STUDIES OF INTERCONNECTION NETWORKS WITH APPLICATIONS IN BROADCASTING

CALIN DAN MOROSAN

A THESIS
IN
THE DEPARTMENT
OF
COMPUTER SCIENCE AND SOFTWARE ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

FEBRUARY 2007

Your file Votre référence
ISBN: 978-0-494-30133-3
Our file Notre référence
ISBN: 978-0-494-30133-3

Abstract

Studies of interconnection networks with applications in broadcasting

CALIN DAN MOROSAN, Ph.D.

Concordia University, 2007

The exponential growth of interconnection networks transformed the communication primitives into an important area of research. One of these primitives is the one-to-all communication, i.e. broadcasting. Its presence in areas such as static and mobile networks, Internet messaging, supercomputing, multimedia, epidemic algorithms, replicated databases, rumors and virus spreading, to mention only a few, shows the relevance of this primitive.

In this thesis we focus on the study of interconnection networks from the perspective of two main problems in broadcasting: the minimum broadcast time problem and the minimum broadcast graph problem. Both problems are discussed under the 1-port constant model, which assumes that each node of the network can communicate with only one other node at a time and the transmitting time is constant, regardless of the size of the message.

In the first part we introduce the minimum broadcast time function and we present two new properties of this function. One of the properties yields an iterative heuristic for the minimum broadcast time problem, which is the first iterative approach in approximating the broadcast time of an arbitrary graph.

In the second part we give exact upper and lower bounds for the number of broadcast schemes in graphs. We also propose an algorithm for enumerating all the broadcast schemes and a random algorithm for broadcasting.

In the third part we present a study of the spectra of Knödel graph and their applications. This study is motivated by the fact that, among the three known infinite families of minimum broadcast graphs, namely the hypercube, the recursive circulant, and the Knödel graph, the last one has the smallest diameter.

In the last part we introduce a new measure for the fault tolerance of an interconnection network, which we name the global fault tolerance. Based on this measure,

we make a comparative study for the above mentioned classes of minimum broadcast graphs, along with other classes of graphs with good communication properties.

Acknowledgments

I would like to express my gratitude to a few people who contributed, directly or indirectly, to this thesis.

First of all, I am highly indebted to my supervisor, Dr. Hovhannes Harutyunyan, for his continuous support and sagacious advice.

I would like to thank the committee members and anonymous reviewers whose feedback improved many of the chapters of this thesis.

I want to dedicate this thesis to my wife and my parents for constantly supporting and encouraging me to persevere in my studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Models of broadcasting

The exponential growth of interconnection networks transformed the communication primitives into an important area of research. These primitives can be defined as follows:

- **Routing** or *one-to-one* communication.
- **Broadcasting** or *one-to-all* communication.
- **Multicasting** or *one-to-many* communication.
- **Gossiping** or *all-to-all* communication.

Broadcasting is the problem of dissemination of information in which one piece of information needs to be transmitted to a group of individuals connected by an interconnection network. This problem finds applications in areas such as fixed and mobile networks, Internet messaging, supercomputing, multimedia, epidemic algorithms, replicated databases, rumors and virus spreading, which shows the importance of this communication primitive.

Among the above mentioned application areas of the broadcasting process, supercomputing seems to be the one which relies heavily on optimal broadcasting. This dependence is due to data dependency arising in real world applications. The well-known Amdahl's law gives the speedup limitation S function of the number of processors p and the fraction β of serial part of the parallel algorithm:

$$S = \frac{p}{\beta p + (1 - \beta)} \tag{1}$$

Usually, it is assumed that β depends only on the algorithm used (constant for a given problem) and the speedup tends to $1/\beta$ as p grows. Unfortunately, this assumption is no longer true if the problem solved involves data dependency, and the algorithm designed to solve it makes appeal to the broadcast primitive. If we have data dependency, then β is a monotonically increasing function of p: $\beta = \beta_0 + f(p)$. This is due to the fact that the time needed to broadcast is lower bounded by a logarithmic factor in terms of the number of processors. Therefore, the speedup theoretically tends to 1 as the number of processors tends to ∞. Also, this excludes a priori the grid networks as underlining architectures for a supercomputer, since they require over logarithmic time for the broadcasting operations, comparing to the logarithmic time encountered in more complex architectures, as hypercubes for example. PVM (Parallel Virtual Machine) [128], MPI (Message Passing Interface) [221], and DECK (Distributed Execution and Communication Kernel) [23, 24] are the major examples of programming environments that provide functions for collective communication.

The theory of broadcasting has been focused on two main problems. The first one, called the *minimum broadcast time* problem, can be defined as follows: given a network, find a strategy, called *broadcast scheme*, such that the time needed to transmit the information is minimized. The second one, called the *minimum broadcast graph* problem, can be defined as follows: find a network architecture on n vertices with broadcast time $\lceil \log_2 n \rceil$ and a minimum number of edges. Clearly, the answer to the above-mentioned problems is highly dependent on the model of communication used.

From the point of view of the underlying topology used to represent the interconnection network, a connected graph $G = (V, E)$ is usually employed, in which the members of the network are the vertices of G, and the communication links are the edges of G.

At the communication link level there are two models discussed in the literature [170]:

- **One-way mode** – also called *telegraph mode* or *half-duplex*. In this mode, one link may be used only in one direction during a communication process involving two adjacent nodes. This process can be modelled using directed graphs as the underlying topology.

- **Two-way mode** – also called *telephone mode* or *full-duplex*. In this mode, one link may be used in both directions during a communication process involving two adjacent nodes. This process can be modelled using undirected graphs as the underlying topology.

Following [117], communication in interconnection networks can be classified based on the ability of the vertices to communicate simultaneously with their neighbours in:

- **Processor-bound** also called *1-port* or *whispering*, in which a vertex can communicate only with one neighbour at a time.

- **Link-bound** also called *n-port* or *shouting*, in which a vertex can call all its neighbours simultaneously.

Here we have to mention *k-broadcasting* which is a model of broadcasting that fits between the two above-mentioned models. Some authors also refer to it as *c-broadcasting* [133, 201, 202, 242]. In this model, a vertex can call simultaneously up to k of its neighbours. A considerable amount of literature [194, 145, 146, 155, 156, 241] is dedicated to this model, which is useful to the study of DMA-bound systems [200] or in computing functions in networks [16, 46, 78].

Another issue in characterizing communication in networks is the necessary time for a message to be prepared, to travel along an edge, and to be received. The are two widely used models addressing this issue:

- The **constant model**, in which the time needed to transmit and receive a message is constant, $T = const.$

- The **linear model**, in which the time needed to communicate is modelled as $T = \beta + L\tau$, where β is the cost of preparing the message, L the length of the message, and τ the propagation time of a data unit length.

Most papers on broadcasting adopt the constant model, since the linear model can be reduced to the constant model by quantifying the larger messages to the shortest possible message. Nevertheless, there are some specific results regarding the linear model for broadcasting (see, for example, [235, 111, 76, 27]).

Using different assumptions regarding the communication model employed, other broadcasting models have been developed and analyzed:

- **Vertex disjoint path mode broadcasting**

  In this model, in every round of communication, the information is transmitted from the informed nodes to the uninformed ones via disjoint sets of vertices, which can form paths of length greater than one. There are two flavors of this model [96, 101, 192, 171, 41, 113, 114, 116] considering either that one end-node broadcasts its whole knowledge to all other nodes along the path or that one of the end-nodes sends its knowledge to the other end-node and the nodes in between do not read the message sent. The first one is the most studied in the literature and it has the property that its complexity differs from the complexity of the accumulation problem. The accumulation problem is the opposite of broadcasting and can be succinctly defined as the all-to-one communication problem.

- **Edge disjoint path mode broadcasting**

  In this model, in every round of communication, the information is transmitted from the informed nodes to the uninformed ones via disjoint sets of edges, which can form paths of length greater than one. This model was also investigated in several papers [96, 101, 172, 170, 115, 113], and is quite similar to the *wormhole routing* model employed for the analysis of permutation routing [74, 184, 4]. Note that both vertex disjoint and edge disjoint path models are called *line model.*

- **Restricted protocol broadcasting**

  In this model, the broadcast protocol may have two types of restrictions: *input restriction* and *output restriction.* A protocol has input restriction i at a given node if the messages transmitted during any outgoing activation have been communicated to this node during at most i of the previous incoming activations. Similarly, the protocol has output restriction o at a given node if the messages received during any incoming activation will be delivered by that node using at most o successive out-going activations. For instance, the broadcast protocols running on d-bounded degree networks are $(1, d)$-restricted, the gossip protocols on d-bounded degree networks are (∞, d)-restricted, and the s-systolic gossip protocols are (i, o)-restricted with $i + o = s$. This model has been examined in a general sense in [110]. Other results concerning specific networks can be found in [235, 173, 117, 170].

- **(i, j) mode broadcasting**

  In this model, in any round, a node can send a message to i neighbours via i incident edges and it can receive messages from j neighbours via j incident edges. Therefore, the two-way mode can be seen as a $(1, 1)$ mode, with the additional constraint holding that the edge for receiving and the edge for transmitting are the same. This model has been investigated in [94].

- **Radio broadcasting**

  In this model communication in the network is assumed to be synchronous, i.e., it occurs in discrete pulses, called *rounds*. Following [89], on each round of communication, each node that knows the message to be transmitted is allowed to send it to all its neighbours. Note that a node can either send the message to all its neighbours or not send it at all, but it can transmit to no strict subset of its set of neighbours. Furthermore, only a processor that receives the message from precisely one neighbor in a certain round is considered to be informed in that round. The intuition behind this constraint is that a message received from more than one neighbor in the same round gets corrupted. There is also a considerable amount of literature concerned with this model (see for example [121, 176, 57, 58, 56, 8, 9, 25, 59, 125, 193, 89, 90, 91, 92]).

- **Neighborhood broadcasting**

  The neighborhood broadcast problem (NBP) was introduced in [70] and is the problem of disseminating a message from a source vertex to all the vertices adjacent to the source vertex under the following constraints:

  i) each call involves only two vertices;

  ii) each call takes one time unit;

iii) each vertex can participate in only one call per time unit;

iv) a vertex can call only a vertex to which it is adjacent.

This model of neighborhood information dissemination has been investigated for some particular network topologies as n-star graphs in [123, 227, 214], hypercubes in [124, 216, 32], Cayley graphs in [215], and in general in [70, 106, 107].

- **Broadcasting with universal lists**

  In the previously presented models, the order in which every informed node will inform its uninformed neighbours depends on the originator of the broadcast. In *broadcasting with universal lists*, also known as *orderly broadcasting*, a (universal) broadcasting scheme is a function assigning a single ordered list of its neighbors, called the universal list, to every node. The list is determined regardless of the source and a node will transmit the received message in the order of the list.

  The problem of broadcasting with universal lists was introduced in [79], taking into account two models: *adaptive* and *nonadaptive*. In the adaptive model, each node knows which neighbors the obtained messages came from and can skip those neighbors in its list. However, in the nonadaptive model, a node does not know the neighbors from which it receives the messages, and may retransmit the messages to those neighbors. Thus, each node obliviously sends the source message to neighbors in order of its list since, in some applications such as radio communication, nodes may not know the origin of the messages [60, 61].

  This model is suitable for nodes with insufficient memory to keep a coordinated protocol or for ad-hoc networks. This problem has been investigated under the name of *broadcasting with universal lists* in [79, 148, 189] and under the name of *orderly broadcasting* in [157] for a 2D torus.

- **Messy broadcasting**

  The *messy broadcasting* model has been introduced in [3]. Unlike the previously presented models, messy broadcasting is concerned with analyzing the worst case performance of the broadcast schemes. In other words, the messy broadcasting model is looking for upper bounds in the broadcast time, following the constraints below:

- one node knows only its neighbors;
- the originator or the time slot is not known;
- there is no coordinating leader;
- 1-port, constant model is assumed.

This model is suitable for nodes with insufficient memory to keep a coordinated protocol or for ad-hoc networks.

In the previously presented models, the nodes know the topology and, except in the orderly broadcasting, they know the originator of the message and the time slot. In practice, however, it is not always realistic to assume that each node of a network will know the network topology, or will know how to make decisions based on a set of stored protocols. In many cases, the nodes of the network have primitive structures with small memories that cannot store such information or make intelligent decisions. On the other hand, building networks in which the nodes have no decision-making responsibility is much simpler and more robust. For these main reasons, the study of messy broadcasting has become especially interesting. One of the major differences between the messy broadcasting and the previously presented broadcasting models is that, in a messy broadcast scheme, the vertices know nothing about the network topology, and at each broadcast round transmit the message to a randomly selected neighbor.

