertex $v_i$ is connected to:

$$L_{v_i} = \{v_j \in V : (v_i, v_j) \in E\}. \tag{2.2}$$

For any set $M$, we denote its cardinality by $|M|$. Therefore, $|V| = N$. The degree of the node $v_i$ of a graph, denoted by $d_{v_i}$, is equal to the number of edges that connect $v_i$ to the other nodes of the graph, i.e., $d_{v_i} = |L_{v_i}|$.

The maximum degree of $G$, denoted by $d_{max}$, and the minimum degree of $G$, denoted by $d_{min}$, are the maximum and minimum degrees of its vertices, respectively.

The degree matrix of a graph, denoted by $D$, is a diagonal matrix of size $N \times N$, which contains information about the degree of each vertex, i.e.,

$$d_{i,j} = \begin{cases} d_{v_i} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

The adjacency matrix of a graph, denoted by $A$, represents which nodes of the graph are connected (adjacent) to each other, i.e.,

$$a_{i,j} = \begin{cases} 1 & \text{if there exists an edge between node } v_i \text{ and node } v_j \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

The Laplacian matrix of a graph, denoted by $L$, is the difference of the degree matrix and the adjacency matrix, i.e., $L = D - A$. From the definition it follows that:

$$l_{i,j} = \begin{cases} d_{v_i} & \text{if } i = j \\ -1 & \text{if } \ i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

The eigenvalues of the Laplacian matrix have many applications in algebraic graph theory. The Laplacian matrix has two significant useful properties [8]:

- it is positive semidefinite, and;
- its row sum is null.

Therefore $\overrightarrow{1}_N$ (the column vector of ones of size $N$) is an eigenvector associated with eigenvalue 0, and the eigenvalues of the Laplacian matrix satisfy $0 = \lambda_1^L \leq \lambda_2^L \leq \lambda_3^L \cdots \leq \lambda_N^L$ (see [9], Th. 7.2.1). Since the graphs considered in this thesis are connected, the second smallest eigenvalue of the Laplacian matrix ($\lambda_2^L$) is strictly positive (see [10], Lemma 1.7).

The diameter of a graph $G$, denoted by $\Delta_G$, is the largest number of edges which must be traversed in order to travel from one vertex to another. Let $\text{dist}(v_i, v_j)$ denote the minimum number of edges that connect the node $v_i$ to the node $v_j$, then,

$$\Delta_G = \max\{\text{dist}(v_i, v_j) : (v_i, v_j) \in V^2\}. \tag{2.6}$$

## 2.3 Asynchronous Time Model

In this thesis, we assume that the nodes of the networks update their local estimate of the maximum value when they receive a message from a neighbor. The order in which nodes transmit is determined by local clocks. Each node has its own clock and can start a communication with its neighbors at its own clock ticks (asynchronous time model). We can assume no collisions happen between communicating nodes since the clock rate at each node can be chosen such that the communication time is significantly shorter compared to the time between clock ticks [3, 4].

If each node's clock ticks is modeled by an independent Poisson process, then a global clock modeled by a Poisson process with intensity $\lambda = \sum_v \lambda_v$ is obtained , where $\lambda_v$ is the Poisson process intensity of node $v$.

We assume that the networks are *connected*, *time invariant*, and *asynchronous*.

## 2.4 Maximum Value Consensus Problem

Using the definition in the previous section for clock ticks, we write $x_n(v)$ as the value of agent $v$ after $n$ global clock ticks. In the maximum value consensus problem, the objective is that all the nodes of the network converge to the maximum of the initial values, i.e., $x_n(u) = \max_{v\in V} x_0(v)$ as $n \to \infty$ for all $u \in V$.

Note that all the algorithms for the maximum value spreading easily generalize to the minimum value spreading problem.

The convergence time $\tau$ is the first time when all the nodes of the network are informed of the maximum of the initial values. This can be written as:

$$\tau = \inf\{n \in \mathbb{N} : \forall v \in V \;\; x_n(v) = \max_{v\in V} \; x_0(v)\}. \tag{2.7}$$

The performance of the algorithms proposed for the maximum value consensus problem is evaluated via three criteria:

- **the mean convergence time**: $\mathbb{E}[\tau]$ where the expectation is with respect to the random order of clock ticks,
- **the convergence time dispersion**: for all $\epsilon \in (0,1)$, finding a bound $B$ such that $Pr[\tau < B] \geq 1 - \epsilon$, gives a general view of the convergence speed, and,
- **the number of communications**: it is obtained by multiplying the number of iterations by the number of communications per iteration.

## 2.5 Maximum Value Spreading Versus Rumor Spreading

There is a difference between the asynchronous maximum value spreading and the *rumor spreading* problems which is worth noting [2]. In the rumor spreading problem, the goal is to spread a local information through the network. Most research papers that are concerned with the rumor spreading consider push and/or pull transmission methods. In the push transmission method, information gets transferred from the caller to the called node. In the pull transmission method, information gets transferred from the called to the calling node [11].

Generally, the push and pull transmission methods are combined to form the *push and pull* algorithms [12–19]. These algorithms are based on one-by-one exchanges between neighbors in the network and are applicable to wired networks.

There are also some research papers that consider *radio* medium for the network communications and propose *broadcast* based algorithms, e.g., [20–23].

Although the maximum value spreading and the rumor spreading problems seem similar, there is a major difference between them. In the maximum value spreading, no sensor is aware if it has the maximum value, while in the rumor spreading, the sensors are aware if they have the rumor or not.

We can consider the maximum value spreading problem as an *asynchronous juiciest rumor spreading* problem, where *juiciest* means that the nodes do not know if they have the rumor [2].

## 2.6 Graphs of Interest

In this section we describe three types of graphs on which we study the maximum value consensus problem. Grids, toruses, and random geometric graphs are the network models that we use in our work. The following subsections provide detailed description of these models.

### 2.6.1 Grid networks

Grid structure is widely used since it provides services that cannot be supported with the traditional structures. Grid networks provide scalable, reliable, and secure mechanisms for assembling, integrating and utilizing multiple resources. These resources may include specialized computers, computer clusters, instruments, or sensors. Some of the general grid characteristics are: resource sharing, flexibility, programmability, and dynamic integration. Further information on the grid networks is provided in [24].

Assume that you have a network of size $N$, where $N$ is a power of two. The grid network that represents this network is a grid of size $\sqrt{N} \times \sqrt{N}$, as illustrated in Figure 2.1.

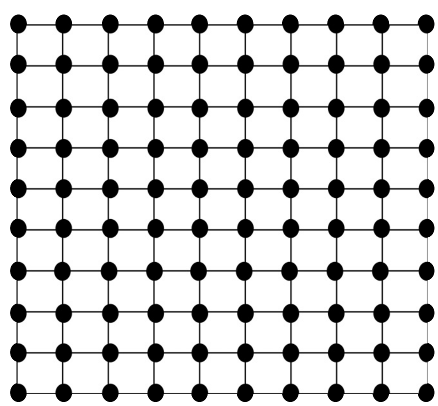


**Fig. 2.1** Grid network of size 100.

### 2.6.2 Random geometric graphs

Another type of graphs on which we study the maximum value consensus problem is random geometric graphs. In random geometric graphs, two nodes are connected if the Euclidean distance between them is smaller than or equal to a predefined radius $r$ [4].

In wireless networks, two nodes can communicate or are connected to each other if they are close (in each others' range of broadcast). If we set the radius $r$ equal to the wireless network range of broadcast, then random geometric graphs will model the wireless networks.

To generate a random geometric graph (RGG) with $N$ nodes, $N$ points are chosen uniformly at random in the unit square $[0, 1] \times [0, 1]$ representing the positions of the nodes. Then, an undirected edge is drawn between any pair of nodes if their Euclidean distance is less than $r$.

Note that $r$ represents the communication radius of a node. By choosing $r = r_0\sqrt{\frac{\log N}{N}}$ with suitable $r_0$, the random geometric graph will be connected with high probability [25, 26]. A random geometric graph with ten nodes is illustrated in Figure 2.2.

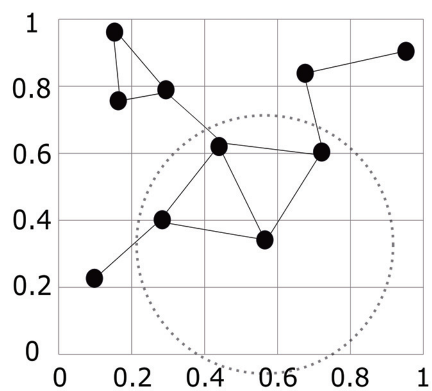



**Fig. 2.2** Construction of a RGG.

### 2.6.3 Torus networks

Torus networks are important to study because they eliminate the border effect of the grid networks and the random geometric graphs [27]. For instance, in the grid networks and the random geometric graphs, the node on the upper left corner cannot easily reach the node in the lower left corner, while in the corresponding torus networks, communication from the upper nodes to the lower nodes is possible. A gridded torus is shown is Figure 2.3.

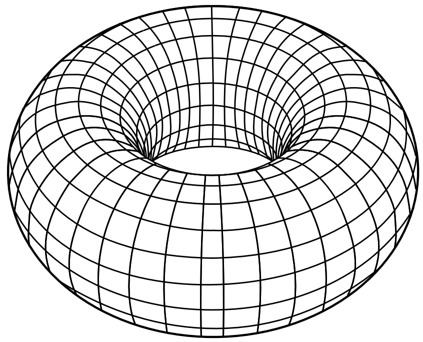


**Fig. 2.3** Gridded torus.

## 2.7 Summary

In this chapter, we introduced the notation that we will use throughout the thesis. We then described the asynchronous time model for distributed systems. Moreover, the maximum value consensus problem and the performance measures were formally discussed. Then, the difference between the maximum value spreading and rumor spreading problems was explained. Grids, random geometric graphs, and toruses which are the network models considered in this thesis were also introduced in this chapter.

# Chapter 3

# Previous Work

## 3.1 Overview

In this chapter, we first provide a brief literature review of the consensus problems. Then, two research papers in the literature analyzing the maximum value consensus problem in the synchronous and the continuous time models are reviewed. The discussion is then followed by a detailed review of a paper on the maximum value consensus problem algorithms in the asynchronous time model and the algorithms' performance metrics.

## 3.2 General Survey on the Consensus Problems

There are many research articles in various branches of the literature which study the consensus problems in distributed systems. For instance, in 1974, DeGroot in [28] developed a model describing how a group of nodes reach consensus by forming a common probability distribution for a parameter whilst each of which initially has a different probability distribution for the unknown value of the parameter.

Another efficient and highly fault-tolerant consensus algorithm, named *PAXOS* was devised by Lamport [29] for reaching consensus in a distributed system. An exclusive survey of early results is provided in [30]. More recent studies concerning the consensus problem are the research papers on *randomized gossip*, e.g., [4, 31, 32].

Note that the consensus problem is also known as the *agreement problem* in social networks, economics, and signal processing fields. In statistical mechanics, the consensus problem is called *synchronization* [33].

Consensus to the average of the initial values of the agents has been extensively studied in the past years, due to its practical applications. A simple idea for the average consensus algorithm would be to wake the agents up randomly and average their values with another node. After a sufficient time, all the agents will have a close approximation of the average of the initial values [3, 25, 34–36].

Another type of consensus problem which is being studied more recently is the maximum value consensus problem [1, 5, 36]. In the maximum value consensus problem, all the nodes of the network reach consensus on the exact maximum of the initial values in a finite time. However, in the average value consensus problem, the network does not converges to the exact average value in a finite time.

Due to the lack of a finite-time convergence, the distributed average consensus algorithm cannot be adopted to obtain a unified measurement across all the nodes in real-life situations. The maximum value consensus algorithms, however, could remedy this problem; the following hybrid approach can be employed in order to distribute a unified measurement amongst all the network nodes, in a finite time:

- the distributed average consensus algorithm is forcefully stopped at a predetermined time-step; when the average consensus algorithm stops, every node possesses an approximate average value.

- at this point, a maximum value consensus algorithm starts to distribute the maximum of the approximate average values. This step does converge in a finite time, and as a result, all the nodes will ultimately possess an identical approximate average value.

All the following sections provide a general overview of the existing research work studying the maximum value consensus problem.

## 3.3 Max Consensus in a Max-Plus Algebraic Setting

Nejad et al. [5] study the maximum value consensus problem. They have proposed to use max-plus algebra to analyze the max consensus algorithms in a synchronous-time model. They represented the max consensus algorithm by a linear system and studied the convergence conditions and the convergence rates.

## 3.4 Distributed Algorithms for Reaching Consensus on General Functions

Another research paper which presents a method for reaching consensus on the maximum/minimum value is by Cortes et al. [36]. Necessary and sufficient conditions characterizing any algorithm that achieves consensus asymptotically are obtained in this paper.

They consider continuous consensus functions of the initial state of the network agents, where the nodes update their values and communicate with the neighbors continuously over time.

## 3.5 Analysis of Max-Consensus Algorithms in Wireless Channels

Iutzeler et al. [1, 2] are the first to study the maximum value consensus problem in the networks with the asynchronous time model. For the rest of this chapter, we will summarize the findings of these articles.

### 3.5.1 The Random-Walk-Max Algorithm

The random walk max algorithm considers a simple random walk on the graph, propagating the maximum encountered value. This algorithm is not asynchronous nor synchronous and is used as a comparison point. The steps of the random walk max algorithm are summarized as follows:

1. Node $v$ (randomly selected) wakes up after the $n^{th}$ clock tick.
2. $v$ chooses a neighbor $w$ uniformly random in $L_v$.
3. $x_{n+1}(w) = \max\left(x_n\left(v\right), x_n\left(w\right)\right)$.
4. $w$ is the active node for the time $n + 1$.

Assume that the random walk algorithm begins from node $v_i$, while node $v_j$ has the maximum initial value. Therefore, the convergence time of the random walk algorithm is the sum of two times:

- $(v_i, v_j)$**-hitting time**: the time the random walk takes to go from $v_i$ to $v_j$, and,

- $v_j$**-cover time**: the time that the algorithm takes to go to all other nodes of the graph starting from $v_j$.