Three models of messy broadcasting have been proposed in literature, depending on the amount of knowledge about neighbors:

- **Model M1**

  At each unit of time, every vertex knows the state of each of its neighbours: informed or uninformed. In this model, each informed vertex must transmit the broadcast message to one of its uninformed neighbours, if any, in each time unit.

- **Model M2**

  Every informed vertex knows from which vertex (vertices) it received the broadcast message and to which neighbours it has sent the message. Thus, it knows that this vertex (or these vertices) are informed. In this model, each informed vertex must transmit the broadcast message to one of its neighbours other than the ones that it knows that are informed, if any, in each time unit.

- **Model M3**

  Every informed vertex knows to which neighbours it has sent the message. In this model, each informed vertex must transmit the broadcast message to one of its neighbours to which it has not yet sent the message, if any, in each time unit.

The worst case scenarios for various network topologies such as paths, cycles, hypercubes, and d-ary trees were studied in [143]. The messy broadcast times for complete bipartite graphs and an improved lower bound for messy broadcast times on hypercubes of arbitrary dimension has been derived in [159, 160]. In [66], the exact values and bounds for the broadcast times on multi-dimensional directed tori are determined. These studies of messy broadcasting have so far concentrated on finding worst-case broadcast time. However, such worst-case scenarios are extremely unlikely to occur in general. The average-case time for completing messy broadcasting in various network topologies can be found in [205].

- **Multiple message broadcasting**

  In the *multiple messages broadcasting*, it is assumed that the originator, also called *the broadcaster* [18], knows m messages $M_1, M_2, \ldots, M_m$ and has to transmit them to all the members of the network. The idea behind this model is that, when communicating large amounts of data, many systems break the data into sequences of messages (or packets) that are sent and received individually. This approach encourages an investigation into the problem of how to disseminate multiple messages efficiently in such systems. The problem of broadcasting multiple messages has been studied in several communication models. The telegraph model (unidirectional) has been studied in [55, 62, 97] and the telephone mode (bidirectional) has been studied in [19, 22, 208, 140, 142].

The problem of broadcasting multiple messages in the postal and LogP models of communication has been investigated in [20, 21, 181]. In these models, each processor can simultaneously send one message and receive another message, but message delivery involves some communication latency.

The multiport model generalizes the one-port model that has been widely investigated. There are examples of parallel systems with k-port capabilities for $k > 1$, such as the nCUBE/2 [223], the CM-2 (where k is the dimension of the hypercube in both machines), and transputer-based machines. This model has been investigated in [18]. A good survey of this model can be found in [245].

- **Broadcasting with randomly placed calls**

  The *broadcasting with randomly placed calls* model originated from the spreading rumors studies [231]. This model assumes that each informed member of a population (seen as an interconnected network) transmits the message to a other members of the population. This model has also been related to the spread of a communicable disease [199], generalizing the previous work on epidemics [183]. In this latter model it is assumed that the probability of transmitting a message depends on the age of the message and on the time since the informed node learned the message. In [198], a slightly different model is introduced, which assumes that whenever a node receives the information, it transmits it on average f times, where f may be a function of time. Further work on this model and its close connection to rumor and virus spreading can be found in [228, 229, 230, 131, 72, 73, 130, 48, 132, 102, 219, 83, 30, 122, 225].

  More recent work on random broadcasting relates to epidemics algorithms in replicated databases (see for example [100, 1, 182, 93]). In Chapter 3, we also introduce and analyze a random algorithm for broadcasting [218].

We have to mention that there is a considerable number of papers dedicated to broadcasting in mobile or ad-hoc networks. Since in this thesis we are dealing only with static networks, we omit to present these models.

There are three surveys partially or totaly dedicated to broadcasting in networks [164, 117, 170].

1.2 Minimum broadcast time problem

Throughout this thesis we adopt a model widely used in the literature, the *1-port constant* model [117]. That is, we consider that each node of the network can communicate with only one other node at a time, the transmitting time is constant, regardless of the size of the message, and the time needed to prepare or to forward the message is negligible. Therefore, all the calls involving different pairs of nodes will be synchronous and will take place in *rounds*.

We model the interconnected network as a simple undirected connected graph $G = (V, E)$, in which the members of the network are the vertices of G, and the communication links the edges of G.

Under the assumptions above, the broadcast process can be seen as a graph theoretic problem. Let $G = (V, E)$ be a graph and let v be a vertex in G. Consider now that v knows a piece of information, $I(v)$, which is unknown to all other vertices in $V \setminus \{v\}$. The broadcasting problem is to find a communication strategy, called *broadcast protocol*, or *broadcast scheme*, so that all nodes of G learn the piece of information $I(v)$.

There is no unified notation in literature for the broadcast scheme representation. For small graphs, an intuitive way to represent the broadcast scheme is to use labels of the edges in graph, corresponding to the number of the round in which the call has been made (like in Figure 1).

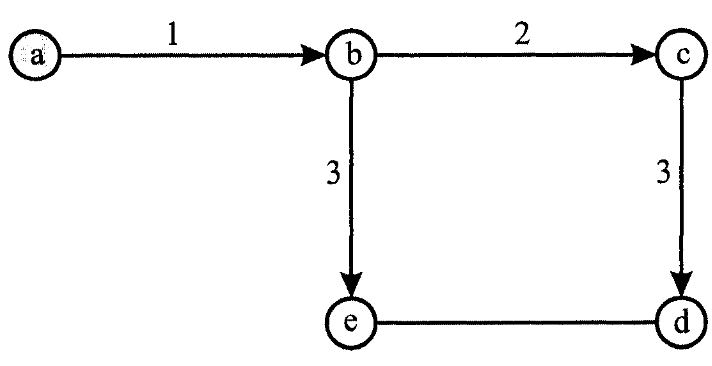

Figure 1: Edge labelling representing a broadcast scheme

Another way to represent a broadcast scheme is to use a set notation $\{(a \to b)\}$, with the meaning that a informs b [170]. Then, a broadcast scheme will consist of a ordered sequence of such sets. For example, for Figure 1, the corresponding broadcast scheme BS_a, having node a as originator is:

$$BS_a = \{(a \to b)\}, \{(b \to c)\}, \{(b \to e), (c \to d)\} \tag{2}$$

An algebraic method to represent a broadcast scheme involving a combined set-matrix notation has been used in [35] for a particular network topology (Knödel graphs).

In Chapter 2 we will introduce a more formal notation of a broadcast scheme. Intuitively, we define the broadcast scheme as a function such that, for the example in Figure 1, $B_a(b, 2) = c$, if and only if vertex b will transmit the information to vertex c at time 2 in a broadcast scheme started from a.

Under this model, we can see now each round of a broadcast scheme, also called *time-slot*, as a match between a subset of the informed vertices and a subset of the uninformed ones.

The broadcast time of a vertex v in a graph $G = (V, E)$, can be defined as the minimum number of rounds necessary to inform all the vertices of G starting from v. Such a broadcast scheme is also called an *optimal broadcast scheme.*

Denoting by $b(v)$ the broadcast time of a vertex v in G, the broadcast time of G, denoted by $b(G)$ by abuse of notation, can be formally defined as follows:

$$b(G) = \max_{v \in V} \{b(v)\} \tag{3}$$

For example, the broadcast time of the graph represented in Figure 1 is $b(G) = 3$.

The proof of the NP-completeness of the minimum broadcast time problem, under the *1-port constant* model, is due to D. J. Johnson and has been presented in [244]. In this paper, the problem is formulated as a *matching problem:*

Given a graph $G = (V, E)$ with a specified set of vertices $V_0 \subseteq V$ and a positive integer k, is there a sequence

$$V_0, E_1, V_1, E_2, V_2, \ldots, E_k, V_k$$

where $V_i \subseteq V$, $E_i \subseteq E$, E_i consists only of edges with exactly one vertex in V_{i-1},

$$V_i = V_{i-1} \cup \{v : uv \in E\},$$

and $V_k = V$? (V_i is the set of vertices who are informed at time i with calls along the edges in E_i.)

This matching problem reduces to the minimum broadcast time problem for the case when $|V_0| = 1$. The above problem can be further related to the three-dimensional matching problem (3DM) [244]:

Let $X = x_1, \ldots, x_m$, $Y = y_1, \ldots, y_m$, $Z = z_1, \ldots, z_m$ and let $M \subseteq X \times Y \times Z$. Does there exist a subset of M of size m such that each pair of elements of the subset disagree in all three coordinates?

This later problem has been shown to be NP-complete in [126].

Due to the NP-completeness of the minimum broadcast time problem, the research has been focused on two main directions: finding the broadcast time for particular graph topologies and developing heuristics for arbitrary topologies. Before discussing in detail each of these aspects, we present some general properties of the minimum broadcast time.

Property 1. $b(G) \geq \lceil \log_2 n \rceil$, *where n is the number of vertices in graph G.*

This is due to the fact that the number of informed vertices can be at most doubled in each round.

If we define the *radius* of a graph G by $rad(G) = \min_{v \in V} \max_{u \in V} d(v, u)$, and the *diameter* of G by $d(G) = max_{v,u \in V} d(v, u)$, where $d(v, u)$ denotes the distance between vertices v and u in G, we have the following property [170].

Property 2. $rad(G) \leq d(G) \leq b(G)$

This lower bound can be slightly improved in some cases [170].

Property 3. *Let G be a graph of diameter D. If there exist three different vertices u, v_1, and v_2 with both v_1 and v_2 at distance D from u, then $b(G) \geq D + 1$.*

Given two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ we say that G_1 is a *spanning subgraph* of G_2 if $V_1 = V_2$ and $E_1 \subseteq E_2$. We have the following property [170].

Property 4. $b(G_1) \leq b(G_2)$

A. The broadcast time for some particular graph topologies

In what follows, we present some of the results for particular topologies along with their formal definition as given in [170]. For a more comprehensive description and study of these network architectures we direct the reader to [203].

a) **The Path P_n**

Definition 1. *[170] The (simple) path of length n, denoted by P_n is the graph whose nodes are all integers from 1 to n and and whose edges connect each vertex i with $i+1$, for $1 \leq i < n$.*

Property 5. *[170]* $b(P_n) = n - 1$.

b) **The Cycle C_n**

Definition 2. *[170] The cycle (ring) of length n, denoted by C_n, is the graph whose nodes are integers from 1 to n and whose edges connect each vertex i $(i \leq i \leq n)$ with $(i+1)(\mathrm{mod} n)$.*

Property 6. *[170]* $b(C_n) = \lceil n/2 \rceil$.

c) **The complete tree T_k^m**

Definition 3. *[170] The complete k-ary tree of height m, denoted by T_k^m, is the graph whose nodes are all k-ary strings of length at most m and whose edges connect each string α of length i $(0 \leq i \leq m)$ with the string αa, $a \in \{0, \ldots, k-1\}$, of length $i+1$.*

Property 7. *[170]* $b(T_k^m) = (k-1)m + 1$.

d) **The complete graph K_n**

Definition 4. *[170] The complete graph (clique) of size n, denoted by K_n, is the graph whose nodes are all integers from 1 to n and whose edges connect each integer i, $1 \leq i \leq n$, with each integer j, $i \neq j$.*

Property 8. *[170]* $b(K_n) = \lceil \log_2 n \rceil$.

e) **The hypercube H_m**

Definition 5. *[170] The (binary) hypercube of dimension m, denoted by* H_m*, is the graph whose vertices are all binary strings of length m and whose edges connect those binary strings which differ in exactly one position.*

Property 9. *[99]* $b(H_m) = m$.

f) **The cube-connected cycles** CCC_m

Definition 6. *[170] The cube-connected cycles of dimension m, denoted by* CCC_m*, has vertex set* $V_m = \{0, 1, \ldots, m-1\} \times \{0,1\}^m$*, where* $\{0,1\}^m$ *denotes the set of length-m binary strings. For each vertex* $v = \langle i, \alpha \rangle \in V_m$*,* $i \in \{0, 1, \ldots, m-1\}$*,* $\alpha \in \{0,1\}^m$*, we call i the level and* α *the position-within-level (PWL) string of v. The edges of* CCC_m *are of two types: for each* $i \in \{0, 1, \ldots, m-1\}$ *and each* $\alpha = a_0 a_1 \ldots a_{m-1} \in \{0,1\}^m$*, the vertex* $\langle i, \alpha \rangle$ *on level i of* CCC_m *is connected*

- *by a straight-edge with vertex* $\langle (i+1)(\bmod m), \alpha \rangle$ *on level* $(i+1)(\bmod m)$
- *by a cross-edge with vertex* $\langle i, \alpha(i) \rangle$ *on level i,*

with $\alpha(i) = a_0 \ldots a_{i-1} \overline{a}_i a_{i+1} \ldots a_{m-1}$ *and* $\overline{a}$ *denotes the binary complement of a.*

Property 10. *[206]* $b(CCC_m) = \lceil 5m/2 \rceil - 1$.

g) **The butterfly network** BF_m

Definition 7. *[170] The butterfly network of dimension m, denoted by* BF_m*, has vertex-set* $V_m = \{0, 1, \ldots, m-1\} \times \{0,1\}^m$*, where* $\{0,1\}^m$ *denotes the set of length-m binary strings. For each vertex* $v = \langle i, \alpha \rangle \in V_m$*,* $i \in \{0, 1, \ldots, m-1\}$*,* $\alpha \in \{0,1\}^m$*, we call i the level and* α *the position-within-level (PWL) string of v. The edges of* BF_m *are of two types: for each* $i \in \{0, 1, \ldots, m-1\}$ *and each* $\alpha = a_0 a_1 \ldots a_{m-1} \in \{0,1\}^m$*, the vertex* $\langle i, \alpha \rangle$ *on level i of* BF_m *is connected*

- *by a straight-edge with vertex* $\langle (i+1)(\bmod m), \alpha \rangle$ *and*
- *by a cross-edge with vertex* $\langle i, \alpha(i) \rangle$

on level $(i+1)(\bmod m)$*. Again,* $\alpha(i) = a_0 \ldots a_{i-1} \overline{a}_i a_{i+1} \ldots a_{m-1}$ *and* $\overline{a}$ *denotes the binary complement of a.*

Property 11. *[190]* $b(BF_m) \leq 2m - 1$.

h) **The shuffle-exchange network** SE_m

Definition 8. *[170] The shuffle-exchange network of dimension m, denoted by* SE_m*, is the graph whose vertices are all binary strings of length m and whose edges connect each string* αa*, where* α *is a binary string of length* $m-1$ *and* $a \in \{0,1\}$*, with the string* $\alpha\bar{a}$ *and with the string* $a\alpha$*. (An edge connecting* αa *with* $\alpha\bar{a}$ *is called an exchange edge, and an edge connecting* αa *with the string* $a\alpha$ *is called a shuffle edge.)*

Property 12. *[169]* $b(SE_m) \leq 2m - 1$.

g) **The de Bruijn network** DB_m

Definition 9. *[170] The de Bruijn network of dimension m, denoted by* DB_m *is the graph whose vertices are all binary strings of length m and whose edges connect each string* $a\alpha$*, where* α *is a binary string of length* $m-1$ *and* $a \in \{0,1\}$*, with the strings* αb*, where* $b \in \{0,1\}$*.*

Property 13. *[38]* $b(DB_m) \leq 3/2(k+1)$.

h) **The grid network** $G[a_1 \times a_2 \times \ldots \times a_d]$

Definition 10. *[170] The d-dimensional grid (mesh) of dimensions* $a_1, a_2, \ldots, a_d$*, denoted by* $G[a_1 \times a_2 \times \ldots \times a_d]$ *is the graph whose nodes are all d-tuples of positive integers* $(z_1, z_2, \ldots, z_d)$*, where* $1 \leq z_i \leq a_i$*, for all i* $(0 \leq i \leq d)$*, and whose edges connect d-tuples which differ in exactly one coordinate by one.*

Property 14. *[98]* $b(G[m \times n]) = m + n - 2$.

More recent results regarding broadcasting in the grid networks can be found in [243, 65].

i) **Knödel graphs** $W_{g,n}$

Definition 11. *[108] Knödel graphs* $W_{g,n}$ *are defined as undirected graphs* $G(V,E)$*, with* $V = \{0, \ldots, n-1\}$*, n even, and the set of edges* $E = \{(i,j) \mid i + j = (2^k - 1) \bmod n\}$*, where* $0 \leq i,j \leq n-1$*,* $1 \leq k \leq g$*, and* $1 \leq g \leq \lfloor \log_2 n \rfloor$*.*

Property 15. *[191]* $b(W_{k,2^k}) = k$.

Property 16. *[186]* $b(W_{k-1,2^k-2}) = k$.

The broadcast process has also been studied for other particular network topologies, such as general trees [244], Kautz graphs [167], pancake and star graphs [39], chordal rings [67], recursive circulants [224], banyan-hypercube [29], cycle prefix digraphs [68], etc. Some general results have been obtained for networks under certain constraints, such as bounded degree networks [37, 212] or planar graphs [165].

B. Heuristics for the minimum broadcast time problem

The first attempt to exactly solve the minimum broadcast time problem is due to Scheuermann and Edelberg [238], who implemented a backtracking algorithm. Another exact algorithm, based on dynamic programming, is due to Scheuermann and Wu [239]. A backtracking algorithm for bounded degree networks is described in [134].

Since the exact algorithms are not efficient for large graphs, in [239] several heuristics have been proposed for achieving efficient, near-optimal schemes. They are based on finding a least-weight maximum matching in a bipartite graph and work in $O(n^3m)$ time, for the whole graph, where n is the number of vertices and m is the number of edges. In fact, most of the heuristics developed after this paper use matching-based methods. This is due to the fact that each round of calls can be seen as a matching process between the set of informed vertices and the set of uninformed ones.

A matching-based method has been employed in [28] to derive a $O(Rn^2m\log n)$ time complexity heuristic, where R is the minimum number of rounds needed to broadcast, which is further lower bounded by $\lceil \log n \rceil$ and upper bounded by $n-1$. The time complexity has been further improved by Harutyunyan and Shao in [154]. Using a matching-based method, they provided a $O(Rnm)$ heuristic, where R is the minimum number of rounds needed to broadcast. A matching-based method has also been used for *partial meshes* topology in [120].

The first approximation result is presented in the work of Ravi [232] who provided a $O\left(\frac{\log^2 n}{\log\log n}\right)$ approximation algorithm working in $O(mn^2\log^2 n)$ time. This algorithm is based on the construction of a spanning tree of approximately minimum poise [1]. The result has been improved by Elkin and Kortsarz [88], who gave

[1]The poise of a tree is defined to be the maximum degree of the tree + the diameter of that tree.

a $O\left(\frac{\log n}{\log\log n}\right)$ approximation algorithm working in $O(mn^2 \log^3 n)$ time (both algorithms are using a minimum flow approach). Moreover, Kortsarz and Peleg [192] gave a $O(\sqrt{n})$ additive approximation algorithm.

Other techniques have also been employed to approximate the minimum broadcast time problem. In [168], a genetic algorithm using a global precedence vector is used to derive a heuristic working in $O(mn^3)$ time. In [17] an integer programming formulation is given to derive a $O(\log n)$ approximation algorithm. In [26], a general approach for structured communications is presented, which can be applied to solve the minimum broadcasting time problem.

In Section 2.2 we present an iterative heuristic for the minimum broadcast time problem working in $O(kmn)$, where k is the number of iterations, which controls the "degree of optimality" of the solution. Even though the optimality is not guaranteed, we obtained exact solutions for relatively big graphs, after a reasonable number of iterations. If we fix $k = \lceil \log n \rceil$, the benchmark results are comparable to those presented in [154]. Note that this is the first iterative approach for this problem.

It is worth mentioning here that the minimum broadcast problem remains NP-complete even for particular topologies [175], or for bounded degree networks [80, 212]. Other results regarding the inapproximability of this problem can be found in [240].

Since throughout this thesis we are using only the *1-port* constant model we do not describe heuristics and results regarding other models of broadcasting.

1.3 Minimum broadcast graph problem

According to Property 1, the minimum time necessary to broadcast in any graph is $\lceil \log_2 n \rceil$, where n is the number of vertices in graph. We also saw that the complete graph is one of the topologies that minimizes the broadcast time (Property 8). Unfortunately, the complete graph is not minimal with respect to the number of edges. That is, we can delete some edges and still be able to broadcast from any originator in $\lceil \log_2 n \rceil$ time. This optimization problem becomes critical in the cases of scarce network resources (in supercomputing for example). Therefore, much research has been focused on designing network topologies which support logarithmic time broadcasting while optimizing the number of connections.

A graph with broadcast time $\lceil \log_2 n \rceil$ is called *broadcast graph*, or shortly *bg*. A broadcast graph with minimum number of edges is called *minimum broadcast graph*, or shortly *mbg*. Finding an *mbg* for a given number of vertices is known as the *minimum broadcast graph problem*.

The *broadcast function* $B(n)$ is defined as the number of edges of an *mbg* on n vertices. Finding $B(n)$ turns out to be a difficult task, even for small values of n. The exact value of $B(n)$ is known when $n = 2^k$ and $n = 2^k - 2$.

- $B(2^k) = k2^{k-1}$, for all $k < 0$.

  This value is attained by three non-isomorphic families of graphs:

  - the hypercube of dimension k [95];
  - the recursive circulant $G(2^k, 4)$ [224];
  - the Knödel graph $W_{k,2^k}$ [191].

- $B(2^k - 2) = (k-1)(2^{k-1} - 1)$, for all $k \geq 3$.

  This value is attained by the Knödel graph $W_{k-1,2^k-2}$ [186].

Apart from these findings, the exact values of $B(n)$ are known only for some particular values of n, mainly under 63. Table 1 summarizes these values along with their references.

Since the exact values of $B(n)$ are known only for a limited number of values of n, many papers have been dedicated to finding sparse broadcast graphs in order to improve the upper bounds of $B(n)$. Chau and Liestman [51] developed an algorithm that constructs broadcast graphs by interconnecting 5, 6 and 7 smaller broadcast graphs. Gargano and Vaccaro [127] proposed three algorithms based on an interconnection of hypercubes of small dimension to build up larger broadcast graphs. Chen [52] presented a method similar to the second algorithm of Gargano and Vaccaro, and then suggested the recursive application of his last method to construct larger broadcast graphs. Khachatrian and Haroutunian [186, 187] presented compounding methods based on the vertex cover in order to construct broadcast graphs with $O(L(n)n)$ edges, where $L(n)$ is the number of leading 1's in the binary representation of $n-1$ [139]. Similar results have been obtained by Grigni and Peleg in [133] who use compounding methods relying on hypercubes and generalized Fibonacci numbers.

Bermond et al. [36] proposed four methods for constructing broadcast graphs and used them to produce new broadcast graphs for $18 \leq n \leq 63$. Ventura and Weng [248]

Table 1: The values of $B(n)$ for some particular values of n.

n	$B(n)$	**Ref.**
1	0	[99]
2	1	[99]
3	2	[99]
4	4	[99]
5	5	[99]
6	6	[99]
7	8	[99]
9	10	[99]
10	12	[99]
11	13	[99]
12	15	[99]
13	18	[99]
14	21	[99]
15	24	[99]
16	32	[99]
17	22	[213]
18	23	[36, 256]
19	25	[36, 256]
20	26	[211]
21	28	[211]
22	31	[211]
26	42	[237, 255]
27	44	[237]
28	48	[237]
29	52	[237]
30	60	[36]
31	65	[36]
32	80	[99]
58	121	[237]
59	124	[237]
60	130	[237]
61	136	[237]
62	155	[95]
63	162	[197]
127	389	[258]
1023	4650	[241]
4095	22680	[241]

developed a method based on the concepts of aggregated nodes and aggregated edges (which are used to replace ordinary nodes and edges, respectively, of known *mbg*'s, for $9 \leq n \leq 15$) to construct sparse broadcast graphs. Another class of combination methods using compound graphs has been developed by Bermond et al. [34].

A more general method that allows for systematic vertex deletion was proposed by Weng and Ventura [251]. The main idea of their method, called the *doubling procedure*, is a center node set, defined by so-called *official broadcasting.* The same method has been investigated in [82] by investigating official broadcasting and center node sets in more detail and developing iterative algorithms based on these constructions. Most of the $B(n)$ upper bounds mentioned above have been improved in [144]. They used compound construction between hypercubes or Knödel graphs and other broadcast graphs, merging after compounding methods, or a generalized construction of Ahlswede et al. [2].

Most of the compounding methods presented above construct broadcast graphs on an even number of vertices. Recently, a vertex addition method has been used in order to obtain broadcast graphs on odd number of vertices [141, 158].

We have to mention here two restricted classes of minimum broadcast graphs. The first one consists of *hierarchical broadcast networks*, which are defined as broadcast networks whose every connected induced subgraph is also a broadcast network. Fraigniaud showed in [112] that the minimum number of edges of a hierarchical broadcast network of order n is $\left\lceil \frac{n(n-2)}{2} \right\rceil$ for any $n \geq 4$. Another restricted class of graphs consists of *minimum broadcast trees*, which poses the problem of finding a tree such that the broadcast time is minimized. This problem has been studied in [226, 195, 185] and has been generalized in [146].