The cover time of the random walk algorithm is well studied in [37, 38].

### 3.5.2 The Random-Pairwise-Max Algorithm

The random pairwise max (RPM) algorithm mimics the *random gossip* algorithm for averaging introduced in [31]. At each time step, a random node wakes up and exchanges its value with another node selected uniformly at random among its neighbors. Then both nodes keep the maximum of their previous and received values. The steps of the RPM algorithm are summarized as follows:

1. Node $v$ (randomly selected) wakes up after the $n^{th}$ clock tick.
2. $v$ chooses a neighbor $w$ uniformly random in $L_v$.
3. $x_{n+1}(v) = x_{n+1}(w) = \max\left(x_n\left(v\right), x_n\left(w\right)\right)$.

This algorithm is particularly suitable for wired networks because it does not need the network to have broadcasting ability. The random pairwise max algorithm converges almost surely to a consensus on the maximum value in a finite time. The two performance metrics (the mean convergence time, and the convergence time dispersion) of this algorithm are presented in the following subsections.

#### Expected convergence time

The mean of the convergence time of the random pairwise max algorithm is proportional to the number of nodes ($N$), the maximum degree of the graph ($d_{max}$), and the harmonic number of order $N-1$, and is inversely proportional to the second smallest eigenvalue of the Laplacian matrix ($\lambda_2^L$). Iutzeler et al. [1] show that:

$$\mathbb{E}[\tau] \leq N d_{max} \frac{h_{N-1}}{\lambda_2^L}, \tag{3.1}$$

where $h_{N-1} = \sum_{k=1}^{N-1} \frac{1}{k}$ is the $(N-1)^{th}$ harmonic number.

Let us describe the idea of the proof briefly. Consider the set $M^k$ of nodes which are informed of the maximum value at time $k$. The probability that a new node will be informed of the maximum value at time $k+1$, is equal to the probability that one node in $M^k$ exchanges value with another node in $V \backslash M^k$. Following this approach until all the nodes of the network are informed, we will reach consensus on the maximum value.

By finding a bound for the $\Pr[X^{k+1} = X^k + 1 | M^k]$, where $X^k = |M^k|$, and performing number of mathematical substitutions, Equation 3.1 is proven.

Figure 3.1 compares the mean convergence time and its bound from Equation 3.1. Simulations were run on 100 connected random geometric graphs, and 100 times for each graph, for every reported graph size.

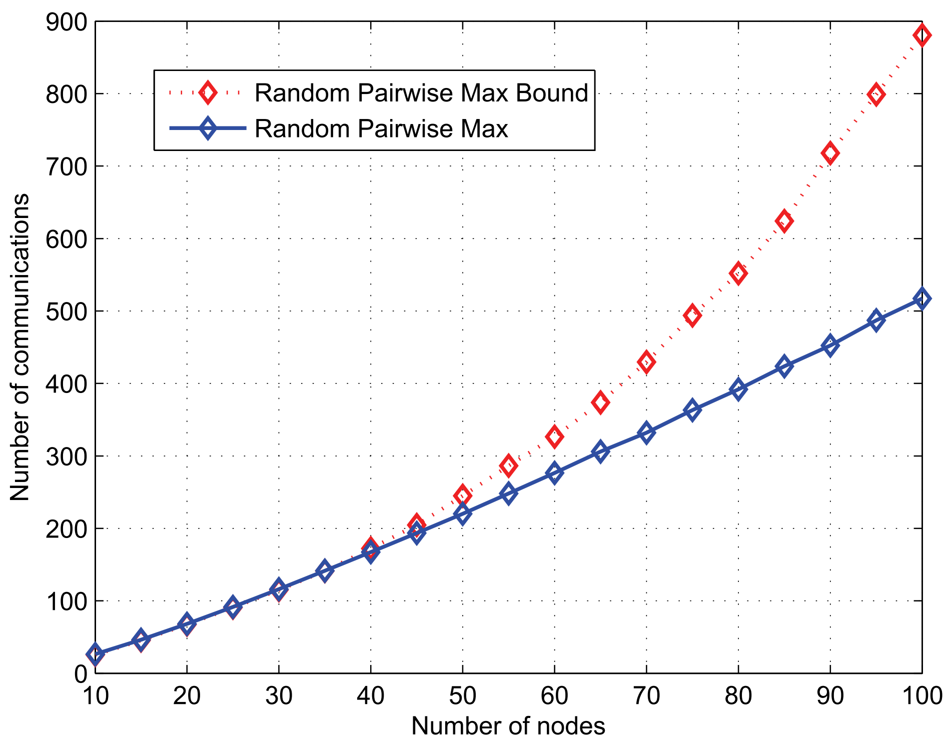



**Fig. 3.1** Mean number of communications and associated bound in RPM algorithm versus $N$.

#### Convergence time dispersion

Iutzeler et al. [1] also prove that for the random pairwise max algorithm, with probability $1-\epsilon$,

$$\tau \leq C(N)\left(1+\log\frac{N}{\epsilon}\right), \tag{3.2}$$

where $C(N)$ is the mean convergence time bound.

If $\epsilon = \frac{1}{N}$, then $\tau = \mathcal{O}\left(C(N)\log(N)\right)$ with probability of $1-\frac{1}{N}$.

### 3.5.3 The Random-Broadcast-Max Algorithm

The random broadcast max (RBM) algorithm takes advantage of the broadcasting capability of the wireless network. In this algorithm, at each clock tick, a random node wakes up and broadcasts its value to all its neighbors. Then the agents which have received the information, update their values to the maximum between their original and received values. The steps of the random broadcast max algorithm are summarized as follows:

1. Node $v$ (randomly selected) wakes up after the $n^{th}$ clock tick.
2. $v$ broadcasts its current value to all its neighbors.
3. $x_{n+1}(w) = \max\left(x_n(v), x_n(w)\right)$ for all $w \in L_v$.

As expected, this algorithm results in a much faster consensus than the random walk max and random pairwise max algorithms because it benefits from the broadcasting ability of the network. This algorithm also converges almost surely to a consensus on the maximum value in a finite time.

#### Expected convergence time

Iutzeler et al. [1] prove the following upper bound for the mean of the convergence time:

$$\mathbb{E}[\tau] \leq N\Delta_G + N(\Delta_G - 1)\log\left(\frac{N-1}{\Delta_G - 1}\right), \tag{3.3}$$

where $\Delta_G$ is the diameter of the graph.

While the complete proof can be found in [1], it is beneficial to go over the proof briefly.

The bound for the mean convergence time is computed on a spanning tree of the graph $G$, which is rooted at the node that has the maximum value at time 0, as shown in Figure 3.2. Noting that the agent with the maximum value is located at the root of the tree, the expected time it needs to inform all its neighbors (called layer one) is counted. Once the first layer is completely informed, the expected time needed for the first layer to inform the second layer is calculated.

This course of action continues until the sensors in the last layer of the spanning tree, which are at the distance of at most $\Delta_G$ from the root, are informed. Then all these expected times are added together to form the upper bound given in Equation 3.3.

For a complete graph, where $\Delta_G = 1$, the upper bound simplifies to $\mathbb{E}[\tau] \leq N$ since the time needed for the propagation of the maximum value is equal to the time it takes for the node with the maximum value to wake up and broadcast its value to all other nodes of the network, which is $N$ in expectation.

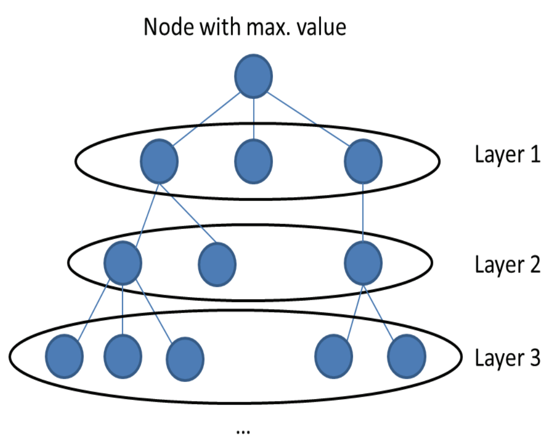



**Fig. 3.2** Spanning tree of a graph rooted at the agent with the maximum value

Figure 3.3 compares the mean convergence time and its bound from Equation 3.3. Simulations were run on 100 random geometric graphs, and 100 times for each graph, for every graph size.

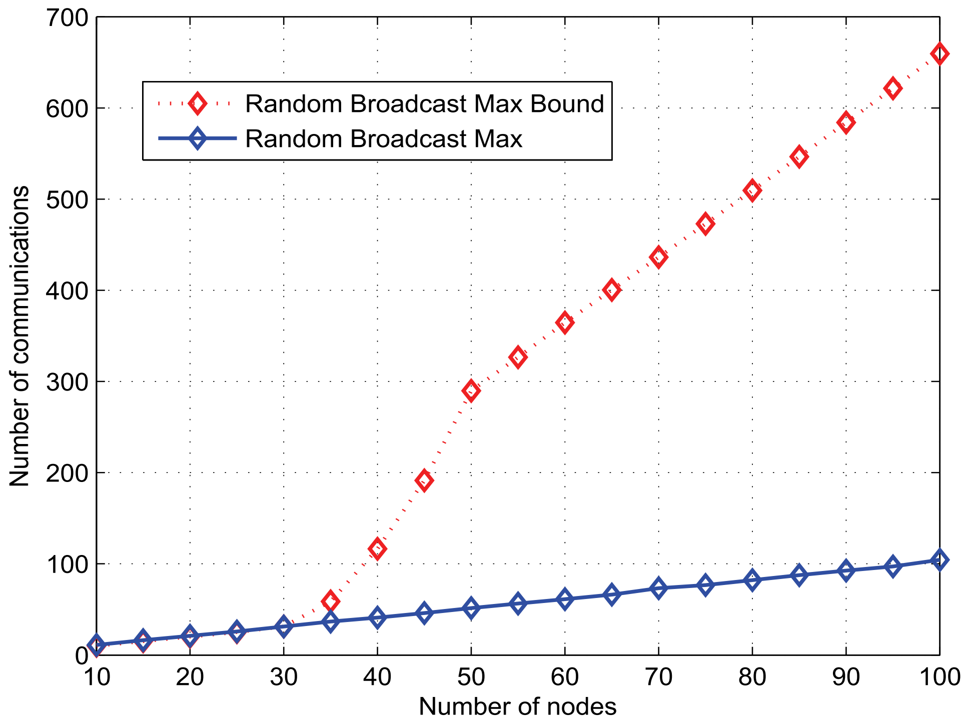



**Fig. 3.3** Mean number of communications and associated bound in RBM algorithm versus $N$.

### Convergence time dispersion

It is proven that in the random broadcast max algorithm, with probability $1-\epsilon$,

$$\tau \leq C(N) + N\Delta_G \left( \log(\frac{\Delta_G}{\epsilon}) - 1 \right), \tag{3.4}$$

where $C(N)$ is the mean convergence time bound.

For complete graphs, where $\Delta_G = 1$, if $\epsilon = \frac{1}{N}$, then $\tau = \mathcal{O}\left(N \log(N)\right)$ with probability of $1 - \frac{1}{N}$.

## 3.5.4 Random-Pairwise-Max Versus Random-Broadcast-Max

The random broadcast max algorithm improves the convergence speed notably by taking advantage of the broadcasting nature of the network. In Figure 3.4, the percentage of the informed nodes versus the number of iterations for a graph of size 50 is shown.

As expected, the random broadcast max algorithm converges faster than the other two algorithms. It is noticeable that for the first 150 iterations, the random walk algorithm outperforms the random pairwise max algorithm, but gradually slows down after that and is the slowest algorithm for higher number of iterations. This behavior can be explained by the fact that random walk informs a new node very often at first steps, but the last sensors take a long time to inform (random walk is a local algorithm).

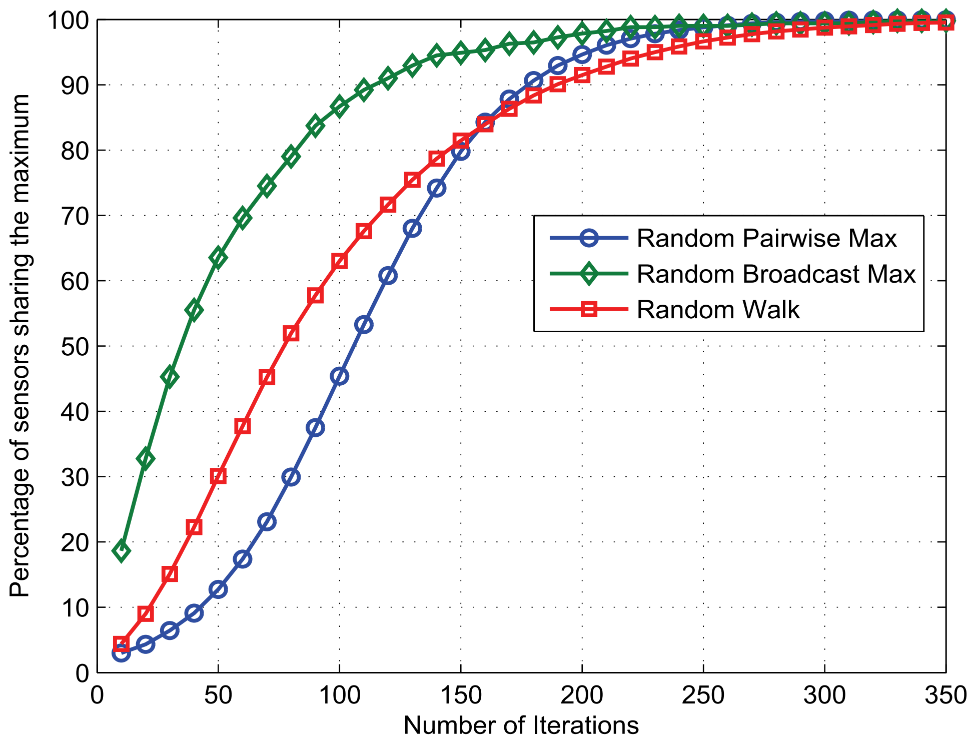



**Fig. 3.4** Percentage of informed nodes versus the number of iterations.

## 3.6 Summary

Several research papers on the maximum value consensus problem were summarized in this chapter. Furthermore, a detailed review of the random pairwise max and random broadcast max algorithms was presented.