We conclude this section by presenting some general properties of $B(n)$. The first property is a direct consequence of Farley's method [95] for constructing *bg*'s.

Property 17. *[95]* $B(n) \leq \frac{n\lceil \log_2 n \rceil}{2}$.

The next property, due to Gargano and Vaccaro, is an upper bound of $B(n)$ and is based on a binomial tree construction.

Property 18. *[127]* $B(n) \geq \frac{n}{2}\left(\lceil \log_2 n \rceil - \log_2\left(1 + 2^{\lceil \log_2 n \rceil} - n\right)\right)$.

The following property is due to Grigni and Peleg.

Property 19. *[133] $B(n) \in \Theta(nL(n))$, where $L(n)$ is the number of leading 1's in the binary representation of $n-1$.*

The following two properties are due to Harutyunyan and Liestman.

Property 20. *[144] $B(n) \leq n(m-k+1) - 2^{m-k} - \frac{1}{2}(m-k)(3m+k-3) + 2k$, for $n = 2^m - 2^k - r$, with $0 \leq k \leq m-2$ and $0 \leq r \leq 2^k - 1$.*

Property 21. *[147] $B(n)$ is non-decreasing for values of n in the interval $2^{m-1} + 1 \leq n \leq 2^{m-1} + 2^{m-3}$.*

1.4 Thesis summary

The thesis is structured as follows. Chapter 2 introduces the minimum broadcast time function and presents two new fundamental properties of this function. One of the properties yields an iterative heuristic for the minimum broadcast time problem, which is the first iterative approach for this problem. The other property establishes a connection between the broadcast time of each vertex and the graph density. This study is motivated by the fact that most of the random graph generators are using graph density as a main parameter. The results from this chapter appear in [150] and [152].

Chapter 3 is dedicated to the study of the number of broadcast schemes in networks. We give tight bounds for the number of broadcast schemes in networks and we present two applications: an algorithm to enumerate all the broadcast schemes in a network and a new random algorithm for broadcasting. The results from this chapter appear in [218].

Chapter 4 is dedicated to the study of the spectra of Knödel graphs and some of its applications. This study is motivated by the fact that, among the only three known infinite families of minimum broadcast graphs, namely hypercube, recursive circulant, and Knödel graph, the last one has the smallest diameter and the highest global fault tolerance. The results from this chapter appear in [149].

Chapter 5 introduces a new measure of the fault tolerance of a network, which we name the *global fault tolerance*. Based on this metric, we make a comparative study of the above mentioned classes of minimum broadcast graphs and we show that, from this point of view, there are better networks than the hypercube. The results from this chapter appear in [153].

As a result of the relative wide area covered by this thesis, most of the specific definitions will be given in the places where they are used. Also, because we wanted the chapters to be as much as possible self contained, some of the definitions and statements will be repeated, in different contexts or forms.

Chapter 2

Two properties of the minimum broadcast time function

Given a graph $G = (V, E)$, the *minimum broadcast time function*, or shortly the b-function, is defined as $b : V \to N$, such that $b(v)$ represents the minimum time necessary to inform all the vertices of G, starting from v. If $S = \{v_1, v_2, ..., v_k\} \subseteq V$ is a subset of vertices, then $b(S)$ is the multiset $\{b(v_1), b(v_2), ..., b(v_k)\}$. By abuse of notation, the broadcast time of G is denoted by $b(G)$ and is defined as $b(G) = \max_{v \in V} \{b(v)\}$.

The minimum broadcast time problem has been proven to be NP complete [244], even for bounded degree graphs [80]. The values of the minimum broadcast time function are known for a very restricted class of graphs, mainly regular ones, and very little is known about this function in general.

In the first part of this chapter we describe a new property which connects the values of the b-function and the behavior of the optimal broadcast schemes. We prove that this new property of the b-function is true for arbitrary trees and we conjecture it for arbitrary graphs [152].

Based on this property we exhibit an iterative global algorithm for the minimum broadcast time problem which performs very well in practice [150]. The algorithm and the simulation results are presented in the second section of the chapter.

Some of the heuristics developed to solve the minimum broadcast time problem have been tested on random graphs which were generated using common generators [150, 154, 168]. Two of the main parameters of the random graph generators are the number of vertices and the density of the graph. In the last section we establish a connection between the minimum broadcast time function and these two parameters by upper bounding the range in which the b-function can take values [152].

2.1 The b-function and optimal broadcast schemes

We denote by $G = (V, E)$ an undirected graph with V the set of vertices and E the set of edges. If we consider the broadcasting process originated in vertex v, at time t, an already informed vertex, say u, will choose one of its uninformed neighbors, say w, and will transmit the information to w. A broadcast scheme will assign to each vertex and time slot another vertex to inform. Therefore, we can associate to an optimal broadcast scheme originated in v a function $B_v : (V, \Gamma) \to V$, where $\Gamma = \{1, ..., b(G)\}$, such that $B_v(u, t) = w$ if and only if vertex u will transmit to vertex w at time t in that optimal broadcast scheme started from v. This is a formal way of representing a broadcast scheme. Note that this function is not defined for the whole domain (V, Γ). That is, there are time-slots t for which $B_v(u, t)$ is not defined if vertex u will not transmit at time t.

The crux of an optimal broadcast scheme resides in the way of choosing the next neighbor to inform. If we denote by $N_v(u, t)$ the set of uninformed neighbors of a vertex u in the graph at time t in a broadcast scheme originated in v, the following theorem establishes a connection between the minimum broadcast time function and optimal broadcast schemes for arbitrary trees.

Theorem 1. *Consider a tree T, an originator vertex v, and the b-function defined for each vertex of T. During any optimal broadcast scheme starting from v, at an arbitrary time $t \leq b(G)$, an informed vertex u will call vertex z, which has the minimum value of the b-function among all uninformed neighbors of u. Formally, $B_v(u, t) = z$ implies $b(z) = \min\{b(N_v(u, t-1))\}$.*

Proof. Let u be an informed vertex at time-slot $t - 1$. Assume by contradiction that vertex u will call at time-slot t one of its neighbours, say w, with a greater broadcast time than z, which has the smallest broadcast time among all uninformed neighbors of u. Formally, we assume for the purpose of contradiction that $B_v(u, t) = w$ and $b(w) > b(z) = \min\{b(N_v(u, t-1))\}$.

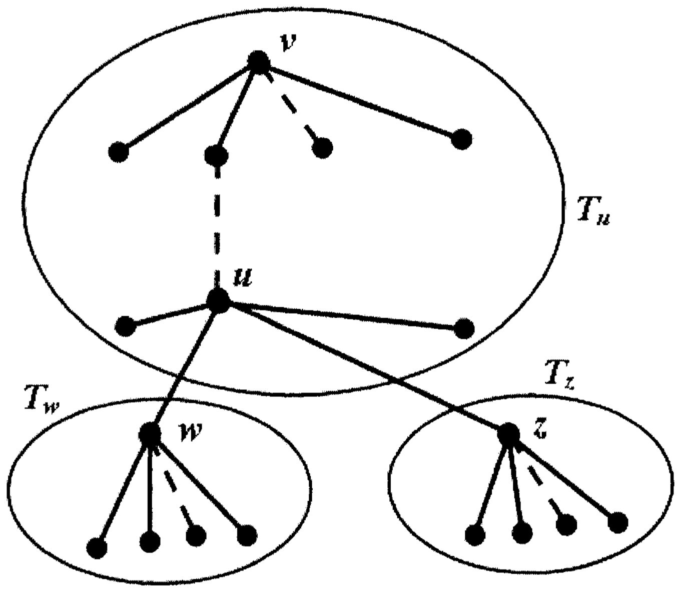

Figure 2: Tree T and the subtrees generated by u, w, and z.

We denote by T_w the subtree rooted at w, and by T_z the subtree rooted at by z, both in an optimal broadcast scheme having v as originator. We denote by $T_u = T - T_w - T_z$, the tree remaining after deleting T_w and T_z from T, rooted at u (see Figure 2). Also, we denote by:

- $b\,(w \to T_w)$ the minimum time needed for w to broadcast in subtree T_w
- $b\,(z \to T_z)$ the minimum time needed for z to broadcast in subtree T_z
- $b\,(u \to T_u)$ the minimum time needed for u to broadcast in subtree T_u

There are 13 possible relationships between $b\,(w \to T_w)$, $b\,(z \to T_z)$, and $b\,(u \to T_u)$, with respect to "<" and "=". These 13 possibilities can be grouped in six cases:

a) $b\,(u \to T_u) < b\,(z \to T_z) < b\,(w \to T_w)$
$b\,(z \to T_z) < b\,(u \to T_u) < b\,(w \to T_w)$
We chose to explain this case in more detail. All the remaining cases have a similar approach.

In this case we have:
$b\,(w) \leqslant b\,(w \to T_w) + 1$, and

$b(z) \geqslant b(w \to T_w) + 2$

The first relation is due to the fact that w can make only one call outside the tree T_w and the rest are all inside calls. Since the broadcast time of u and z are less than $b(w \to T_w)$, one of the last calls will be in T_w.

The second relation is due to the fact that z must first call u and then u must first call w, since broadcasting in T_w takes the most time.

Adding the two relations we get $b(w) + 1 \leqslant b(z)$ (contradiction).

b) $b(z \to T_z) = b(w \to T_w) = b(u \to T_u)$
$b(u \to T_u) < b(w \to T_w) = b(z \to T_z)$
In these cases $b(w) = b(z)$ (contradiction).

c) $b(u \to T_u) = b(z \to T_z) < b(w \to T_w)$
In this case we have:
$b(w) \leqslant b(w \to T_w) + 1$, and
$b(z) \geqslant b(w \to T_w) + 1$
Adding we get $b(w) \leqslant b(z)$ (contradiction).

d) $b(z \to T_z) < b(w \to T_w) < b(u \to T_u)$
$b(w \to T_w) < b(z \to T_z) < b(u \to T_u)$
$b(w \to T_w) = b(z \to T_z) < b(u \to T_u)$
In this case we have either:
$b(w) = b(z) = b(u \to T_u) + 1$, or
$b(w) = b(z) = b(u \to T_u) + 2$ (contradiction).

e) $b(z \to T_z) < b(w \to T_w) = b(u \to T_u)$
In this case we have:
$b(w) \leqslant b(w \to T_w) + 2$, and
$b(z) \geqslant b(w \to T_w) + 2$
Adding we get $b(w) \leqslant b(z)$ (contradiction).

f) $b(u \to T_u) < b(w \to T_w) < b(z \to T_z)$
$b(w \to T_w) < b(u \to T_u) < b(z \to T_z)$
$b(w \to T_w) = b(u \to T_u) < b(z \to T_z)$
$b(w \to T_w) < b(u \to T_u) = b(z \to T_z)$

All these relations imply that:

$$b(w \to T_w) < b(z \to T_z) \tag{4}$$

Inequality (4) shows that u cannot call w before z since the obtained broadcast scheme will not be optimal. More formally, $b(w \to T_w) < b(z \to T_z) \Rightarrow B_v(u,t) \neq w$. □

Note that the converse is not true since we can find more than one neighbor of a vertex with the same minimum broadcast time. Formally, $b(z) = \min\{b(N_v(z,t-1))\}$ does not necessarily imply $B_v(u,t) = z$, since there may be another vertex y such that $b(y) = \min\{b(N_v(z,t-1))\}$ and $B_v(u,t) = y$.

The same property can be proven in a similar way for the product graph $P_2 \times P_n$. We have explored a large variety of small irregular graphs with known broadcast scheme without finding a counterexample to this property. Also, based on this property, we have designed an iterative global heuristic in order to find the minimum broadcast time of an arbitrary graph, which works very well in practice [150]. All these results convinced us to make the following conjecture.

Conjecture *Given an undirected connected graph G, an arbitrary originator v, and a label attached to each vertex corresponding to its minimum broadcast time, during an optimal broadcast process starting from v, an informed vertex will always call one uniformed neighbor with the smallest label.*

2.2 An iterative algorithm for the minimum broadcast time problem

Due to the NP-completeness some heuristic algorithms have been developed in order to give approximate solutions for the minimum broadcast time problem ([239, 232, 192, 168, 119, 28, 154, 87]). Most of these heuristics are based on graph matching techniques. There are two main criteria to compare between these heuristics: the complexity of the algorithm and the "optimality" of the solution. While the first criterion is relatively easy to compute, the second one raises some questions, especially for randomly generated graphs, for which an optimal solution is generally not known a priori. There are three approaches to overcome this problem: to obtain theoretical results regarding the inapproximability of the problem, to run the algorithm on graphs with known broadcast time, mainly regular ones, or to do benchmarks on random graphs, generated using common generators.

To our knowledge, the best result regarding the inapproximability of the minimum broadcast time problem is due to Schindelhauer, who showed in [240] that the broadcast time cannot be approximated within a factor of $\frac{57}{56} - \varepsilon$, for any $\varepsilon > 0$. The best result regarding the approximability of the minimum broadcast time problem is due to Elkin and Kortsarz [88] who gave a $O\left(\frac{\log n}{\log \log n}\right)$ approximation algorithm.

The best known complexity is $O(Rnm)$, for the entire graph [154], where R is the approximated broadcasting time, n is the number of vertices, and m is the number of edges.

In this section we describe an iterative global algorithm which works in $O(knm)$ time for the entire graph, where k is the number of iterations, n is the number of vertices, and m is the number of edges. The algorithm is iterative in the sense that, at each iteration, it attempts to find a better broadcast scheme than the previous known one and it is global in the sense that it gives broadcast schemes for all the vertices of the graph and not just for one vertex. In order to capture this global behavior, we define the *average broadcast time* of a graph G:

$$\overline{b(G)} = \frac{1}{n} \sum_{v=1}^{n} b(v) \tag{5}$$

We have experimentally found that, during the first $\log n$ iterations, the approximate solution has the greatest rate of convergence, which gives a $O(nm \log n)$ average complexity. We have extensively tested this algorithm on a wide range of graph topologies and we present here the results, which are similar to those from [154].

Note that this algorithm approximates not only the minimum broadcast time of a graph but also the values of the $b-$function for all the vertices in the graph.

2.2.1 Algorithm description

We use the adjacency lists as a data structure for keeping G in memory, each list being organized as a hash table of length n. To each vertex $v \in G$ we attach a label, corresponding to its known broadcast time. Since, at the beginning, there is no broadcast time available, we assign to all vertices a broadcast time $b(v) = n$, the number of vertices, which is clearly an upper bound for $b(v)$. Also, for each adjacency list, denoted by *adj* (v), we keep a pointer to the neighbour with the smallest label.

The algorithm repeatedly parses all the vertices from G, considering each of them at a time as originators. The number of iterations will correspond to how many times we parse the whole set of vertices of G.

Once an originator is chosen, say v, we construct the adjacency lists for all vertices, picking up the neighbours in random order and inserting them in the hash tables according to their broadcast time. Then we include v in the set S of informed vertices and we start to broadcast the message from the vertices belonging to S to the rest of vertices from $G - S$. The rule of choosing one neighbour to inform by a vertex, say u, is: at each time every informed vertex will call its uninformed neighbour, say w, with the smallest known broadcast time corresponding to its label, $b(w)$. If more than one vertex has the smallest label, the algorithm will pick up the first one. The edge used for this call will be deleted from both adjacency lists: *adj* (u) and *adj* (w). The new informed vertices will be included in *NI*, the set of new informed vertices after a round of calls, only if their adjacency list is not empty. If *adj* (u) is also empty, we delete u from S. When all the vertices from S have made a call, we increment the broadcast time by one and we attach the set *NI* to the set S, setting $NI = \emptyset$.

This process stops when all the vertices from G are informed. At this moment, we have a new broadcast scheme available for the vertex v, and a new broadcast time. If this new broadcast time is smaller then the previously known one, we modify its label $b(v)$ accordingly and continue by choosing the next vertex in G as originator. One iteration is considered finished when we have considered all the vertices from G as originators.

We denote by *MAX_ITER* the number of desired iterations. This can be seen as a free parameter for our algorithm and it will control the "optimality" of the resulted broadcast schemes. This parameter can also be set automatically to be the number of iterations after which there are no changes in the broadcast time of any of the vertices form G. As you will see in the next section, this approach can be tricky since the value of the labels decreases very fast at the beginning but can be stationary for a long time after a certain number of iterations, until it decreases again.

Formally, the algorithm can be written as follows:

```
for i = 0 to MAX_ITER
  for each v ∈ G
    build the adjacency lists of all vertices from G
    S = {v}
    informed_vertices = 1
    b_time = 0;   NI = ∅
    while informed_vertices < |G|
      for each u ∈ S
        w = neighbor of u with b(w) = min.
        delete u from adjacency list of w
        delete w from adjacency list of u
        if adj(w) is not empty
          NI = NI ∪ {w}
        if adj(u) is empty
          S = S − {u}
        informed_vertices = informed_vertices + 1
      end for
      b_time = b_time + 1
      S = S ∪ NI
    end while
    if b(v) > b_time
      b(v) = b_time
  end for
end for
```

2.2.2 Algorithm analysis

The crux of the algorithm is the method of choosing the next neighbor to inform: the one with the minimum known broadcast time, which corresponds to its label. This is clearly suggested by Theorem 1 and the simulation results sustain the conjecture that we have made in the previous section.

In order to analyse the time-complexity of the algorithm we have to analyse the following operations:

1. The outer **for** loop represents the number of iterations, $MAX\_ITER = k$. This is a free parameter for the algorithm and is controlling the "solution's optimality". Even though there is still no theoretical support for the necessary number of iterations in order to exactly converge to the optimal solution, experimental results suggest that during the first $\log n$ iterations, the convergence rate is the highest, and the solution is near optimal (see Section 2.2.3, Figure 3). Nevertheless, for some graphs, as full binary trees, or binomial trees, the algorithm was able to discover the optimal solution after only one iteration. This number also relies on the starting vertex and the parsing order but, for large graphs and bigger number of iteration, this dependency is very weak.

2. The middle **for** loop, which gives n operations, corresponding to the number of vertices.

3. The adjacency lists building operation, which takes $O(m)$ time, where m is the number of edges from G.

4. The combination between the inner **for** and **while** loops, corresponding to the total number of calls, which is n.

5. The number of operations necessary to retrieve the neighbor with the smallest label. This can be done in $O(1)$ by keeping a hash table and a pointer to the first non-empty entry in this table.

6. The number of operations necessary to delete the vertices u and t from the adjacency lists. This can be also done in $O(1)$ by keeping an n^2 array with the vertices addresses in the adjacency lists.

From the above analysis, the driving operations for complexity are in the steps 1), 2), and 3), yielding a $O(kmn)$ time complexity.

One could expect that the complexity will increase by increasing the degree of the vertices in the graph. Nevertheless, experimental results show that, once the graph density increases and the graph tends to the complete graph, the number of necessary iterations needed decreases. For the complete graph or for trees only one iteration is needed. The experimental results show that the graphs with density close to 0.5 require a greater number of iterations than those with densities close to either 0 or 1.

We are using the adjacency list as data structure for keeping the graph in memory, and an $n \times n$ array with pointers to vertices, which yields a $O(n^2)$ space complexity.

Since, at the beginning, all the vertices have the same label, there is no difference between a random algorithm (see [100]) and this one, for the first originator. Once this originator has a smaller broadcast time assigned, during the next calls there will already be an available choice to make. Even though this "warm up" part of the algorithm seems to be somehow "uncontrollable", the results derived from our experiments show that the value of labels decreases from the beginning at a very high rate (Figure 3).

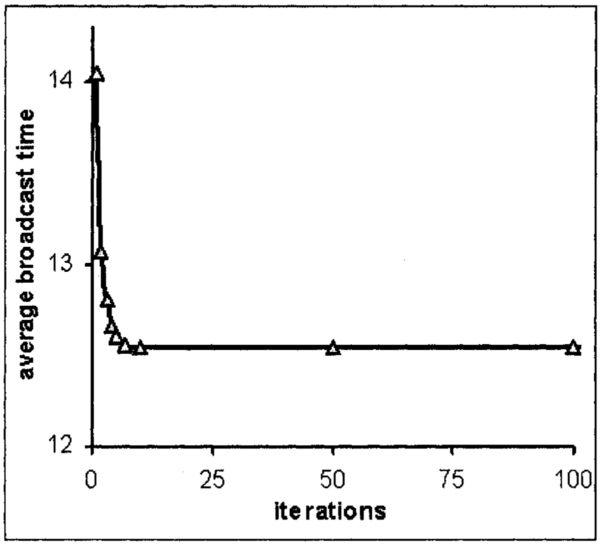

Figure 3: The average broadcast time function of the number of iteration steps for a transit-stub randomly generated graph on 100 vertices. The majority of vertices got their broadcast time approximation after the first 6 steps.

2.2.3 Experimental results

As we mentioned in introduction, there are two main types of experiments characteristic to such heuristics: to run them on graphs with known broadcast time, and to do benchmarks on random graphs generated using common generators. We studied the algorithm behavior on both types:

- Graphs with known broadcast time: full binary trees bT, binomial trees BT, hypercubes on dimension d (H_d), Knödel graphs on 2^d vertices and dimension d ($W_{d,2^d}$), cube-connected-cycles (CCC), de Bruijn graphs (DB), butterfly (BF), and shuffle exchange (SE), all of them up to dimension 10.

- Two types of random graphs: GT-ITM pure random and GT-ITM transit-stub random graphs [253].

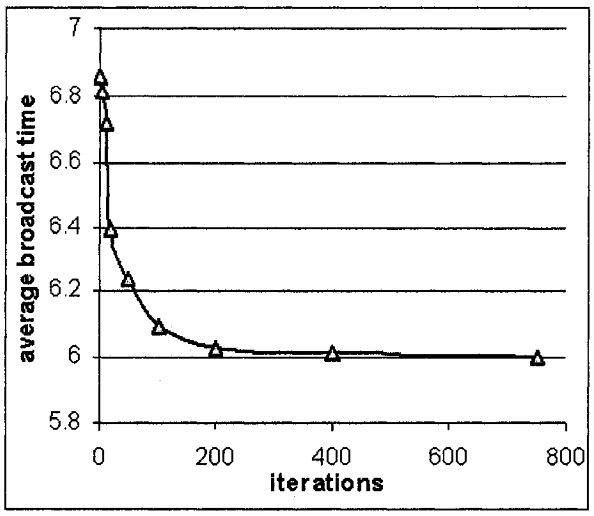

Figure 4: The average broadcast time function of the number of iteration steps for the hypercube on dimension 6.

Two types of tests were done regarding the "solution's optimality": determining the average broadcast time after $\lceil \log_2 n \rceil$ iterations for the whole graph and determining the number of iterations needed in order to obtain the optimal solution for the graphs, where this is known (hypercubes and Knödel graphs) and where this number is achievable in a reasonable amount of time.

In our experiments we observed a rapid decrease in the average broadcast time during the first $\lceil \log_2 n \rceil$ iterations (Figure 3). More than that, for the hypercube in dimension 6, the solution converges to an optimal broadcast scheme after 750 iterations (Figure 4).

The tests on regular graphs are not very relevant for this algorithm since the main idea behind choosing the neighbour with the minimum broadcast time is not exploited due to the uniform distribution of the b-function values among graph's vertices. From this point of view, once all the vertices get almost the same label, there is no substantial difference between this algorithm and a random one. This can also be seen comparing the figures 3 and 4. We observe that the curve from Figure 3 decreases more sharply than the one from Figure 4.

The algorithm performs better on irregular networks with a significant irregularity of the b-function values. This result was the reason for testing this algorithm on transit-stub random networks, which are closer to "real world" networks, miming the cluster-type networks encountered in the real world (Table 2). It is also the reason for the high performance of this algorithm on trees.

Table 2: The average broadcast time after $\log n$ iterations, for pure random graphs and transit-stub random graphs.

type	**vertices** n	**iterations** $\log n$	**average** b(G)
pure-random	100	7	10.10
transit-stub	100	7	12.06
transit-stub	600	10	18.77

Table 3: The average broadcast time after $\log n$ iterations, for hypercubes in dimensions 3 to 10.

dimension	**iterations**	**average b(G)**
3	3	3
4	4	4
5	5	5.1
6	6	6.7
7	7	8
8	8	9
9	9	10.4
10	10	11.5

Table 4: The average broadcast time after d iterations, for Knödel graphs on 2^d vertices and degree d.

dimension	iterations	average b(G)
3	3	3
4	4	4.1
5	5	5.9
6	6	7
7	7	8
8	8	9
9	9	10
10	10	11

Table 5: The average broadcast time after d iterations for shuffle-exchange graphs on 2^d vertices.

dimension	iterations	average b(G)
3	3	4.12
4	4	6.12
5	5	8.12
6	6	10.07
7	7	12.07
8	8	14.29
9	9	16.37
10	10	18.75

In Tables 2, 3, 4, and 5 we present the testing results for some regular networks, considering the average broadcast time as it is defined in (5), for the whole graph, after $\lceil \log_2 n \rceil$ iterations.