We also mentioned that the random broadcast max algorithm reaches consensus substantially faster than the random pairwise max algorithm and is a more suitable candidate to be used in the networks with broadcasting capability, such as wireless networks.

The proposed bounds in [1] for the expected convergence time of the random pairwise max and random broadcast max algorithms were also studied in details in this chapter.

# Chapter 4

# Bounds on the RPM Algorithm in Reliable Networks

## 4.1 Overview

In this chapter, we propose new upper and lower bounds on the expected convergence time of the random pairwise max algorithm on grid and torus networks. Note that reliable networks are the networks in which all the communications are successful (no link failure). The tightness of the proposed bounds significantly surpass the bounds from [1].

## 4.2 Random Pairwise Max on Grids

In this section, we derive an upper bound and a lower bound for the mean convergence time of the random pairwise max algorithm on grid networks. This facilitates the derivation of the bounds for the mean convergence time of the random broadcast max algorithm on random geometric graphs in Section 5.2.

Assume we have a grid of size $\sqrt{N} \times \sqrt{N}$ ($N$ nodes in total) and only one node is informed of the maximum value in the network at time 0. Let $M^k = \{v_i \in V : x_k(v_i) = \max_{v \in V} \; x_0(v)\}$ be the set of nodes that are informed of the maximum value at time $k$. We denote the cardinality of $M^k$ as $X^k$, i.e., $|M^k| = X^k$.

In the random pairwise max algorithm, the only possibility for an uninformed node to be informed of the maximum is to communicate with an informed neighbor. At each iteration either one new node is informed or no new node is informed, i.e., $X^k \leq X^{k+1} \leq X^k + 1$.

We find a tight bound for the probability that $X^{k+1} > X^k$.

Let $v_i$ and $v_j$ be connected nodes exchanging values at time $k$ of the algorithm. If $v_i$ is in $M^k$ and $v_j$ is in $V \backslash M^k$ (or vice-versa), then at time $k+1$ a new node will be informed. Thus,

$$\Pr[X^{k+1} = X^k + 1 | M^k] = \Pr[(v_i, v_j) \in \partial M^k | M^k], \tag{4.1}$$

where $\partial M^k$ is the set of edges with one end in $M^k$ and the other end in $V \backslash M^k$.

Edge $(v_i, v_j)$ is selected if node $v_i$ is clocked uniformly in $V$ (with probability $\frac{1}{N}$) and then node $v_j$ is chosen uniformly in $L_{v_i}$ (with probability $\frac{1}{d_{v_i}}$) or if $v_j$ is clocked and then $v_i$ is chosen from its neighbors. Therefore,

$$\Pr[(v_i, v_j) \text{ be selected}] = \frac{1}{N}\left(\frac{1}{d_i} + \frac{1}{d_j}\right). \tag{4.2}$$

Hence, we can write:

$$2\frac{1}{N}\frac{1}{d_{max}} \leq \Pr[(v_i, v_j) \text{ be selected}] \leq 2\frac{1}{N}\frac{1}{d_{min}}. \tag{4.3}$$

The probability of the edge $(v_i, v_j)$ being chosen in $\partial M^k$ is:

$$2\frac{|\partial M^k|}{N d_{max}} \leq \Pr[(v_i, v_j) \in \partial M^k | M^k] \leq 2\frac{|\partial M^k|}{N d_{min}}. \tag{4.4}$$

Since the second smallest eigenvalue of the Laplacian matrix ($\lambda_2^L$) gets very small for the grids compared to random geometric graphs, the bound in [1] (Equation 3.1) gets significantly loose for grids.

Let us develop Equation 4.4. For simplicity, we assume that at time 0, only one node has the maximum value. We define the stopping times $\tau_i = \inf\{k \in \mathbb{N} : X^k = i\}$ such that $\tau_1 = 0$ and the convergence time $\tau = \sum_{i=1}^{N-1}[\tau_{i+1} - \tau_i]$.

Let the random variable $Y^k$ be equal to $X^{k+1} - X^k$ given $X^k$. Therefore:

$$Y^k = \left[X^{k+1} - X^k | X^k\right] = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } 1-p. \end{cases}$$

where $\frac{|\partial M^k|}{N d_{max}} \leq p \leq 2\frac{|\partial M^k|}{N d_{min}}$.

Hence, $Y^k$ is Bernoulli distributed and its parameter is lower bounded by $2\frac{|\partial M^k|}{Nd_{max}}$ and upper bounded by $2\frac{|\partial M^k|}{Nd_{min}}$.

Similarly, the geometrically distributed random variable $Y^{\tau_i}$ is defined as $X^{\tau_i+1} - X^{\tau_i}$ given $X^{\tau_i}$, and $\tau_{i+1} - \tau_i$ is the number of trials on $Y^{\tau_i}$ to obtain success (a new node be informed) when $X^{\tau_i} = i$ (the number of iterations needed to have $X^{\tau_{i+1}} = X^{\tau_i} + 1$).

Hence, the expected value of the random variable $\tau_{i+1} - \tau_i$ is equal to $\frac{1}{p}$, the expected value of the geometrically distributed random variable $Y^{\tau_i}$. It follows that:

$$\frac{Nd_{min}}{2}\frac{1}{|\partial M^{\tau_i}|} \leq \mathbb{E}[\tau_{i+1} - \tau_i] \leq \frac{Nd_{max}}{2}\frac{1}{|\partial M^{\tau_i}|}. \tag{4.5}$$

We aim to find bounds for the mean convergence time by finding tight bounds for $|\partial M^{\tau_i}|$. Note that a lower bound for $|\partial M^{\tau_i}|$ leads to an upper bound for the mean convergence time.

### 4.2.1 Expected convergence time upper bound

Figure 4.1 shows a general positioning of the informed nodes, and the edges that are in $\partial M^{\tau_i}$ are highlighted in red. The perimeter of the polygon shape that the informed nodes generate has a direct impact on $|\partial M^{\tau_i}|$.

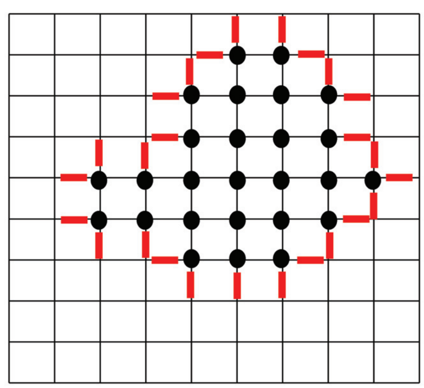


**Fig. 4.1** General positioning of informed nodes on a grid. Red lines are the edges in $\partial M^{\tau_i}$.

Therefore, to minimize $|\partial M^{\tau_i}|$, the perimeter of the polygon shape that the informed nodes create should be minimized. Harary et al. [39] state the problem when we are given $s$ unit squares and we want to build a shape with the minimum perimeter using these unit squares. They prove that the minimum perimeter that one can obtain is $2\lceil 2\sqrt{s}\rceil$. The algorithm which results in a polygon of the minimum perimeter follows spiral polygon construction, as illustrated in Figure 4.2 [39, 40]. This algorithm leads to a close-to-square polygon shape at every step.

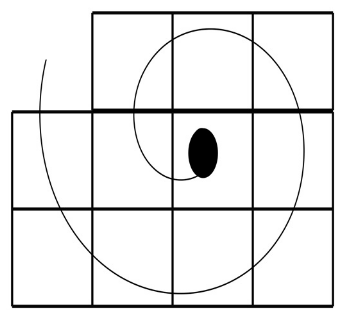


**Fig. 4.2** Spiral growth of the shape.

Since the algorithm proposed by Harary et al. [39] considers infinite space dimensions, it cannot be used in grids which are of the finite dimensions without some modifications.

We use this algorithm to position the informed nodes in the grid, i.e., for every number of informed nodes, $i$, we position the nodes usisg the spiral algorithm, then, we place the obtained shape on one of the corners of the grid, such that the sides with maximum length are placed on the borders.

Note that putting the polygon shape on the corner decreases $|\partial M^{\tau_i}|$ by at least half. Figure 4.3 shows the positioning of 26 informed nodes using the spiral approach. We name this approach as *building square* in this thesis.

Hence, we can write:

$$|\partial M^{\tau_i}| = \begin{cases} 2\sqrt{i} & \text{if } i \text{ is a perfect square} \quad (4.6) \\ 2\lfloor\sqrt{i}\rfloor + n \;\; \text{for } 1 \leq n \leq 2 & \text{if } i \text{ is not a perfect square.} \quad (4.7) \end{cases}$$

Also, consider the case where the number of the informed nodes is greater or equal to $\sqrt{N}$. Let us position the informed nodes such that they fill up the rows of the grid (which we call *filling up row-by-row*). By doing so, $|\partial M^{\tau_i}|$ will be bounded by a constant, $(\sqrt{N}+1)$, for all values of $i$. Figure 4.4 shows the obtained shape using this algorithm for

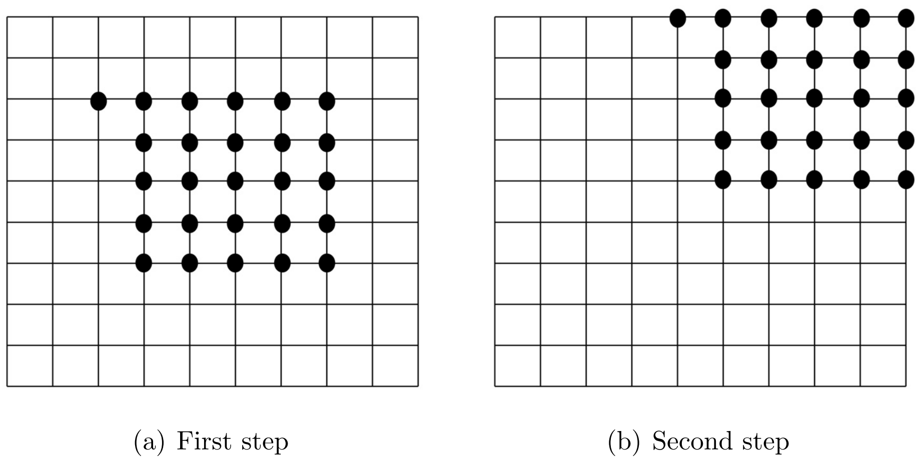



**Fig. 4.3** Positioning the informed nodes using spiral approach.

26 informed nodes, and the informed nodes which influence $|\partial M^{\tau_i}|$ are depicted in yellow.

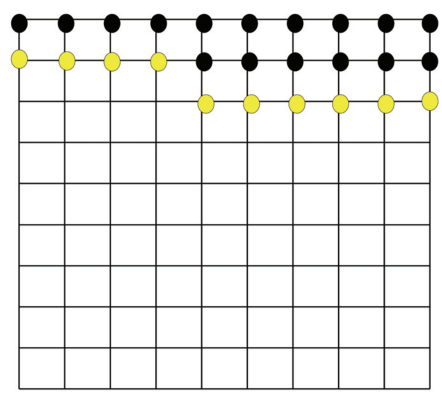


**Fig. 4.4** Filling up row by row. Yellow dots influence $|\partial M^{\tau_i}|$.

By employing the filling up row-by-row method, we have:

$$|\partial M^{\tau_i}| = \begin{cases} \sqrt{N} & \text{if the rows are completely filled up} \quad (4.8) \\ \sqrt{N}+1 & \text{otherwise.} \quad (4.9) \end{cases}$$

Therefore, from Equations 4.6, 4.7, 4.8, and 4.9, $|\partial M^{\tau_i}|$ is lower bounded as follows:

$$|\partial M^{\tau_i}| \geq \min\left(\sqrt{N}, \sqrt{N}+1, 2\sqrt{i}, 2\lfloor\sqrt{i}\rfloor + n\right) \quad \text{for } 1 \leq n \leq 2, \tag{4.10}$$

which simplifies to:

$$|\partial M^{\tau_i}| \geq \min\left(\sqrt{N}, 2\lfloor\sqrt{i}\rfloor\right). \tag{4.11}$$

Now by substituting the bound on $|\partial M^{\tau_i}|$ (Equation 4.11) in the right hand side of the inequality in Equation 4.5, we have:

$$\mathbb{E}[\tau] = \sum_{i=1}^{N-1} \mathbb{E}[\tau_{i+1} - \tau_i)] \leq N d_{max} \sum_{i=1}^{\lceil \frac{N}{2} \rceil} \frac{1}{\min\left(\sqrt{N}, 2\lfloor\sqrt{i}\rfloor\right)}. \tag{4.12}$$

In the previous equation, $2 \sum_{i=1}^{\lceil \frac{N}{2} \rceil} \frac{1}{|\partial M^{\tau_i}|}$ is substituted for $\sum_{i=1}^{N-1} \frac{1}{|\partial M^{\tau_i}|}$. Because when $i$ becomes greater than $\frac{N}{2}$, we can assume that $M^{\tau_i}$ and $V \backslash M^{\tau_i}$ change places, since it has no impact on $|\partial M^{\tau_i}|$.

### 4.2.2 Expected convergence time lower bound

Consider the case in which all of the $i$ informed nodes are of the maximum degree and they are connected to each other through only one edge. This situation leads to the maximum number of edges that $\partial M^{\tau_i}$ can obtain. Note that we assume that only one node has the maximum value at time 0.