2.3 The b-function and the graph density

Some heuristics developed for finding an approximate solution for the minimum broadcast time problem have been tested on random graphs. Since most random graph generators have the number of vertices and the density of the graph as main parameters, we considered it opportune to study the relationship between the b-function and these two parameters. More precisely, considering a vertex v having the smallest broadcast time $b_{\min}$, and a vertex w having the greatest broadcast time $b_{\max}$, we give an upper bound for the difference $b_{\max} - b_{\min}$, in terms of graph density ρ and the number of vertices n. We show that this upper bound is tight by an additive factor of two.

Definition 12. *For an undirected graph $G = (V, E)$ on n vertices and m edges, the graph density ρ is defined by:*

$$\rho = \frac{|E|}{|E_{K_n}|} = \frac{2m}{n(n-1)} \tag{6}$$

where $|E_{K_n}|$ represents the number of edges in the complete graph on n vertices K_n.

Although the result of the following lemma seems to be somehow trivial, it turns out to be helpful to our study.

Lemma 1. *If there is an edge between vertices v and w in graph, then:*

$$|b(v) - b(w)| \leqslant 1$$

Proof. Without loss of generality, assume that $b(v) > b(w)$ and $b(v) - b(w) > 1$, which implies that $B_v(v, 1) \neq w$. Considering a new broadcast scheme B' in which v is calling w in the first time-slot, $B'(v, 1) = w$. After this call, we can follow the broadcast scheme having w as originator. We obtain for v a broadcast time $b'(v) = b(w) + 1 < b(v)$. This contradicts the definition of $b(v)$ as being the minimum broadcast time for v. □

This lemma has a straightforward corollary, which bounds $|b(v) - b(w)|$ for two arbitrary vertices v and w in graph, in terms of graph's diameter.

Corollary 1. *Given a graph $G(V, E)$ with diameter D, $|b(v) - b(w)| \leqslant D$, for any $v, w \in V$.*

Using this corollary we can now bound the range in which the minimum broadcast function can take values.

Theorem 2. *For any connected graph $G(V, E)$ on n vertices, having density ρ, the following inequality holds:*

$$|b(v) - b(w)| \leqslant \left\lfloor \frac{2n + 1 - \sqrt{4(\rho n(n-1) - 2n) + 17}}{2} \right\rfloor,$$

for any $v, w \in V$.

Proof. By Corollary 1, we have transformed the problem of upper bounding $b_{\max} - b_{\min}$ into the problem of upper bounding the diameter of the graph in terms of graph density and number of vertices. This problem has been considered in [220] and later in [47], in a slightly different context. Since we are looking for a closed formula for the maximum diameter of a graph in terms of the number of vertices and the graph density, we will use Theorem 3.1 from [220], which gives an upper bound for the number of edges of a graph, denoted by $m_{\max}$, given the diameter D, and the number of vertices n:

$$m_{\max} \leqslant D + \frac{1}{2}(n - D - 1)(n - D + 4) \tag{7}$$

In order to apply this result, we have to introduce *diameter critical* graphs: a graph G is called diameter critical if the addition of any edge decreases the diameter (Figure 4).

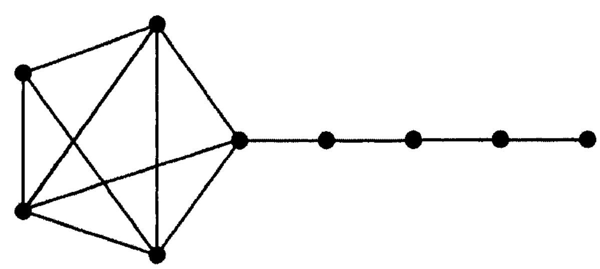

Figure 5: Diameter critical graph on 9 vertices with diameter 6.

Ore proved in [220] that the upper bound of (7) is attained for diameter critical graphs. Therefore, given a graph with n vertices and m edges, the maximum possible diameter, $D_{\max}$, satisfies the following inequalities:

$$m \leqslant D_{\max} + \frac{1}{2}(n - D_{\max} - 1)(n - D_{\max} + 4) \tag{8}$$

$$m > D_{\max} + 1 + \frac{1}{2}(n - D_{\max} - 2)(n - D_{\max} + 3) \tag{9}$$

Inequality (9) is obtained applying Ore's theorem for the case in which the diameter would be $D_{\max} + 1$. Solving the last two inequalities for $D_{\max}$ we get:

$$D_{\max} \in \left(\left\lceil \frac{2n - 1 - S}{2} \right\rceil, \left\lfloor \frac{2n + 1 - S}{2} \right\rfloor \right] \cup \left[\left\lceil \frac{2n - 1 + S}{2} \right\rceil, \left\lfloor \frac{2n + 1 + S}{2} \right\rfloor \right) \tag{10}$$

where $S = \sqrt{8(m-n)+17}$. Since $\left\lceil \frac{2n-1+\sqrt{8(m-n)+17}}{2} \right\rceil > n-1$ for any $m > n-1$, the maximum diameter must be in the interval:

$$D_{\max} \in \left(\left\lceil \frac{2n-1-S}{2} \right\rceil, \left\lfloor \frac{2n+1-S}{2} \right\rfloor \right] \tag{11}$$

We observe that the limits of the interval from (11) satisfy:

$$\left\lceil \frac{2n-1-S}{2} \right\rceil = \left\lfloor \frac{2n+1-S}{2} \right\rfloor \tag{12}$$

except in the case when $\sqrt{8(m-n)+17}$ is an odd integer. In this case:

$$\left\lceil \frac{2n-1-S}{2} \right\rceil = \left\lfloor \frac{2n+1-S}{2} \right\rfloor - 1 \tag{13}$$

Since the left side of the interval from (11) is opened, we obtain the following formula for the maximum diameter of a graph:

$$D_{\max} = \left\lfloor \frac{2n+1-\sqrt{8(m-n)+17}}{2} \right\rfloor \tag{14}$$

If we substitute $m = \rho n(n-1)/2$ in (14) we obtain the claimed bound. □

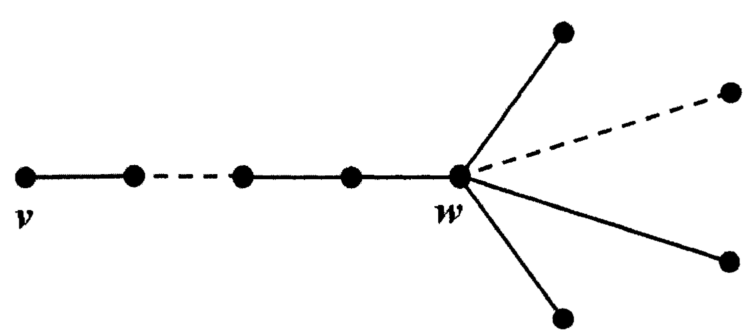

Figure 6: A path of length n_1 joined by an edge with a star on n_2 vertices

Observation

The upper bound of Theorem 2 is tight by an additive factor of 2. This can be proved by finding a graph for which $b_{\max} - b_{\min} = D - 2$. We can draw such a graph by joining the extremity of a path on n_1 vertices with the center of a star on n_2 vertices (Figure 6). It can be seen that if n_1 is smaller than the number of neighbors of v then $b(v) - b(w) = (n_1 + n_2 - 1) - n_2 = n_1 - 1 = D - 2$.

Chapter 3

The number of broadcast schemes in networks

3.1 Problem description

Throughout this chapter we model the interconnection network by a simple undirected connected graph $G = (V, E)$, in which the members of the network are the vertices of G and the communication lines are the edges of G. Many models of communication have been proposed and analyzed for broadcasting. For the purpose of this chapter we consider the *1-port*, *constant-time* model [117]. That is, once a member of the network knows the information, he can transmit it only to one neighbor per time-unit and the transmitting time is constant. At each time-unit, each member can be in one of the following three states: transmitting, receiving, or idle. Note that we allow an informed node to be idle, even though one of its neighbors is not informed. This approach may be seen as a suboptimal one, but we consider it more realistic since, for example, one processor could be busy at a certain moment of time. Also, in this way we cover the whole set of possible broadcast schemes. The same model has been used in [100] in order to analyse a random algorithm for broadcasting.

In a broadcast scheme, each informed vertex has to transmit the information to a subset of its neighbors in a certain order. Throughout this chapter we consider that two broadcast schemes are different if:

a) they have different originators, or

b) one vertex transmits to a different subset of neighbors, or

c) one vertex transmits to the same subset of neighbors but in different order.

In this chapter we give a tight lower bound, corresponding to a path, and a tight upper bound, corresponding to the complete graph, for the number of all possible broadcast schemes, under the above mentioned assumptions. We count them by mapping this set into a superset of the set of rooted spanning trees. We give the exact value for complete bipartite graphs and we establish an upper bound for regular graphs. Based on these results we describe an algorithm for enumerating all the broadcast schemes in networks. This work finds applications in a closely related problem: minimum broadcast graphs. This problem consists of finding a graph such that the broadcast time is $\lceil \log n \rceil$ and the number of edges is minimized. Minimum broadcast graphs are known only for some particular values of the number of vertices, mainly around $n = 2^k$ or small values of n. The smallest value for which the minimum broadcast graph is not known is $n = 23$. Such a small number suggests a brute force approach, which subsequently requires a knowledge of the dimension of the problem and an algorithm to enumerate all the broadcast schemes.

A random algorithm for broadcasting has been described and analyzed in [100]. As a byproduct of the method of counting the broadcast schemes, we describe and analyze a new random algorithm for broadcasting.

The chapter is organized as follows. In Section 3.2 we compute the lower and the upper bounds for arbitrary graphs. In Section 3.3 we count the number of broadcast schemes in complete bipartite graphs. Finally, in Section 3.4 we describe and analyze an algorithm for enumerating all the broadcast schemes along with a new random algorithm for broadcasting in networks. The definitions will be given in the places where they are used.

3.2 Bounding the number of the broadcast schemes

Given a simple connected undirected graph G, we denote by d_i the degree of a vertex i, by $ST(G)$ the set of spanning trees of G, by $S\overline{T}(G)$ the set of rooted spanning trees of G, by $SB(G)$ the set of all possible broadcast schemes in G, and by $|S|$ the cardinality of a set S.

The following lemma offers some helpful insights about the number of broadcast schemes in networks.

Lemma 2. *Given a tree T on n vertices, with the degree sequence $d_1, ..., d_n$, the* *number of all possible broadcast schemes in T is:*

$$|SB(T)| = 2(n-1)\prod_{j=1}^{n}(d_j - 1)! \tag{15}$$

Proof. Assume that vertex i is the originator. This vertex can choose to send the information among its d_i neighbors. Thus, the number of all possible calls for vertex i to its neighbours will be $d_i!$. Consider now one of its informed neighbors, say j. This one can choose among only $d_j - 1$ neighbors, giving a total of $(d_j - 1)!$ possibilities. Thus, the number of all possible broadcast schemes, having vertex i as originator, will be $d_i! \prod_{\substack{j=1 \\ j\neq i}}^{n} (d_j - 1)!$. Considering now all possible n originators

$$|SB(T)| = \sum_{i=1}^{n} d_i! \prod_{\substack{j=1 \\ j\neq i}}^{n} (d_j - 1)! = \sum_{i=1}^{n} d_i \prod_{j=1}^{n} (d_j - 1)! \tag{16}$$

Since the degree sum of all the vertices in a tree is $2n - 2$, the result follows. □

The following theorem gives the exact lower and upper bounds for the cardinality of $SB(G)$.

Theorem 3. *Given a graph G on n vertices, the following inequalities hold:*

$$2n - 2 \leq |SB(G)| \leq \frac{(2n-2)!}{(n-1)!} \tag{17}$$

Proof. Assuming a given broadcast scheme in G, we can construct a subgraph formed by all the vertices of G and by the edges that were used to transmit the information. This subgraph is a spanning tree in G because it contains all the vertices of G, it is connected, and it has no cycles (a vertex will never transmit to another already informed vertex). Furthermore, if we identify the originator, we can map the set of broadcast schemes into the set of rooted spanning trees of G. This is a many-to-one map since, as we saw in Lemma 2, there can be more broadcast schemes corresponding to one rooted tree, and every rooted spanning tree has at least one corresponding broadcast scheme.