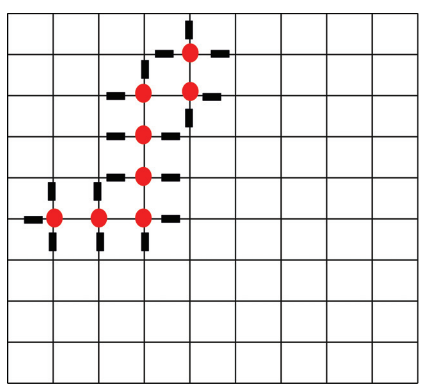


**Fig. 4.5** RPM on grid with max $|\partial M^{\tau_i}|$. Edges with black lines are in $\partial M^{\tau_i}$.

As shown in Figure 4.5, two of the informed nodes are of the degree $(d_{max} - 1)$ and the other informed nodes are of the degree $(d_{max} - 2)$. Therefore, $|\partial M^{\tau_i}|$ is upper bounded by $(i-2)(d_{max} - 2) + 2(d_{max} - 1)$. Substituting this bound on $|\partial M^{\tau_i}|$ in the left hand side of

Equation 4.5, we get:

$$\frac{Nd_{min}}{2[id_{max} - 2i + 2]} \leq \mathbb{E}[\tau_{i+1} - \tau_i]. \tag{4.13}$$

Finally, we can calculate the mean of the convergence time as:

$$\mathbb{E}[\tau] = \sum_{i=1}^{N-1} \mathbb{E}[\tau_{i+1} - \tau_i] \geq Nd_{min} \sum_{i=1}^{\lfloor \frac{N}{2} \rfloor} \frac{1}{id_{max} - 2i + 2}. \tag{4.14}$$

The above mean convergence time lower bound works for the random pairwise max algorithm on all graph types. Since we know that in grids $d_{max} = 4$ and $d_{min} = 2$, we can simplify the inequality in Equation 4.14 as:

$$\mathbb{E}[\tau] \geq N \sum_{i=1}^{\lfloor \frac{N}{2} \rfloor} \frac{1}{i+1} = N(h_{\lfloor \frac{N}{2} \rfloor} - 1), \tag{4.15}$$

where $h_{\lfloor \frac{N}{2} \rfloor}$ is the harmonic number of order $\lfloor \frac{N}{2} \rfloor$.

## 4.3 Random Pairwise Max on Gridded Toruses

To eliminate the border effect of the grids, we will study the random pairwise max algorithm on toruses. Assume that $N$ nodes are placed on the surface of a torus instead of a grid. Divide the surface of the torus into a square grid of size $\sqrt{N} \times \sqrt{N}$ such that one node is placed on every corner of the squares, as shown in Figure 4.6.

Our goal is to find an upper bound for the expected convergence time of the random pairwise max algorithm on toruses. We will use the same notation as Section 4.2.

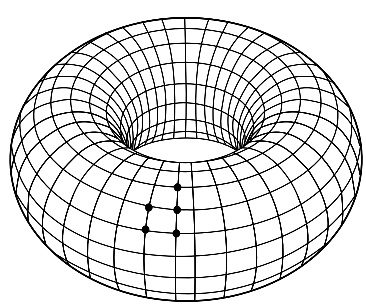


**Fig. 4.6** Sample nodes are placed on the gridded torus.

Starting from Equation 4.2 and noting that all the nodes are of degree 4 in the torus, we have:

$$\Pr[(v_i, v_j) \text{ be selected}] = \frac{1}{N}\left(\frac{1}{d_i} + \frac{1}{d_j}\right) \tag{4.16}$$

$$= \frac{1}{N}(\frac{1}{4} + \frac{1}{4}) \tag{4.17}$$

$$= \frac{1}{2N} \tag{4.18}$$

Therefore, the probability of the edge $(v_i, v_j)$ being selected in $\partial M^k$ is:

$$\Pr[(v_i, v_j) \in \partial M^k | M^k] = \frac{|\partial M^k|}{2N} \tag{4.19}$$

Following the same procedure we take to get from Equation 4.4 to 4.5, we get to:

$$\mathbb{E}[\tau_{i+1} - \tau_i] = \frac{2N}{|\partial M^{\tau_i}|} \tag{4.20}$$

Our target is to find an upper bound for the mean convergence time by finding a tight bound for $|\partial M^{\tau_i}|$. Note that a lower bound for $|\partial M^{\tau_i}|$ leads to an upper bound for the mean convergence time.

### 4.3.1 Expected convergence time upper bound

Note that the two ways of minimizing $|\partial M^{\tau_i}|$ (*filling up row-by-row* and *building square*), also apply here.

#### I. Filling up row-by-row (for $i \leq \frac{N}{2}$)

As shown in Figure 4.7, with $i$ informed nodes in a torus, $|\partial M^{\tau_i}|$ is bounded by:

$$|\partial M^{\tau_i}| = \begin{cases} 2\sqrt{N} & \text{if the rows are completely filled up} \quad (4.21) \\ 2\sqrt{N} + 1 & \text{otherwise.} \quad (4.22) \end{cases}$$

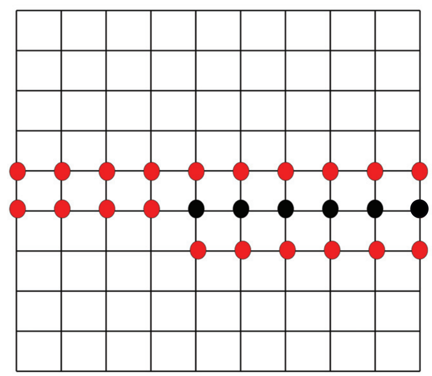


**Fig. 4.7** Filling up row by row in a torus. Red nodes influence $|\partial M^{\tau_i}|$.

### II. Building square (for $i \leq \frac{N}{2}$)

As shown in Figure 4.8, for $i$ informed nodes in a torus, $|\partial M^{\tau_i}|$ is bounded by:

$$|\partial M^{\tau_i}| = \begin{cases} 4\sqrt{i} & \text{if } i \text{ is a perfect square} \quad (4.23) \\ 4\lfloor\sqrt{i}\rfloor + n \ \text{ for } n > 0 & \text{if } i \text{ is not a perfect square.} \quad (4.24) \end{cases}$$

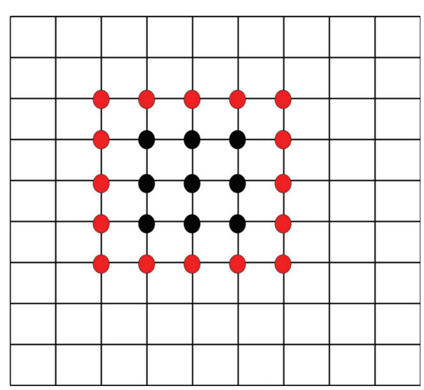


**Fig. 4.8** Building square in a torus. Red nodes influence $|\partial M^{\tau_i}|$.

Therefore, from Equations 4.21, 4.22, 4.23, and 4.24, we write,

$$|\partial M^{\tau_i}| \geq \min\left(4\lfloor\sqrt{i}\rfloor, 2\sqrt{N}\right). \tag{4.25}$$

Substitution of the bound on $|\partial M^{\tau_i}|$ (Equation 4.25) in Equation 4.20, leads to:

$$\mathbb{E}[\tau] = \sum_{i=1}^{N-1} \mathbb{E}[\tau_{i+1} - \tau_i] \leq 4N \sum_{i=1}^{\lceil \frac{N}{2} \rceil} \frac{1}{\min\left(2\sqrt{N}, 4\lfloor\sqrt{i}\rfloor\right)}. \tag{4.26}$$

The above inequality can be simplified as:

$$\mathbb{E}[\tau] \leq 2N \sum_{i=1}^{\lceil \frac{N}{2} \rceil} \frac{1}{\min\left(\sqrt{N}, 2\lfloor\sqrt{i}\rfloor\right)}. \tag{4.27}$$

The upper bound for the mean convergence time of the random pairwise max algorithm on gridded torus is half the bound we derived for the random pairwise max algorithm on grids (Equation 4.12).

## 4.4 Summary

In this chapter, we proposed upper and lower bounds for the mean convergence time of the random pairwise max algorithm on grids and toruses. Since the random broadcast max algorithm is substantially faster than the random pairwise max algorithm, we did not analyze the random pairwise max algorithm on the random geometric graphs.

The procedure of finding bounds for the mean convergence time of the random pairwise max algorithm in grids provides us the insights on how to analyze the random broadcast max algorithm in the random geometric graphs.

# Chapter 5

# Bounds on the RBM Algorithm in Reliable Networks

## 5.1 Overview

In this chapter, new upper and lower bounds for the random broadcast max algorithm on random geometric graphs, grids, and toruses are proposed. In terms of their tightness, the proposed bounds significantly outperform the bounds from [1].

## 5.2 Random Broadcast Max on Random Geometric Graphs

For analyzing the random broadcast max algorithm on random geometric graphs, we will divide the unit square into bins. As proven in [41], for constants $c > 1$ and $\mu \geq 1$, if $r^2 = c\mu\frac{\log N}{N}$, then with high probability (w.h.p), $G(V, E)$ is *$\mu$-geo-dense*, that is, any bin area of size $\frac{r^2}{\mu}$ in $G(V, E)$ contains $\Theta(\log N)$ nodes with high probability.

For our purpose, we choose $\mu$ such that any node in a bin is able to communicate with every node in its four adjacent bins. Without loss of generality, we consider the bins to be squares.

As we can see in Figure 5.1, $r$ is the longest distance between nodes in adjacent bins. Let us define $A$ as the area of the squares.

$$A = \frac{r^2}{\mu} \Longrightarrow \text{square side length } = \sqrt{A} = \frac{r}{\sqrt{\mu}}. \tag{5.1}$$

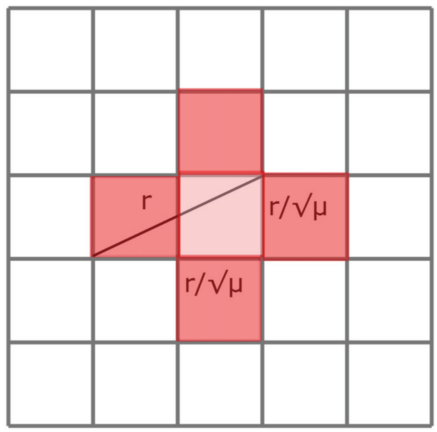



**Fig. 5.1** Dividing the unit square into bins.

Using the Pythagorean theorem, we can determine the value of $\mu$.

$$(\frac{2r}{\sqrt{\mu}})^2 + (\frac{r}{\sqrt{\mu}})^2 = r^2 \Longrightarrow \mu = 5. \tag{5.2}$$

With $\mu = 5$ and $c = 2$, we determine $r$:

$$r = \sqrt{2}\sqrt{5}\sqrt{\frac{\log N}{N}}. \tag{5.3}$$

Before continuing our analysis, we should make sure that for $r_0 = \sqrt{10}$, the random geometric graphs that we are going to study are connected. Figure 5.2 shows two important characteristics of the random geometric graphs versus the number of nodes for $r_0 = \sqrt{10}$, which are:

- percentage of connected graphs, which ensures that for $r_0 = \sqrt{10}$, the random geometric graphs are connected, and,
- mean percentage of edges, which is the number of edges of the graph over the number of edges of the complete graph with the same number of nodes, which indicates that the graphs are adequately far from the complete graphs for $r_0 = \sqrt{10}$.

The total number of bins with area $\frac{r^2}{\mu}$ is $B = \frac{N}{2\log N}$. As shown in Figure 5.1, $(\lfloor\frac{\sqrt{\mu}}{r}\rfloor)^2$ number of bins can be squares.

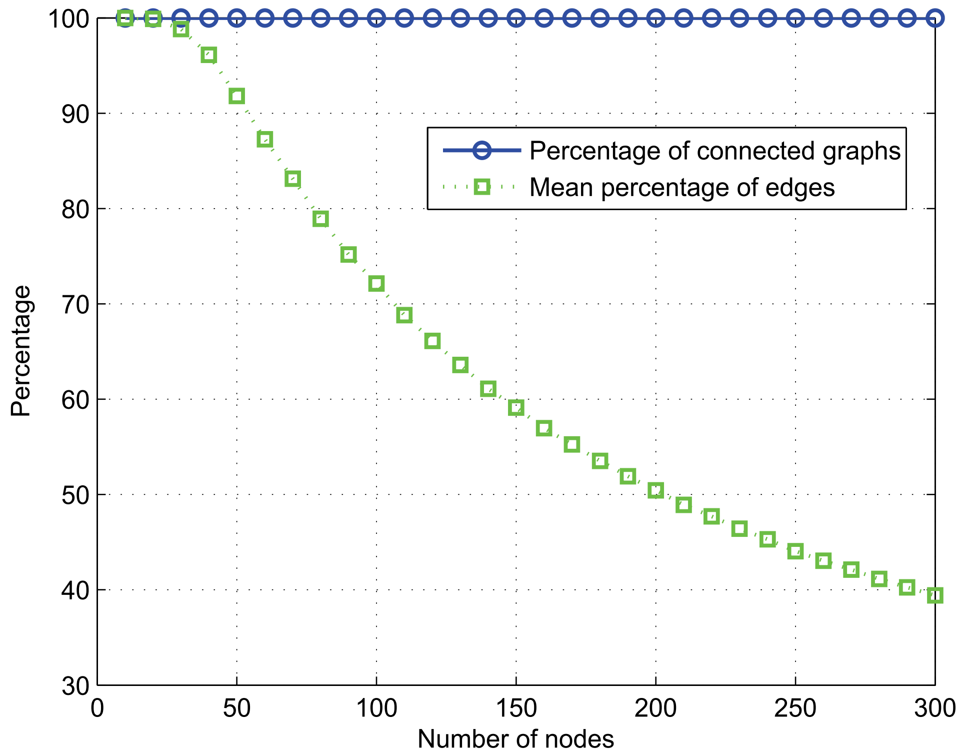



**Fig. 5.2** Connectivity of RGGs with $r_0 = \sqrt{10}$

It is shown in [42] that in the balls-and-bins problem, after throwing $m$ balls independently and uniformly at random into $n$ bins, the distribution of the number of balls in a given bin is approximately Poisson with mean $\frac{m}{n}$.

Therefore, in random geometric graphs, after throwing $N$ nodes independently and uniformly at random into $B$ bins, the distribution of the number of the nodes in a given bin ($b$) is approximately Poisson with mean $\frac{N}{B}$.

Using this Poisson approximation, we can estimate the minimum ($b_{min}$) and maximum ($b_{max}$) number of nodes in each square bin.

We will use $b_{min}$ and $b_{max}$ in our equations, when we want to estimate the number of nodes in each bin, i.e.,

$$b_{min} = F_b^{-1}(\frac{1}{N}), \tag{5.4}$$

and

$$b_{max} = F_b^{-1}(1 - \frac{1}{N}), \tag{5.5}$$

where $F_b$ is the Poisson cumulative distribution function (CDF) of $b$, with mean $\frac{N}{B}$.

Before moving on to the calculations, we should note that in this section, we use slightly different definitions for $M^k$, $X^k$, $\partial M^k$, and $|\partial M^k|$.

$M^k$: Set of bins that all their nodes are informed of the maximum at time $k$.
$X^k$: Number of bins that all their nodes are informed of the maximum at time $k$.
$\partial M^k$: Set of informed bins which have uninformed adjacent bins.
$|\partial M^k|$: Number of informed bins which have uninformed adjacent bins.

### 5.2.1 Expected convergence time upper bound

Without loss of generality, we assume that the node with the maximum value is in the corner square bin. In this case, its bin has less neighboring bins.

Since every node in a square bin is connected to all other nodes in its own bin and the four adjacent bins, whenever a node in $M^k$ wakes up, it informs all the nodes in the adjacent bins.

$$\Pr[X^{k+1} \geq X^k + 1 | M^k] = \Pr[v \in \partial M^k | M^k], \tag{5.6}$$

$$= \frac{\text{Number of nodes in a bin}}{N} |\partial M^k|, \tag{5.7}$$

$$\geq \frac{b_{min}}{N} |\partial M^k|. \tag{5.8}$$

Following the same steps to proceed from Equation 4.4 to 4.5, we obtain:

$$\mathbb{E}[\tau_{i+1} - \tau_i] \leq \frac{N}{b_{min}} \frac{1}{|\partial M_i^{\tau}|}. \tag{5.9}$$

We follow the exact same procedure to find a lower bound for $|\partial M^{\tau_i}|$ as we did for random pairwise max algorithm in grids in Section 4.2 (filling up row-by-row and building square). The bound in Equation 4.11 will be modified as:

$$|\partial M^{\tau_i}| \geq \min\left(\lfloor\frac{\sqrt{\mu}}{r}\rfloor, 2\lfloor\sqrt{i}\rfloor - 1\right), \tag{5.10}$$

where $\lfloor\frac{\sqrt{\mu}}{r}\rfloor$ and $2\lfloor\sqrt{i}\rfloor - 1$ correspond to the $|\partial M^{\tau_i}|$ in the filling up row-by-row and building square methods, respectively.

Substituting the bound on $|\partial M^{\tau_i}|$ (Equation 5.10) in Equation 5.9, gives:

$$\mathbb{E}[\tau] = \sum_{i=1}^{N-1} \mathbb{E}(\tau_{i+1} - \tau_i) \leq \frac{2N}{b_{min}} \sum_{i=1}^{\lceil\frac{B}{2}\rceil} \frac{1}{\min\left(\lfloor\frac{\sqrt{\mu}}{r}\rfloor, 2\lfloor\sqrt{i}\rfloor - 1\right)}. \tag{5.11}$$

### 5.2.2 Expected convergence time lower bound

Starting from Equation 5.7, we can write:

$$\Pr[v \in \partial M^k | M^k] \leq \frac{b_{max}}{N} |\partial M^k|. \tag{5.12}$$

Following the same procedure we take to get from Equation 4.4 to 4.5, we get to:

$$\mathbb{E}(\tau_{i+1} - \tau_i) \geq \frac{N}{b_{max}} \frac{1}{|\partial M^{\tau_i}|}. \tag{5.13}$$

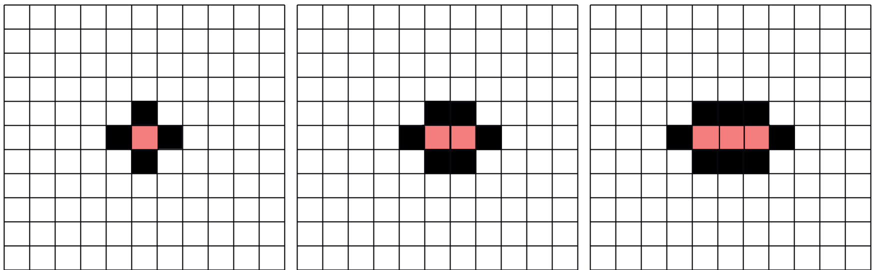
To find an upper bound for $|\partial M^{\tau_i}|$, consider the case where the node with the maximum value is in the center of the unit square. In this case, it has the maximum number of adjacent bins. As we see in Figure 5.3, in every iteration that a node wakes up in $\partial M^{\tau_i}$, it increases the size of the boundary ($|\partial M^{\tau_i}|$) by at most 2.

Substituting the bound for $|\partial M^{\tau_i}|$ in Equation 5.13, leads to:

$$\mathbb{E}[\tau] = \sum_{i=1}^{N-1} \mathbb{E}(\tau_{i+1} - \tau_i) \geq \frac{2N}{b_{max}} \sum_{i=4}^{\lfloor\frac{B}{2} - 2\rfloor} \frac{1}{i+2}. \tag{5.14}$$

## 5.3 Random Broadcast Max on Toruses

Assume that $N$ nodes are positioned on the surface of a torus. Using the same $\mu$ and $c$ as in Section 5.2, we will determine $r$ and divide the torus into bins. Recall that $b_{min}$ and $b_{max}$



(a) First step (b) Second step (c) Third step

**Fig. 5.3** $\partial M^{\tau_i}$ shown in black for the first three steps.

are the minimum and the maximum number of nodes in each square bin, respectively, and with high probability. Our goal is to find an upper bound for the expected convergence time of the random broadcast max algorithm on toruses.

### 5.3.1 Expected convergence time upper bound

Starting from Equation 5.8, we need to find a lower bound for $|\partial M^{\tau_i}|$. The only difference from Section 5.2 is that on the torus, there are no borders and corners. Therefore, the size of the boundary will be modified to:

$$|\partial M^{\tau_i}| \geq \min\left(2\lfloor \frac{\sqrt{\mu}}{r} \rfloor, 4(\lfloor \sqrt{i} \rfloor - 1)\right). \tag{5.15}$$

By substituting Equation 5.15 in 5.9, we get the mean convergence time upper bound as:

$$\mathbb{E}[\tau] = \sum_{i=1}^{N-1} \mathbb{E}(\tau_{i+1} - \tau_i) \leq \frac{2N}{b_{min}} \sum_{i=1}^{\lceil \frac{B}{2} \rceil} \frac{1}{\min\left(2\lfloor \frac{\sqrt{\mu}}{r} \rfloor, 4(\lfloor \sqrt{i} \rfloor - 1)\right)}, \tag{5.16}$$

which simplifies to:

$$\mathbb{E}[\tau] \leq \frac{N}{b_{min}} \sum_{i=1}^{\lceil \frac{B}{2} \rceil} \frac{1}{\min\left(\lfloor \frac{\sqrt{\mu}}{r} \rfloor, 2(\lfloor \sqrt{i} \rfloor - 1)\right)}. \tag{5.17}$$

## 5.4 Random Broadcast Max on Grids

In the previous two sections, we studied the random broadcast max algorithm on random geometric graphs and torus networks. In this section, we derive an upper bound on the mean convergence time of the RBM algorithm on grids. Note that the grids are of size $\sqrt{N} \times \sqrt{N}$. $M^k$ is the set of nodes that are informed of the maximum value at time $k$, $X^k = |M^k|$, and $\partial M^k$ is the set of informed nodes that have uninformed neighbors.

Starting from Equation 5.6, we get:

$$\Pr[X^{k+1} \geq X^k + 1 | M^k] = \Pr[v \in \partial M^k | M^k], \tag{5.18}$$

$$= \frac{1}{N} |\partial M^k|. \tag{5.19}$$

Following the same steps to proceed from Equation 4.4 to 4.5, we get:

$$\mathbb{E}(\tau_{i+1} - \tau_i) \leq \frac{N}{|\partial M^{\tau_i}|}. \tag{5.20}$$

The lower bound for $|\partial M^{\tau_i}|$ is very similar to what we had for the random broadcast max algorithm on random geometric graphs, i.e.,

$$|\partial M^{\tau_i}| \geq \min\left(\sqrt{N}, 2\lfloor\sqrt{i}\rfloor - 1\right), \tag{5.21}$$

where $\sqrt{N}$ and $2\lfloor\sqrt{i}\rfloor - 1$ correspond to the $|\partial M^{\tau_i}|$ in the filling up row-by-row and building square methods, respectively.

Substituting the bound for $|\partial M^{\tau_i}|$ (Equation 5.21) in Equation 5.20, we get an upper bound for the mean convergence time as:

$$\mathbb{E}[\tau] = \sum_{i=1}^{N-1} \mathbb{E}(\tau_{i+1} - \tau_i) \leq 2N \sum_{i=1}^{\lceil \frac{N}{2} \rceil} \frac{1}{\min\left(\sqrt{N}, 2\lfloor\sqrt{i}\rfloor - 1\right)}. \tag{5.22}$$

## 5.5 Summary

In this chapter, novel upper and lower bounds for the mean convergence time of the random broadcast max algorithm on random geometric graphs, grids, and toruses were proposed. Simulation results in Chapter 7 show that the bounds that we derived for the expected convergence time of the random broadcast max algorithm are significantly tighter than the bound proposed in [1].

# Chapter 6

# Bounds on the Max Consensus Algorithms in Unreliable Networks

## 6.1 Overview

In this chapter, the probability of failure is introduced to the network links and the bounds proposed in this thesis (in Chapter 4 and 5) as well as the bounds proposed in [1] for the mean convergence times of the random pairwise max and random broadcast max algorithms are modified such that they can be used in the presence of link failures.

## 6.2 RPM Algorithm in Networks With Unreliable Links

Wireless networks do not always operate perfectly due to hardware failures or the environmental conditions. In some occasions, the network can experience node or link failure which has a direct impact on the consensus convergence time. In this section, we will modify the bound in [1] and the bound we proposed in Section 4.2 for the expected convergence time of the random pairwise max algorithm, by introducing the probability of failure to the network links, denoted by $P_{fail}$. We use the same notations as in Section 4.2.

### 6.2.1 Modifying the bound in [1] for RGGs

To adjust the bound for the cases when link failures happen in the network, the probability of failure of a link should be inserted in Equation 4.1, i.e.,

$$\Pr[X^{k+1} = X^k + 1 | M^k] = \Pr[(v_i, v_j) \in \partial M^k | M^k] \Pr[\text{No link failure}], \tag{6.1}$$

where,

$$\Pr[(v_i, v_j) \in \partial M^k | M^k] \geq 2\frac{|\partial M^k|}{N d_{max}}, \tag{6.2}$$

and,

$$\Pr[\text{No link failure}] = 1 - P_{fail}. \tag{6.3}$$

Iutzeler et al. in [1], have used the Cheeger's inequality to find a bound for $|\partial M^k|$, which indicates:

$$|\partial M^k| \geq \lambda_2^L X^k (1 - \frac{X^k}{N}). \tag{6.4}$$

where $\lambda_2^L$ is the second smallest eigenvalue of the Laplacian matrix.

Following the same procedure we take to go from Equation 4.4 to 4.5, we obtain,

$$\mathbb{E}(\tau_{i+1} - \tau_i) \leq \frac{N^2 d_{max}}{2\lambda_2^L (1 - P_{fail}) i (N - i)}. \tag{6.5}$$

Using the fact that $\frac{2}{N} \sum_{i=1}^{N-1} \frac{1}{i} = \sum_{i=1}^{N-1} \frac{1}{i(N-i)}$ ,

$$\mathbb{E}[\tau] = \sum_{i=1}^{N-1} \mathbb{E}(\tau_{i+1} - \tau_i) \leq \frac{N d_{max}}{\lambda_2^L (1 - P_{fail})} \sum_{i=1}^{N-1} \frac{1}{i}. \tag{6.6}$$

Equation 6.6 is the modified bound for the mean convergence time of the random pairwise max algorithm when the network has link failures with probability of $P_{fail}$. Note that this bound is simply the bound proposed in [1] being multiplied by $\frac{1}{(1-P_{fail})}$. Therefore, we do not show extra simulation results for this bound.

### 6.2.2 Modifying the bound in Section 4.2 for grids

In Section 4.2, we proposed bounds for the mean convergence time of the random pairwise max algorithm on grids. In this subsection we modify our bound in the presence of link failures with probability of $P_{fail}$.

Equation 4.1 is changed by inserting the probability of failure, i.e.,

$$\Pr[X^{k+1} = X^k + 1 | M^k] = \Pr[(v_i, v_j) \in \partial M^k | M^k] \Pr[\text{No link failure}], \tag{6.7}$$

where,

$$\Pr[(v_i, v_j) \in \partial M^k | M^k] \geq 2 \frac{|\partial M^k|}{N d_{max}}, \tag{6.8}$$

and,

$$\Pr[\text{No link failure}] = 1 - P_{fail}. \tag{6.9}$$

Following the same procedure we take to get from Equation 4.4 to 4.5, we have:

$$\mathbb{E}(\tau_{i+1} - \tau_i) \leq \frac{N d_{max}}{2(1 - P_{fail})} \frac{1}{|\partial M^{\tau_i}|}. \tag{6.10}$$

Note that the bound for $|\partial M^{\tau_i}|$ is shown in Equation 4.11, which indicates:

$$|\partial M^{\tau_i}| \geq \min\left(2\lfloor\sqrt{i}\rfloor, \sqrt{N}\right) \quad \text{for } i \leq \frac{N}{2}. \tag{6.11}$$

By substituting Equation 6.11 in 6.10, the modified bound will be:

$$\mathbb{E}[\tau] = \sum_{i=1}^{N-1} \mathbb{E}(\tau_{i+1} - \tau_i) \leq \frac{N d_{max}}{1 - P_{fail}} \sum_{i=1}^{\lceil \frac{N}{2} \rceil} \frac{1}{\min\left(\sqrt{N}, 2\lfloor\sqrt{i}\rfloor\right)}. \tag{6.12}$$

Equation 6.12 shows the bound for the mean convergence time of the random pairwise max algorithm on grids proposed in this thesis being modified for the cases when we have probability of link failure in the network. Since this modified bound is simply the bound in the Equation 4.12 being multiplied by the factor $\frac{1}{1-P_{fail}}$, we will not show simulation results for it.