From Lemma 2, we deduce that, for a graph G on n vertices, in order to minimize the cardinality of $SB(G)$ one must minimize both the degrees of the spanning trees and the number of spanning trees. We know that the path graph P_n on n vertices meets these requirements. Since P_n is a tree with the degree sequence $1, 2, ..., 2, 1$, applying the result from Lemma 2 we obtain the lower bound:

$$2n - 2 \leq |SB(G)| \tag{18}$$

For the upper bound we consider the complete graph on n vertices K_n, which maximizes both the degrees of vertices and the number of spanning trees. Let $T(n; d_1, ..., d_n)$ be a spanning tree in K_n with the degree sequence $d_1, ..., d_n$. The number of spanning trees with a given degree sequence in the complete graph has been computed in terms of generating functions first by Harary and Prins in [137] and then explicitly by Berge in [31]:

$$|ST(n; d_1, ..., d_n)| = \binom{n-2}{d_1 - 1, ..., d_n - 1} = \frac{(n-2)!}{\prod_{j=1}^{n} (d_j - 1)!} \tag{19}$$

Applying the result from Lemma 2, we obtain all the broadcast schemes corresponding to all the trees with a given degree sequence $d_1, ..., d_n$ in K_n

$$|SB(T_{d_1,...,d_n})| = \frac{(n-2)!}{\prod_{j=1}^{n} (d_j - 1)!} 2(n-1) \prod_{j=1}^{n} (d_j - 1)! = 2(n-1)! \tag{20}$$

We observe that the result does not depend on a specific degree sequence. Now we have to count how many possible different degree sequences realizing a tree are in K_n. This is given by the number of integer solutions of the equation:

$$d_1 + d_2 + ... + d_n = 2n - 2 \tag{21}$$

subject to constraints $1 \leq d_i \leq n - 1$, since the maximum possible degree is $n - 1$ and there are at least 2 vertices with degree 1 [31].

Since we will need this result later, we will solve a slightly more general problem: "Find the number of integer solutions of $x_1 + x_2 + ... + x_n = m$, subject to constraints $1 \leq x_i \leq k$ for all $1 \leq i \leq n$." The original problem is the particular case $m = 2n - 2$ and $k = n - 1$.

Let $f(m, k)$ be the number of solutions of our general problem, and define

$$G(z) = (z + z^2 + z^3 + ... + z^k)^n \tag{22}$$

The exponent of z in $G(z)$ is the sum of the exponents in the n terms. Thinking of the exponents as being the x_i's, in order to get z^m in the product the exponents

(x_i's) must sum to m. The number of ways we can get z^m in the product is exactly the number of solutions of $x_1 + x_2 + ... + x_n = m$. The constraints $1 \leq x_i \leq k$ are reflected in the fact that each of the integers $1, 2, ..., k$ is found exactly once in the exponents in $(z + z^2 + z^3 + \ldots + z^k)$. We note that:

$$G(z) = z^n(1-z)^{-n}(1-z^k)^n \tag{23}$$

Expanding the second term in an infinite series and using the binomial theorem we get:

$$G(z) = z^n \sum_{i=0}^{\infty} \binom{n+i-1}{i} z^i \sum_{j=0}^{n} (-1)^j \binom{n}{j} z^{jk} \tag{24}$$

where $\binom{n}{m}$ is the number of combinations of n objects taken m at a time. To get the z^m term in the result, we must have $n + i + jk = m$, i.e.

$$f(m,k) = \sum_{j=0}^{\lfloor (m-n)/k \rfloor} (-1)^j \binom{n+i-1}{i} \binom{n}{j} \tag{25}$$

where $n + i + jk = m$. We can eliminate i in the above equation by letting $i = m - n - kj$:

$$f(m,k) = \sum_{j=0}^{\lfloor (m-n)/k \rfloor} (-1)^j \binom{n}{j} \binom{m-kj-1}{m-n-kj}, \tag{26}$$

We can simplify this result by noting that $\binom{m-kj-1}{m-n-kj} = \binom{m-kj-1}{n-1}$. Therefore,

$$f(m,k) = \sum_{j=0}^{\lfloor (m-n)/k \rfloor} (-1)^j \binom{n}{j} \binom{m-kj-1}{n-1} \tag{27}$$

The original problem is just the case $f(2n-2, n-1)$. The number of solutions is then

$$f(2n-2, n-1) = \sum_{j=0}^{\lfloor (n-2)/(n-1) \rfloor} (-1)^j \binom{n}{j} \binom{2n-kj-3}{n-1} = \binom{2n-3}{n-1} \tag{28}$$

Thus, the total number of broadcast schemes in K_n will be

$$|SB(K_n)| = 2(n-1)! \frac{(2n-3)!}{(n-2)!(n-1)!} = \frac{(2n-2)!}{(n-1)!} \tag{29}$$

which is also the upper bound for any simple, connected, undirected graph. □

It is interesting to compare the number of broadcast schemes with the cardinality of the set of solutions of other NP-complete problems. For example, the ratio between the number of broadcast schemes in the complete graph to the number of Hamiltonian cycles in the same graph, denoted here by $|HC(K_n)|$, has an exponential order of magnitude:

$$\frac{|SB(K_n)|}{|HC(K_n)|} = \frac{(2n-2)!}{(n-1)!n!} \approx 2^{2n} \tag{30}$$

3.3 Results for particular topologies

In this section we give the exact value for the number of broadcast schemes in the complete bipartite graphs and we establish an upper bound for regular graphs.

Theorem 4. *The number of broadcast schemes of the complete bipartite graph $K_{p,q}$ is:*

$$|SB(K_{p,q})| = 2\frac{(p+q-1)!\,(p+q-2)!}{(p-1)!\,(q-1)!} \tag{31}$$

Proof. It can be shown by induction over the number of vertices that the number of spanning trees in $K_{p,q}$, with a given degree sequence $c_1, ..., c_p$, $d_1, ..., d_q$ is:

$$|ST(p+q; c_1, ..., c_p, d_1, ..., d_q)| = \frac{(p-1)!\,(q-1)!}{\prod_{i=1}^{p}(c_i-1)!\prod_{j=1}^{q}(d_j-1)!} \tag{32}$$

where $1 \le c_i \le q$, $1 \le i \le p$, $1 \le d_j \le p$, and $1 \le j \le q$.

We present here a proof of (32) due to Igor Pak [222]. For $n = 2$ the statement is true: $|ST(2; 1, 1)| = 1$. Assume that the statement is true for all $n < p + q$. Assume now that $n = p + q$. Consider $T_{p,q}$ a spanning tree in $K_{p,q}$. Denote the vertices $l_1, ..., l_p$ with degrees $c_1, ..., c_p$ the "left" partition and the vertices $r_1, ..., r_p$ with degrees $d_1, ..., d_q$ the "right" partition of $T_{p,q}$. Assuming that the degrees are ordered in descending order, we delete one vertex with degree one (there must be one) in $T_{p,q}$, say vertex l_p. We obtain a spanning tree with degree sequence $c_1, ..., c_{p-1}, d_1, ..., d_k - 1, ..., d_q$, if the vertex was adjacent to the k^{th} vertex from the right side in that particular spanning tree. Considering now all the edges between l_p and the "right" partition (q edges) and using the fact that the number of spanning trees in $K_{p-1,q}$ is known by the induction hypothesis, the number of spanning trees in $K_{p,q}$ with degree sequence $c_1, ..., c_p, d_1, ..., d_q$ will be:

$$
\begin{aligned}
&|ST\,(p+q; c_1, ..., c_p, d_1, ..., d_q)| = \\
&= \sum_{k=1}^{q} \frac{(p-2)!\,(q-1)!}{(c_1-1)!...\,(c_{p-1}-1)!\,(d_1-1)!...\,(d_k-2)!...\,(d_q-1)!} = \\
&= \frac{(p-2)!\,(q-1)!\,(d_1-1+...+d_q-1)}{(c_1-1)!...\,(c_{p-1}-1)!\,(d_1-1)!...\,(d_k-1)!...\,(d_q-1)!} = \\
&= \frac{(p-2)!\,(q-1)!\,(d_1+...+d_q-q)}{(c_1-1)!...\,(c_{p-1}-1)!\,(d_1-1)!...\,(d_k-1)!...\,(d_q-1)!}
\end{aligned} \tag{33}
$$

Since $K_{p,q}$ is bipartite

$$c_1 + ... + c_p = d_1 + ... + d_q \tag{34}$$

On the other hand

$$c_1 + ... + c_p + d_1 + ... + d_q = 2(p+q-1) \tag{35}$$

Hence,

$$
\begin{aligned}
&|ST\,(p+q; c_1, ..., c_p, d_1, ..., d_q)| = \\
&= \frac{(p-2)!\,(q-1)!\,(p+q-1-q)}{(c_1-1)!...\,(c_{p-1}-1)!\,(d_1-1)!...\,(d_q-1)!} = \\
&= \frac{(p-1)!\,(q-1)!}{(c_1-1)!...\,(c_{p-1}-1)!\,(d_1-1)!...\,(d_q-1)!}
\end{aligned} \tag{36}
$$

Applying Lemma 2, we obtain the number of broadcast schemes for the above set of trees:

$$|SB\,(p+q; c_1, ..., c_p, d_1, ..., d_q)| = 2\,(p+q-1)\,(p-1)!\,(q-1)! \tag{37}$$

The number of different degree sequences in $K_{p,q}$ is the number of integer solutions of the equation:

$$c_1 + ... + c_p + d_1 + ... + d_q = 2\,(p+q) - 2 \tag{38}$$

subject to constrains:

$$
\begin{aligned}
&1 \le c_i \le q, \text{ with } 1 \le i \le p, \text{ and} \\
&1 \le d_j \le p, \text{ with } 1 \le j \le q.
\end{aligned} \tag{39}
$$

We denote by N_c the number of integer solutions of the equation

$$c_1 + ... + c_p = p + q - 1 \tag{40}$$

and by N_d the number of integer solutions of the equation

$$d_1 + ... + d_q = p + q - 1 \tag{41}$$

subject to constrains (39). According to (27):

$$N_c = \binom{p+q-2}{p-1} \text{ and } N_d = \binom{p+q-2}{q-1} \tag{42}$$

Hence, the number of solutions of (38) will be:

$$N_c N_d = \left(\frac{(p+q-2)!}{(p-1)!\,(q-1)!} \right)^2 \tag{43}$$

Combining (43) and (37) we obtain the number of broadcast schemes in the complete bipartite graph $K_{p,q}$:

$$\begin{aligned} |SB\,(K_{p,q})| &= 2\,(p+q-1)\,(p-1)!\,(q-1)! \left(\frac{(p+q-2)!}{(p-1)!\,(q-1)!} \right)^2 = \\ &= 2\frac{(p+q-1)!\,(p+q-2)!}{(p-1)!\,(q-1)!} \end{aligned} \tag{44}$$

□

Consider now a star tree on $p+1$ vertices, formed by one vertex connected by p edges to p vertices. According to Lemma 2, the number of broadcast schemes in a star tree on $p+1$ vertices is $2p!$. On the other hand, using (44) we obtain

$$|SB\,(K_{p,1})| = 2\frac{(p+1-1)!\,(p+1-2)!}{(p-1)!\,(1-1)!} = 2p! \tag{45}$$

Also, in a cycle on n vertices C_n there are n spanning trees, with degree sequence $1, 2, ..., 2, 1$. For each spanning tree we have $2(n-1)$ broadcast schemes, according to Lemma 2. Therefore, the number of broadcast schemes in a cycle on n vertices C_n is $|SB\,(C_n)| = 2n\,(n-1)$. On the other hand, $K_{2,2}$ is the cycle on 4 vertices C_4. Applying (44) we obtain: $|SB\,(K_{2,2})| = 2\frac{(2+2-1)!(2+2-2)!}{(2-1)!(2-1)!} = 4!$.

Theorem 5. *The number of broadcast schemes for any regular graph on n vertices and degree k, denoted by $R_{n,k}$, is upper bounded by:*

$$|SB\,(R_{n,k})| \leq 2k\,((k-1)!)^{\left\lceil \frac{n-1}{k-1} \right\rceil} \left(\frac{nk}{n-1} \right)^{n-2} \tag{46}$$

Proof. Let us consider a tree T on n vertices, with an ascending order sorted degree sequence $d_1, \ldots, d_i, \ldots, d_j, \ldots, d_n$. Without loss of generality, assume that there exist i and j, $i < j$, such that $2 \leq d_i \leq d_j$. By Lemma 2, the number of broadcast schemes in T can be expressed as:

$$|SB(T)| = c(d_i - 1)!(d_j - 1)! \tag{47}$$

where c is where c is the remaining product of factorials.