## 6.3 RBM Algorithm in Networks With Unreliable Links

In this section, we will adjust the bound proposed in [1] as well as the bound proposed in this thesis (in Section 5.2) for the expected convergence time of the random broadcast max algorithm in random geometric graphs, when the network suffers from link failures with probability of $P_{fail}$.

### 6.3.1 Modifying the bound in [1] for RGGs

We went briefly through the proof of the bound for the mean convergence time of the random broadcast max algorithm proposed in [1] in Subsection 3.5.3. Now we want to modify that bound for the case when we have communication loss or link failures.

Here, the term $\mathbb{E}(\tau_{i+1} - \tau_i)$ corresponds to the average time needed to completely fill up layer $(i+1)$ given the nodes that are informed of the maximum value at time $\tau_i$, i.e., given that layer $i$ was already filled up (shown in Figure 3.2). Therefore, the expected convergence time is upper bounded by $\sum_{i=0}^{\Delta_G - 1} \mathbb{E}(\tau_{i+1} - \tau_i)$.

To calculate the term $\mathbb{E}(\tau_{i+1} - \tau_i)$, the same proof framework as in the standard coupon collector problem [43] is suggested, i.e.,

$$\mathbb{E}(\tau_{i+1} - \tau_i) \leq N h_{|L^i|} \tag{6.13}$$

where $|L^i|$ is the number of nodes in the layer $i$, and $h_{|L^i|}$ is the harmonic number of order $|L^i|$.

The only difference when we have loss in the network is that we should take the probability of a successful broadcast into the calculations. Therefore, Equation 6.13 is modified to:

$$\mathbb{E}(\tau_{i+1} - \tau_i) \leq N \frac{h_{|L^i|}}{(1 - P_{fail})^{d_{max}}}. \tag{6.14}$$

Note that $(1 - P_{fail})^{d_{max}}$ is the probability of a successful broadcast, where $d_{max}$ is the maximum number of neighbors that each node can have. Therefore,

$$\mathbb{E}(\tau) \leq \sum_{i=0}^{\Delta_G - 1} \mathbb{E}(\tau_{i+1} - \tau_i) \leq \frac{N}{(1 - P_{fail})^{d_{max}}} \sum_{i=0}^{\Delta_G - 1} h_{|L^i|}. \tag{6.15}$$

By a couple of mathematical simplifications proposed in [1], the bound is simplified as:

$$\mathbb{E}(\tau) \leq \frac{N}{(1-P_{fail})^{d_{max}}}\Delta_G + \frac{N}{(1-P_{fail})^{d_{max}}}(\Delta_G - 1)\log\left(\frac{N-2}{\Delta_G - 1}\right). \tag{6.16}$$

Equation 6.16 is the modified bound for the mean convergence time of the random broadcast max algorithm on random geometric graphs, when the network experiences link failures. Since $(1-P_{fail})^{d_{max}}$ gets small and kills the tightness of the bound, we will not illustrate the simulation results for this bound.

### 6.3.2 Modifying the bound in Section 5.2 for RGGs

Before going into the details, we should point out that some of the definitions of the notations are slightly different from Section 5.2, i.e.,

$M^k$: Set of bins that at least half of their nodes are informed of the maximum at time $k$.
$X^k$: Number of bins that at least half of their nodes are informed of the maximum at time $k$.
$\partial M^k$: Set of bins that at least half of their nodes are informed and have uninformed adjacent bins.
$|\partial M^k|$: Number of bins that at least half of their nodes are informed and have uninformed adjacent bins.

Seeing that the network's communications fail probabilistically, we say that a bin is *almost informed* if at least half of its nodes have received the maximum value. Assume there are $b$ nodes in each bin and the nodes with indices of 1 to $\frac{b}{2}$ are uninformed while other nodes are informed.

Throughout the following calculations, we derive the expected time for an almost informed bin to become completely informed when one of its informed nodes broadcasts.

Let $T_n$ be the time it takes for the $n^{\text{th}}$ node to successfully receive the maximum value, where $1 \leq n \leq \frac{b}{2}$. Therefore, $T_n$ is geometrically distributed, where $\Pr(T_n = q) = (1 - P_{fail})(P_{fail})^{q-1}$ is the probability that the $n^{\text{th}}$ node successfully receive the information at the $q^{\text{th}}$ clock tick.

Hence, the time needed for all the nodes to be successfully informed is equal to $T = \max_{1\leq n\leq \frac{b}{2}}(T_n)$. To get the expected value of $T$, we should calculate $E[T] = E[\max_{1\leq n\leq \frac{b}{2}}(T_n)]$, where $E[T] = \sum_{q=0}^{\infty} \Pr(T > q) = \sum_{q=0}^{\infty} 1 - \Pr(T \leq q)$.

The event $T \leq q$ is the intersection of the independent events $T_n \leq q$ for $1 \leq n \leq \frac{b}{2}$, where $\Pr(T_n \leq q) = 1 - (P_{fail})^q$. Therefore,

$$E[T] = \sum_{q=0}^{\infty} 1 - \Pr(T \leq q) = \sum_{q=0}^{\infty} 1 - \Pr(T_1 \leq q)^{\frac{b}{2}}. \tag{6.17}$$

As an example, for $N = 100$, $\mu = 5$, and $c = 2$, we have $b_{max} = 18$. The expected time for $0 \leq q \leq 100$ is 0.7 for $P_{fail} = 0.1$, which indicates that after one clock tick, an almost informed bin becomes completely informed.

To find the bound for the mean convergence time of the random broadcast max algorithm, we introduce the probability of failure into Equation 5.6 to form the following equation:

$$\Pr[X^{k+1} \geq X^k + 1 | M^k] = \Pr[v \in \partial M^k | M^k] \Pr[\text{at least half the communications be successful}], \tag{6.18}$$

where,

$$\Pr[v \in \partial M^k | M^k] = \frac{\text{Number of informed nodes in a bin}}{N} |\partial M^k|, \tag{6.19}$$

$$\geq \frac{b_{min}}{2N} |\partial M^k|, \tag{6.20}$$

and,

$$\Pr[\text{at least half the communications be successful}] \geq \binom{b_{min}}{\frac{b_{min}}{2}} (1 - P_{fail})^{\frac{b_{min}}{2}} (P_{fail})^{\frac{b_{min}}{2}}. \tag{6.21}$$

Following the same steps to proceed from Equation 4.4 to 4.5, we obtain:

$$\mathbb{E}(\tau_{i+1} - \tau_i) \leq \frac{2N}{b_{min} \binom{b_{min}}{\frac{b_{min}}{2}} (1 - P_{fail})^{\frac{b_{min}}{2}} (P_{fail})^{\frac{b_{min}}{2}}} \frac{1}{|\partial M^{\tau_i}|}. \tag{6.22}$$

Substituting the bound on $|\partial M^{\tau_i}|$ (Equation 5.10) in Equation 6.22, the derived bound is:

$$\mathbb{E}[\tau] = \sum_{i=1}^{N-1} \mathbb{E}(\tau_{i+1} - \tau_i) \leq \frac{4N}{b_{min}\binom{b_{min}}{\frac{b_{min}}{2}}(1-P_{fail})^{\frac{b_{min}}{2}}(P_{fail})^{\frac{b_{min}}{2}}} \sum_{i=1}^{\lceil \frac{B}{2} \rceil} \frac{1}{\min\left(\lfloor \frac{\sqrt{\mu}}{r} \rfloor, 2\lfloor \sqrt{i} \rfloor - 1\right)}. \tag{6.23}$$

Equation 6.23 displays the modified bound for the mean convergence time of the random broadcast max algorithm in random geometric graphs when the network suffers from link failures.

## 6.4 Summary

In this chapter, we modified the bounds proposed in the previous two chapters as well as the bounds in [1] for the situations when the network faces link failure with probability of $P_{fail}$.

In next chapter, we will compare the proposed bounds with the previous state-of-the-art bounds presented in [1] to see how much improvement is gained by the novel bounds.

# Chapter 7

# Experimental Results

## 7.1 Overview

In this chapter, we provide comparisons of the performance of the proposed bounds on the mean convergence time of the maximal value spreading algorithms with the previous state-of-the-art bounds proposed in [1] (which is detailed in Chapter 2). Since the standard deviation of the proposed bounds are small, error bars are not plotted in the simulations. Experiments were performed on an Intel 3.40 GHz Core i7-2600 CPU with 16 GB of RAM and Windows 7 Professional 64-bit Operating System.

## 7.2 Random Pairwise Max Algorithm on Grids

Let us rewrite the bounds on the expected convergence time of the random pairwise max algorithm as a reminder. The upper bound which was suggested in [1] is

$$\mathbb{E}[\tau] \leq N d_{max} \frac{h_{N-1}}{\lambda_2^L}, \tag{7.1}$$

where $h_{N-1} = \sum_{k=1}^{N-1} \frac{1}{k}$ is the $(N-1)^{th}$ harmonic number, and $\lambda_2^L$ is the second smallest eigenvalue of the Laplacian matrix.

The upper and lower bounds that were derived in this thesis for the random pairwise max on grids (Equation 4.15) are

$$N\sum_{i=1}^{\lfloor \frac{N}{2}\rfloor}\frac{1}{i+1}\leq \mathbb{E}[\tau]\leq Nd_{max}\sum_{i=1}^{\lceil \frac{N}{2}\rceil}\frac{1}{\min\{\sqrt{N},2\lfloor\sqrt{i}\rfloor\}}. \tag{7.2}$$

We simulated the algorithm and compared the proposed bounds to the bound from [1]. The simulations were conducted on grids which sizes were perfect squares between 100 and 625 (i.e., 100, 121, etc.). Note that the simulations were run 100 times for each grid. The results are shown in Figure 7.1.

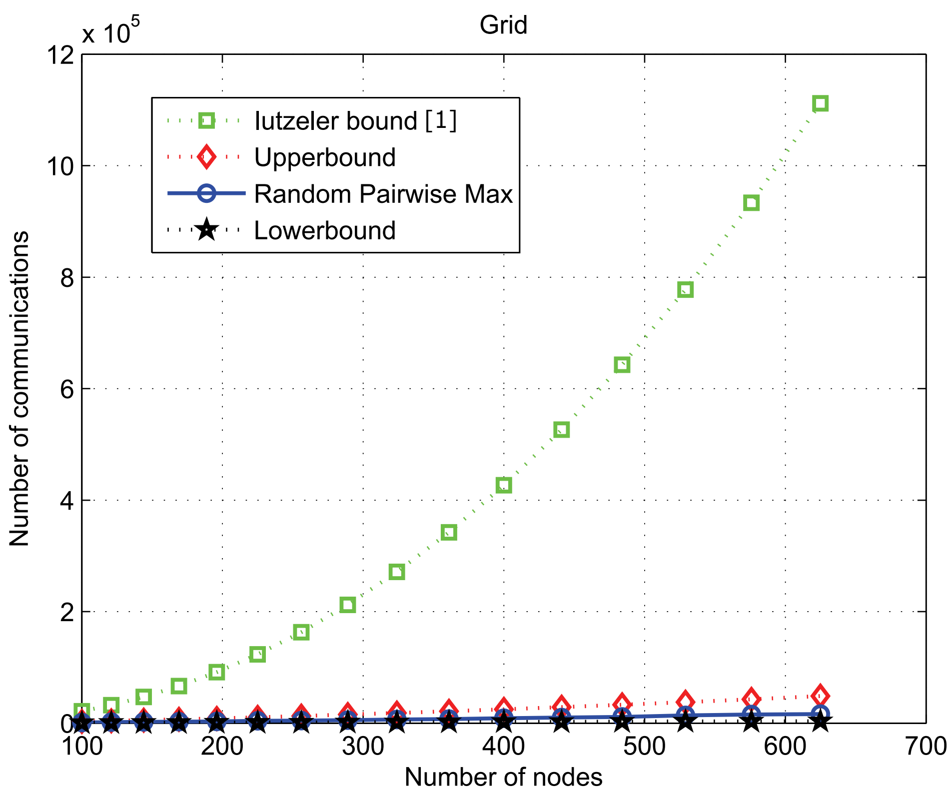



**Fig. 7.1** Empirical mean of the number of communications and corresponding bounds verses $N$ for the RPM algorithm on grids.

As we can see in Figure 7.1, the proposed bound is significantly tighter than the bound in [1]. Figure 7.2 only demonstrates the upper and lower bounds that are proven in this thesis.

Table 7.1 compares the mean convergence time and its corresponding upper bound

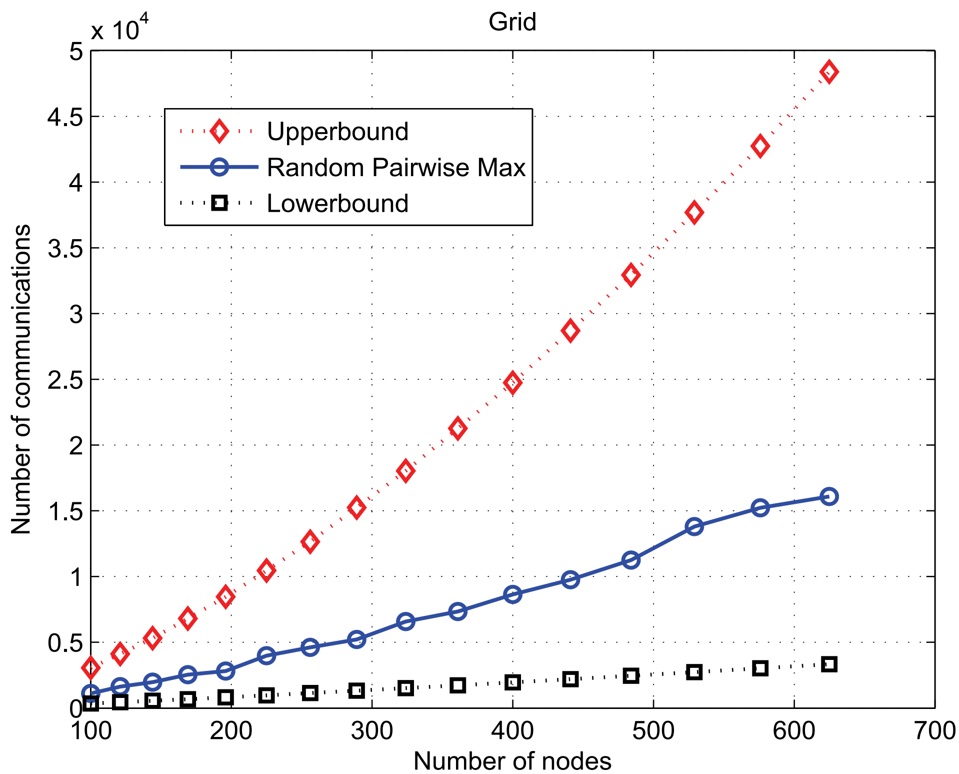



**Fig. 7.2** Empirical mean of the number of communications and proposed bounds verses $N$ for the RPM algorithm on grids.

from [1] and the upper bound which is proposed in this thesis. A significant improvement of 85% to 95% is obtained.

This notable improvement is due to the fact that $\lambda_2^L$ is very small for grids and that makes the bound from [1] loose.

## 7.3 Random Pairwise Max Algorithm on Toruses

In this section, we will show the simulation results for the random pairwise max algorithm on gridded toruses. The upper bound that we proved in Section 4.3 is

$$\mathbb{E}[\tau] \leq 2N \sum_{i=1}^{\lceil \frac{N}{2} \rceil} \frac{1}{\min\left(\sqrt{N}, 2\lfloor\sqrt{i}\rfloor\right)}. \tag{7.3}$$

| Grid size | $\mathbb{E}[\tau]$ | Upperbound from [1] | Proposed upperbound | Improvement |
|---|---|---|---|---|
| 100 | 1118 | 21157 | 3057 | 85% |
| 121 | 1629 | 32075 | 4117 | 87% |
| 144 | 1974 | 46855 | 5314 | 88% |
| 169 | 2541 | 66350 | 6808 | 89% |
| 196 | 2812 | 91508 | 8464 | 90% |
| 225 | 3984 | 123373 | 10467 | 91% |
| 256 | 4622 | 163087 | 12656 | 92% |
| 289 | 5211 | 211890 | 15239 | 92% |
| 324 | 6587 | 271122 | 18033 | 93% |
| 361 | 7342 | 342226 | 21271 | 93% |
| 400 | 8641 | 426745 | 24743 | 94% |
| 441 | 9761 | 526327 | 28707 | 94% |
| 484 | 11247 | 642724 | 32931 | 94% |
| 529 | 13801 | 777792 | 37695 | 95% |
| 576 | 15229 | 933496 | 42743 | 95% |
| 625 | 16091 | 1111906 | 48379 | 95% |

**Table 7.1** Performance comparison of upper bound in [1] and proposed upper bound for the RPM algorithm on grids.

Simulations were performed on gridded toruses whose sizes are perfect squares between 100 to 625 (i.e., 100, 121, 144, etc.). The results shown in Figure 7.3, indicate that the proposed bound is significantly tighter that the bound in [1].

## 7.4 Random Broadcast Max Algorithm on RGGs

Before going through the simulation results, it will be more convenient if we take a look at the bounds which we are going to compare.

The upper bound which was offered in [1] for the random broadcast max algorithm is

$$\mathbb{E}[\tau] \leq N\Delta_G + N(\Delta_G - 1)\log\left(\frac{N-1}{\Delta_G - 1}\right), \tag{7.4}$$

where $\Delta_G$ is the diameter of the graph.

The upper and lower bounds that are proposed in this thesis on the expected convergence

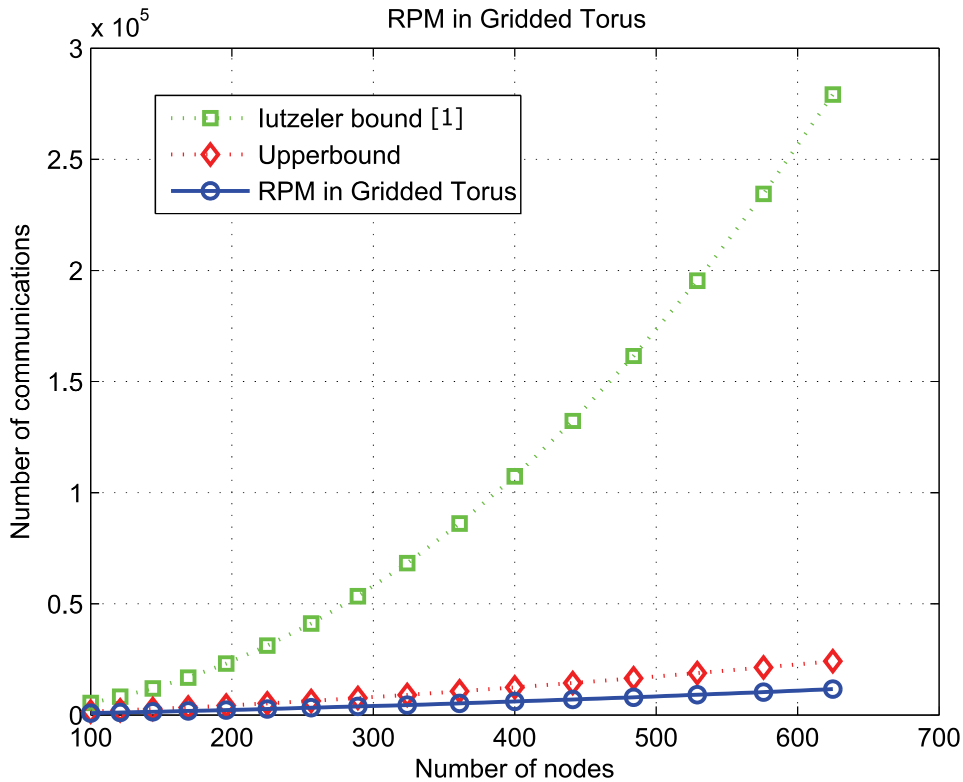



**Fig. 7.3** Empirical mean of the number of communications and corresponding bound verses $N$ for RPM algorithm on toruses.

time of the random broadcast max algorithm in random geometric graphs are

$$\frac{2N}{b_{max}} \sum_{i=2}^{\lfloor \frac{B}{2} \rfloor - 2} \frac{1}{i+2} \leq \mathbb{E}[\tau] \leq \frac{2N}{b_{min}} \sum_{i=1}^{\lceil \frac{B}{2} \rceil} \frac{1}{\min\left(\lfloor \frac{\sqrt{\mu}}{r} \rfloor, 2\lfloor \sqrt{i} \rfloor - 1\right)}. \tag{7.5}$$

The results of the simulations indicate that the proposed upper bound is much tighter than the bound in [1]. The simulations were conducted on 100 different random geometric graphs for every $N$(number of nodes), and 100 times for every graph, as shown in Figure 7.4.

The upper bound from Iutzeler et al. [1] and the proposed upper bound are compared in Table 7.2. Notable improvement of 58% to 82% is observed.

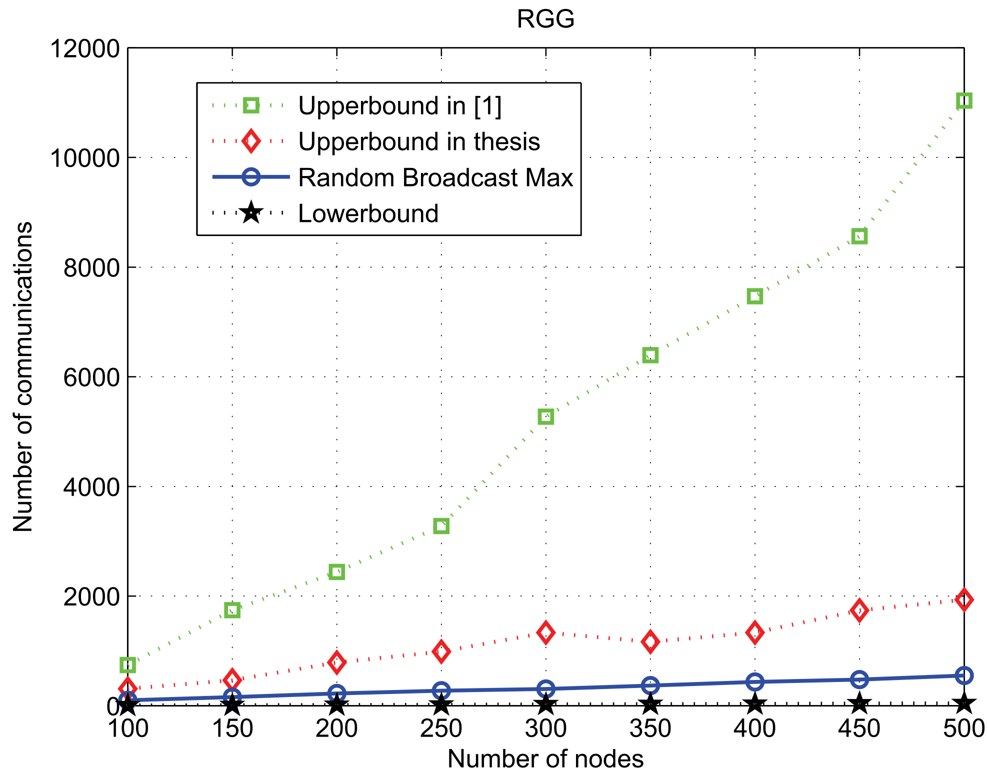



**Fig. 7.4** Empirical mean of the number of communications and corresponding bounds verses $N$ for the RBM algorithm on RGGs.

| Graph size | $\mathbb{E}[\tau]$ | Upperbound from [1] | Proposed upperbound | Improvement |
|---|---|---|---|---|
| 100 | 97 | 744 | 311 | 58% |
| 150 | 158 | 1743 | 466 | 73% |
| 200 | 222 | 2440 | 788 | 67% |
| 250 | 274 | 3277 | 986 | 69% |
| 300 | 304 | 5270 | 1333 | 75% |
| 350 | 369 | 6394 | 1166 | 81% |
| 400 | 433 | 7468 | 1333 | 82% |
| 450 | 478 | 8561 | 1740 | 79% |
| 500 | 550 | 11036 | 1933 | 82% |

**Table 7.2** Performance comparison of upper bound in [1] and the proposed upper bound for the RBM algorithm on RGGs.

## 7.5 Random Broadcast Max Algorithm on Toruses

The bound that was derived in Section 5.3 is

$$\mathbb{E}[\tau] \le \frac{N}{b_{min}} \sum_{i=1}^{\lceil \frac{B}{2} \rceil} \frac{1}{\min\left( \lfloor \frac{\sqrt{\mu}}{r} \rfloor, 2(\lfloor \sqrt{i} \rfloor - 1) \right)}. \tag{7.6}$$

Figure 7.5 displays the simulation results performed for the random broadcast max algorithm on toruses. Simulations were run on random geometric graphs of sizes 100 to 280 with steps of 10. Results of the simulations show great improvement in the tightness of the proposed bound compared to the bound from [1].

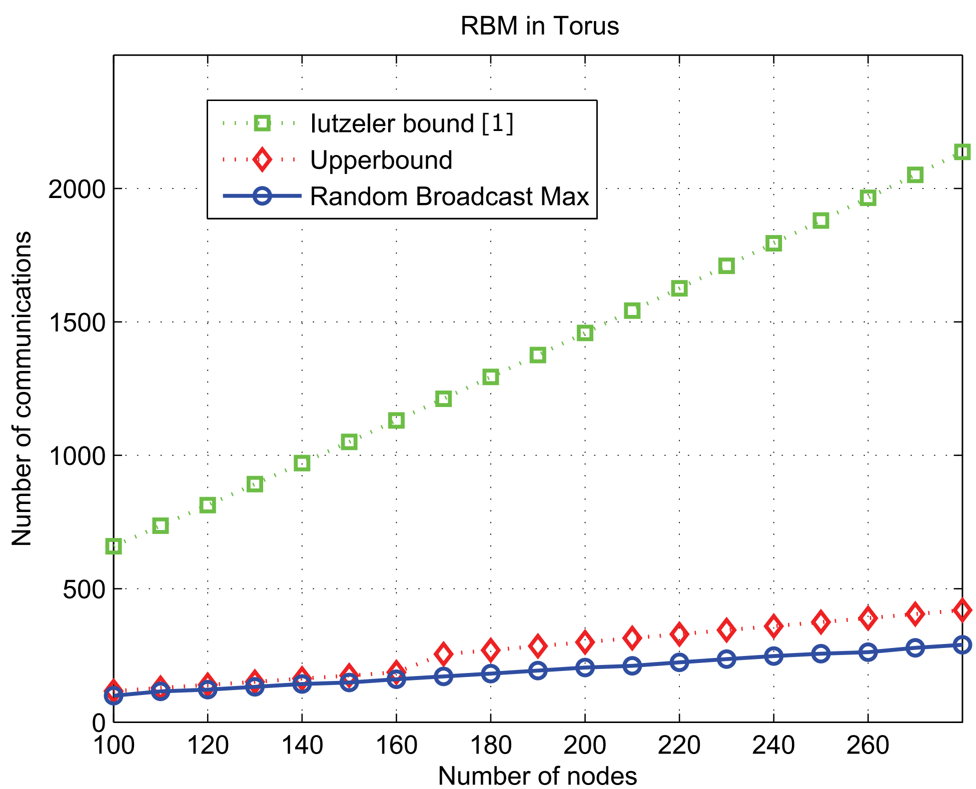



**Fig. 7.5** Empirical mean of the number of communications and corresponding bound verses $N$ for RBM in torus.

## 7.6 RBM Algorithm on RGG With Link Failure

In this section we compare the bound on the expected convergence time of the random broadcast max algorithm in the presence of link failures to the bound in loss-less networks in random geometric graphs. The modified bound from Subsection 6.3.2 is as follows.

$$\mathbb{E}[\tau] \leq \frac{4N}{b_{min}\binom{b_{min}}{\frac{b_{min}}{2}}(1-P_{fail})^{\frac{b_{min}}{2}}(P_{fail})^{\frac{b_{min}}{2}}} \sum_{i=1}^{\lceil \frac{B}{2} \rceil} \frac{1}{\min\left(\lfloor \frac{\sqrt{\mu}}{r} \rfloor, 2\lfloor \sqrt{i} \rfloor - 1\right)}. \tag{7.7}$$

Since in the random broadcast max algorithm, each node broadcasts its value to all its neighbors, it is expected that a small probability of failure does not affect the convergence speed much. Hence, for the simulation results to be more readable, we set the probability of failure to 70%. Note that in real networks, the probability of failure is much less than 70%. Figure 7.6 represents the simulation outcomes.

As illustrated in Figure 7.6, the mean of the number of communications for the random broadcast max algorithm even in the presence of 70% probability of link failure is almost the same as the case when the network is error free.

## 7.7 Summary

In this chapter, the performances of the proposed bounds were compared to the previous state-of-the-art bounds in [1].

A remarkable improvement of 58% to 82% was observed for the expected convergence time bound of the random broadcast max algorithm on random geometric graphs. Also, 85% to 95% enhancement was obtained for the bound on the mean convergence time of the random pairwise max algorithm on grids.

The simulation results for both the random pairwise max and random broadcast max algorithms on toruses indicated that the proposed bounds are remarkably tighter compared to the bounds in [1].

In last section of this chapter, simulation outcome for the random broadcast max algorithm on random geometric graphs under the assumption of the presence of link failures was displayed.

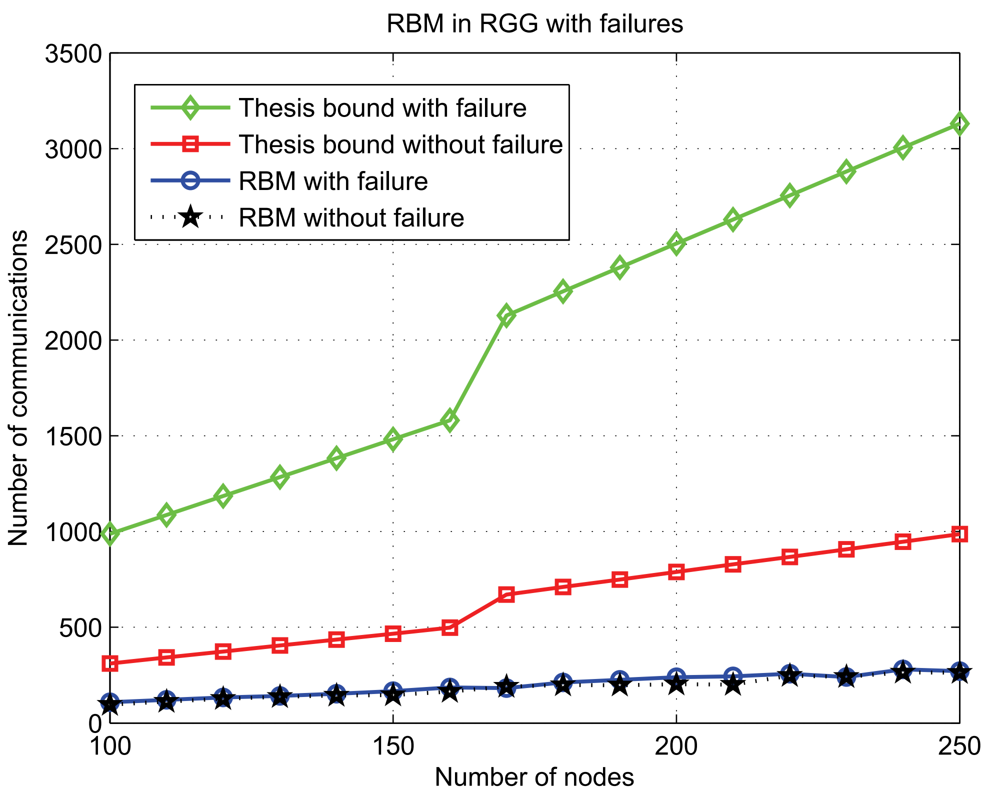



**Fig. 7.6** Empirical mean of the number of communications and corresponding bound verses $N$ for RBM in RGG with link failure.

# Chapter 8

# Conclusion and Future Work

## 8.1 Concluding Remarks

In this thesis we reviewed several research articles analyzing the maximum value consensus algorithms and their corresponding bounds on the mean convergence time. The bounds that were proposed in [1] did not demonstrate significant tightness for the expected convergence time of the random pairwise max and random broadcast max algorithms.

Therefore, we aimed to find tighter bounds on the mean convergence time of these algorithms by tackling the problem from a different point of view.

First, we derived bounds for the random pairwise max algorithm on grids, which showed 85% to 95% improvement in comparison to the bounds in [1]. This substantial improvement is due to the fact that the bound on the expected convergence time of the random pairwise max algorithm in [1] is inversely proportional to the second smallest eigenvalue of the Laplacian matrix ($\lambda_2^L$), is relatively small for grids in comparison to random geometric graphs.

Then, we derived bounds for the random broadcast max algorithm in random geometric graphs by dividing the unit square into bins. The random pairwise max algorithm in grids provided us the insights on how to analyze the random broadcast max algorithm in random geometric graphs. Simulation results revealed an improvement of 58% to 82% in comparison to the bounds in [1].

The random pairwise max and random broadcast max algorithms were also analyzed on the networks whose nodes were located on the surfaces of toruses. Torus networks are important to study because they eliminate the border effect of the grids and random geometric graphs (unit square). Simulations indicated that the proposed bounds for the random pairwise max and random broadcast max algorithms in toruses are significantly tighter than the bounds in [1].

Finally, we enhanced the bounds to account for link failure. We know that due to the environmental conditions or hardware failures, the networks may occasionally encounter communication failures. Therefore, the bounds that we proved in Chapters 4 and 5 were modified such that they could be used in the presence of link failures.

## 8.2 Future Work

Since the maximum value consensus is not widely studied yet, there is a number of research topics in this area that have not been fully explored so far. As an example, our suggestion is to further develop the *geographic gossip* algorithm proposed in [25, 44] by combining it with the random broadcast max algorithm. The geographic gossip algorithm for averaging consensus problem computes the average faster than the random pairwise algorithm. We suspect that the combined algorithm will result in a lower convergence time compared to the random broadcast max algorithm by taking advantage of the geographic gossip algorithm.

# References

[1] Franck Iutzeler, Philippe Ciblat, and Jérémie Jakubowicz. Analysis of max-consensus algorithms in wireless channels. *IEEE Transactions on Signal Processing*, 60(11):6103–6107, 2012.

[2] Franck Iutzeler. *Distributed Estimation and Optimization for Asynchronous Networks*. PhD thesis, Telecom ParisTech, 2013.

[3] Tuncer C. Aysal, Mehmet E. Yildiz, Anand D. Sarwate, and Anna Scaglione. Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal Processing*, 57(7):2748–2761, 2009.

[4] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.

[5] B. Monajemi Nejad, Sid Ahmed Attia, and Jörgz Raisch. Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies. In *Proceedings of International Symposium on Information, Communication and Automation Technologies, ICAT*, pages 1–7, 2009.

[6] Amit Bhatia and Emilio Frazzoli. Decentralized algorithm for minimum-time rendezvous of dubins vehicles. In *Proceedings of IEEE American Control Conference*, pages 1343–1349, 2008.

[7] Giuseppe Notarstefano and Francesco Bullo. Distributed consensus on enclosing shapes and minimum time rendezvous. In *Proceedings of 45th IEEE Conference on Decision and Control*, pages 4295–4300, 2006.

[8] Daniel Spielman. Spectral graph theory. *Lecture Notes, Yale University*, pages 740–0776, 2009.

[9] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

[10] Fan R.K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[11] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing*, pages 1–12, 1987.

[12] Peter J. Slater, Ernest J. Cockayne, and Stephen T. Hedetniemi. Information dissemination in trees. *SIAM Journal on Computing*, 10(4):692–701, 1981.

[13] Boris Pittel. On spreading a rumor. *SIAM Journal on Applied Mathematics*, 47(1):213–223, 1987.

[14] Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. Randomized broadcast in networks. *Random Structures & Algorithms*, 1(4):447–460, 1990.

[15] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 174–185, 1999.

[16] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. In *Proceedings of 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 565–574, 2000.

[17] Srinivas Kashyap, Supratim Deb, KVM Naidu, Rajeev Rastogi, and Anand Srinivasan. Efficient gossip-based aggregate computation. In *Proceedings of the 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 308–317, 2006.

[18] Benjamin Doerr, Tobias Friedrich, and Thomas Sauerwald. Quasirandom rumor spreading: Expanders, push vs. pull, and robustness. In *Proceeding of International Colloquium on Automata, Languages and Programming*, pages 366–377. 2009.

[19] Nikolaos Fountoulakis, Anna Huber, and Konstantinos Panagiotou. Reliable broadcasting in random networks and the effect of density. In *Proceedings of IEEE INFOCOM*, pages 1–9, 2010.

[20] Imrich Chlamtac and Shay Kutten. On broadcasting in radio networks–problem analysis and protocol design. *IEEE Transactions on Communications*, 33(12):1240–1246, 1985.

[21] Artur Czumaj and Wojciech Rytter. Broadcasting algorithms in radio networks with unknown topology. In *Proceedings of 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 492–501, 2003.

[22] Eyal Kushilevitz and Yishay Mansour. An $\omega$(d\log(n/d)) lower bound for broadcast in radio networks. *SIAM journal on Computing*, 27(3):702–712, 1998.

[23] Reuven Bar-Yehuda, Oded Goldreich, and Alon Itai. On the time-complexity of broadcast in radio networks: an exponential gap between determinism randomization. In *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing*, pages 98–108, 1987.

[24] Franco Travostino, Joe Mambretti, and Gigi Karmous-Edwards. *Grid Networks: Enabling Grids with Advanced Communication Technology*. John Wiley & Sons, 2006.

[25] Alexandros D.G. Dimakis, Anand D. Sarwate, and Martin J. Wainwright. Geographic gossip: Efficient averaging for sensor networks. *IEEE Transactions on Signal Processing*, 56(3):1205–1216, 2008.

[26] Mathew Penrose. *Random Geometric Graphs*. Oxford University Press Oxford, 2003.

[27] Arjun Singh, William J Dally, Amit K Gupta, and Brian Towles. Goal: a load-balanced adaptive routing algorithm for torus networks. *ACM SIGARCH Computer Architecture News*, 31(2):194–205, 2003.

[28] Morris H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[29] Leslie Lamport. The part-time parliament. *ACM Transactions on Computer Systems (TOCS)*, 16(2):133–169, 1998.

[30] Michael J. Fischer. The consensus problem in unreliable distributed systems (a brief survey). In *Foundations of Computation Theory*, pages 127–140, 1983.

[31] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Analysis and optimization of randomized gossip algorithms. In *43rd IEEE Conference on Decision and Control, CDC*, volume 5, pages 5310–5315, 2004.

[32] Fabio Fagnani and Sandro Zampieri. Asymmetric randomized gossip algorithms for consensus. In *Proceedings of IFAC World Congress*, pages 9051–9056, 2008.

[33] Luca Schenato and Giovanni Gamba. A distributed consensus protocol for clock synchronization in wireless sensor network. In *Proceedings of 46th IEEE Conference on Decision and Control*, pages 2289–2294, 2007.

[34] Tuncer C. Aysal, Mark J. Coates, and Michael G. Rabbat. Distributed average consensus with dithered quantization. *IEEE Transactions on Signal Processing*, 56(10):4905–4918, 2008.

[35] Tuncer C. Aysal, Mark Coates, and Michael Rabbat. Distributed average consensus using probabilistic quantization. In *Proceedings of 14th IEEE/SP Workshop on Statistical Signal Processing, SSP'07*, pages 640–644, 2007.

[36] Jorge Cortés. Distributed algorithms for reaching consensus on general functions. *Automatica*, 44(3):726–737, 2008.

[37] Uriel Feige. A tight upper bound on the cover time for random walks on graphs. *Random Structures & Algorithms*, 6(1):51–54, 1995.

[38] Colin Cooper and Alan Frieze. The cover time of random geometric graphs. *Random Structures & Algorithms*, 38(3):324–349, 2011.

[39] Frank Harary and Heiko Harborth. Extremal animals. *Journal of Combinatorics, Information and System Sciences*, 1(1):1–8, 1976.

[40] Sascha Kurz. Counting polyominoes with minimum perimeter. *Ars Combinatoria*, 88:161–174, 2008.

[41] Chen Avin and Gunes Ercal. On the cover time and mixing time of random geometric graphs. *Theoretical Computer Science*, 380(1):2–22, 2007.

[42] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[43] Rajeev Motwani and Prabhakar Raghavan. Randomized algorithms. *ACM Computing Surveys (CSUR)*, 28(1):33–37, 1996.

[44] Alexandros G. Dimakis, Anand D. Sarwate, and Martin J. Wainwright. Geographic gossip: efficient aggregation for sensor networks. In *Proceedings of 5th ACM International Conference on Information Processing in Sensor Networks*, pages 69–76, 2006.