Let us consider now a new tree T' on n vertices with the degree sequence

$$d_1, ..., d_i', ..., d_j', ..., d_n. \tag{48}$$

That is, only two vertices change their degree: vertex i has degree $d_i' = d_i - 1$ and vertex j has degree $d_j' = d_j + 1$. The number of broadcast schemes in T' can be expressed as

$$|SB(T')| = c(d_i - 2)!d_j! \tag{49}$$

We observe that

$$|SB(T)| < |SB(T')| \tag{50}$$

Thus, by increasing the bigger degrees in the detriment of the smaller degrees we obtain a tree with a bigger number of broadcast schemes. Since the degrees in regular graphs $R_{n,k}$ are upper bounded by k, we conclude that the degree distribution which yields maximum number of broadcast schemes in such a tree is

$$\underbrace{1, 1, ..., 1}_{n-p-1}, k', \underbrace{k, k, ..., k}_{p} \tag{51}$$

where $1 \leq k' \leq k$ and $p \leq \left\lceil \frac{n-1}{k-1} \right\rceil$. It is clear that, in general, such a spanning tree might not exist in a regular graph $R_{n,k}$, but it will upper bound the number of broadcast schemes of any other spanning tree in $R_{n,k}$ by:

$$\begin{aligned} \left|SB\left(T_{\text{spanning tree in } R_{n,k}}\right)\right| &\leq 2(n-1) \prod_{j=1}^{\left\lceil \frac{n-1}{k-1} \right\rceil} (k-1)! = \\ &= 2(n-1)\left((k-1)!\right)^{\left\lceil \frac{n-1}{k-1} \right\rceil} \end{aligned} \tag{52}$$

On the other hand, Biggs [40] gives a tight upper bound for the number of spanning trees of any regular graph on n vertices and degree k:

$$|ST(R_{n,k})| \leq \frac{1}{n}\left(\frac{nk}{n-1}\right)^{n-1} \tag{53}$$

Combining this result with (52) we obtain the claimed upper bound. Note that the claimed bound is not tight, even though the bound provided by Biggs is tight for complete graphs. □

The observation from (50), which states that increasing the bigger degrees in the detriments of the smaller ones increases the number of broadcast schemes, yields the following corollary:

Corollary 2. *The maximum number of broadcast schemes of a tree on n vertices is*

$$|ST(T_n)| = 2(n-1)! \tag{54}$$

and it is obtained for the star tree $K_{n-1,1}$.

3.4 Applications

3.4.1 An algorithm for enumerating all the broadcast schemes

A first straightforward application of the counting method described in Section 3.2 is an algorithm for enumerating all possible broadcast schemes in arbitrary graphs.

Recall that there is a many-to-one correspondence from the set of all possible broadcast schemes to the set of the rooted spanning trees. Also, there is a many-to-one correspondence from the set of the rooted spanning trees to the set of the spanning trees in graph. Moreover, Lemma 2 shows a direct connection between a given rooted spanning tree and all possible broadcast schemes in that tree. Consider now the rooted trees as directed trees, where the orientation of the edges is from the root to the leaves. All the possible broadcast schemes in such a tree can be obtained by generating all the possible permutations of the outgoing edges for each node. That is, we can relabel all outgoing edges according to each permutation, for each vertex.

In order to obtain all the rooted trees, we can run an algorithm for enumerating all the spanning trees ([178] for example), and, for each such tree, we consider one root at a time. The complexity of enumerating all the rooted spanning trees is driven by their number [178], and the edge labelling of a spanning tree takes $O(n)$ time, where n is the number of vertices in graph. According to Lemma 2 and Theorem 3, the overall complexity is $O(nN)$, where N is the number of all possible broadcast schemes.

3.4.2 A random algorithm for broadcasting

The only known random algorithm for broadcasting has been described and analyzed in [100]. It is based on a natural idea: at each time unit, every informed vertex will randomly choose to inform one of its neighbors.

Our algorithm has two phases:

a) Randomly choose a spanning tree.

b) On this spanning tree run the algorithm of Slater, Cockayne and Hedetniemi [244] which gives in $\Theta(n)$ time an optimal broadcast scheme for this tree.

There are two main issues in comparing the two random algorithms for broadcasting. The first one concerns the distributed character of algorithm in [100]. This issue can be overcome by transforming our algorithm into a distributed one as follows: chose a random spanning tree as in [7] and find the broadcast time of this tree as in [15].

The second issue is related to the probability of choosing an optimal broadcast scheme. Since it is still an open question if the algorithm of [100] is equivalent to choosing a broadcast scheme uniformly at random from the set of all possible broadcast schemes, we cannot compare the two algorithms using this criterion. Nevertheless, we encounter two extreme cases: a tree and a complete graph. For any tree with more than two vertices, our algorithm chooses an optimal broadcast scheme with probability one, while the algorithm from [100] yields a probability strictly less than one. In order to consider the complete graph case we use the following proposition.

Proposition 1. *The number of optimal broadcast schemes in the complete graph K_n on $n = 2^k$ vertices is $|SB_{opt}(K_n)| = n!$.*

Proof. Since the number of vertices is $n = 2^k$, the minimum broadcast time will be k [95]. That is, at each call, we can at most double the number of informed vertices. Therefore, the spanning tree yielded by this broadcast scheme is a binomial tree and it is easy to show that there is no other tree structure which would have this broadcast time in K_n [249].

We have n possibilities to choose the root, $(n-1)(n-2)$ possibilities to choose the next two vertices to inform, $(n-3)(n-4)(n-5)(n-6)$ possibilities to choose the next four vertices to inform, and so on. Due to the completeness of the graph, we have $n!$ such broadcast schemes. Note that this number also represents the number of binomial trees embedded in the complete graph on $n = 2^k$ vertices. $\square$

Since the number of spanning trees in K_{2^k} is $2^{k(2^k-2)}$ [49], we can estimate now the probability of choosing a tree which yields an optimal broadcast scheme in K_{2^k} using our algorithm:

$$p\left(K_{2^k}\right) = \frac{2^k!}{2^{k(2^k-2)}} \tag{55}$$

Although this result applies only to complete graphs on 2^k vertices, it gives us an idea of how well the algorithm performs.

From the point of view of time complexity, the worst case of the algorithm from [100] is upper bounded by the longest path in graph and the expected cover time is $O(\log n)$ for almost all graphs [100]. For our algorithm, the time complexity of randomly choosing a spanning tree and then running a $O(n)$ deterministic algorithm on it is clearly driven by the complexity of the first step. There are many algorithms in literature for generating uniformly a random spanning tree (for example, see [63] or [43]). To our knowledge, the fastest one is described in [252]. The time complexity of most of these algorithms is in $O(n^3)$ and has an expected time in $O(n \log n)$.

Chapter 4

The spectra of Knödel graphs

Knödel graphs $W_{d,n}$, are regular graphs on an even number of vertices n and degree d. They have been introduced by W. Knödel [191] and have been proven to be minimum gossip graphs and minimum broadcast graphs for degree $d = \lfloor \log_2 n \rfloor$.

Knödel graphs $W_{g,n}$ are defined as undirected graphs $G(V, E)$, with $|V| = n$ even, and the set of edges:

$$E = \left\{(i,j) \mid i + j = 2^k - 1 \bmod n \right\} \tag{56}$$

where $k = 1, 2, ..., g,\ 0 \leq i, j \leq n-1,$ and $1 \leq g \leq \lfloor \log_2 n \rfloor$. [108]

Knödel graphs $W_{g,n}$ can be also defined as Cayley graphs [166]. For any even n and $1 \leq d \leq \lfloor \log_2 n \rfloor$, $W_{g,n}$ is a Cayley graph on the semi-direct product $G = Z_2 \times Z_{n/2}$ with the set of generators $S = \{ (1, 2^i),\ 0 \leq i \leq g-1\}$, and the multiplicative law $(x, y)(x', y') = (x + x', y + (-1)^x y')$, where $x, x' \in Z_2$ and $y, y' \in Z_{n/2}$. Fertin and Raspaud wrote a comprehensive survey on Knödel graphs in [108].

Recently, it has been proven that Knödel graphs $W_{d,2^d}$ on 2^d vertices and degree d have diameter $\lceil d/2 + 1 \rceil$, which is the smallest known diameter among all minimum broadcast graphs on 2^d vertices [108]. We believe that this is also a lower bound on diameter for all regular graphs on 2^d vertices and degree d.

A major drawback of the Knödel graphs $W_{d,2^d}$, compared to their counterparts, hypercubes and recursive circulants, was the lack of a logarithmic routing algorithm. In [151] we present such an algorithm which guarantees a path of length at most double than the length of an actual minimum path.

In [108] (Proposition 9) it is shown that $W_{d,2^d}$ is bipancyclic. That is, it holds cycles of any even length $4 \leq 2m \leq 2^d$. This property is particulary important for a parallel computer architecture, since it guarantees a natural embedding of even length arrays among the processors.

While hypercubes and recursive circulants can be defined only for a number of nodes power of two, Knödel graphs can be defined for any even number of vertices. Furthermore, some of them present interesting properties in terms of broadcasting and gossiping (see Table 6), under different communication models. Of particular interest is $W_{d-1,2^d-2}$, which forms another infinite family of minimum broadcast graphs [186]. Knödel graphs have also been used in [144] to obtain broadcast graphs on number of vertices $n \neq 2^d$ and $n \neq 2^d - 2$.

Table 6: Different Knödel graphs and their communication properties [108].

Type of Knödel graph	**Properties**
$W_{k,2^k}$	Minimum broadcast graph [99] Minimum gossip graph [191] Minimum linear gossip graph [118]
$W_{k-1,2^k-2}$	Minimum broadcast graph [186] Minimum gossip graph [196] Minimum linear gossip graph [118]
$W_{k-1,2^k-4}$	Minimum gossip graph [196] Minimum linear gossip graph [118]
$W_{k-1,2^k-6}$	Minimum linear gossip graph [118]
$W_{k-2,n}$ $2^{k-1}+2 \leq n \leq 3 \cdot 2^{k-2} - 4$	Broadcast graph [103] Linear gossip graph [104] Gossip graph [103]
$W_{k-1,n}$ $3 \cdot 2^{k-2} - 4 \leq n \leq 2^k - 4$	Broadcast graph [103] Linear gossip graph [104] Gossip graph [103]

Knowing the spectrum of a graph has a great impact on other characteristics of the graph. For example, the complexity of a graph is $\kappa(G) = \frac{1}{n}\prod_{k=1}^{n-1}(\lambda_n - \lambda_k)$, where n is the number of eigenvalues, and λ_n is the greatest eigenvalue [40]. In [138], Harary and Schwenk describe a spectral approach to determine the number of walks in a graph. In [13] and [86], a connection between the spectrum of a graph and the discrete Green's functions of graphs is described. Diaconis in [77] and Lovász in [210] show strong relationships between the characteristics of the random walks in graph (hitting time, commute time, cover time, and mixing time) and the spectrum of a graph.

The goal of this study is to efficiently compute the spectra of Knödel graphs, first for $W_{d,2^d}$, and then for arbitrary degree g and number of vertices n. We use results from Fourier analysis, circulant matrices and PD-matrices. Based on these results we give a $O(ng)$ formula for the number of spanning trees in Knödel graphs.

4.1 Definitions and notations

We denote the adjacency matrix of an undirected simple graph by $A = [a_{ij}]$, where $1 \leq i, j \leq |V| = n$, with $a_{ij} = 1$ whenever vertex i is adjacent to vertex j, and 0 otherwise. The set of eigenvalues of A, together with their multiplicities, is said to be *the spectrum* of G. If we denote the identity matrix by I, then the *characteristic polynomial* of G is defined as $P(\lambda) = \det|\lambda I - A|$. The spectrum of G will be the set of solutions of the equation $P(\lambda) = 0$.

As of this writing, spectra are known for some particular graphs: path, cycle, complete graph, complete bipartite graph, complete tree, hypercube, k-dimensional lattice, star graph, etc. (see [71] and [129] for further references).

If M is a matrix, we denote the transpose of M by M^T, the complex conjugate of M by $\overline{M}$, the transpose complex conjugate of M by M^*, and the inverse of M by M^{-1}. We denote a permutation by π:

$$\pi = \begin{pmatrix} 1 & 2 & \ldots & n \\ \sigma(1) & \sigma(2) & \ldots & \sigma(n) \end{pmatrix}, \tag{57}$$

and the corresponding permutation matrix of π by $P(\pi) = [a_{ij}]$, where $a_{i,\sigma(i)} = 1$ and $a_{i,j \neq \sigma(i)} = 0$.

If $z \in \mathbb{C}$, we denote the complex conjugate of z by $\overline{z}$, and the norm of z by $\|z\| = \sqrt{z\overline{z}}$.

We denote the diagonal matrix with the elements of the main diagonal $(\lambda_1, \lambda_2, ..., \lambda_n)$ by $diag(\lambda_1, \lambda_2, ..., \lambda_n)$.

We denote a circulant matrix with the first row $(a_1, a_2, ..., a_n)$ by $circ(a_1, a_2, ..., a_n)$. That is, the rest of the rows will be circular permutations of the first row toward right. Thus, $a_{i,j} = a_{1,\, i-j+1 \bmod n}$. If the step of the shift is an integer $q \neq 1$, we call this matrix a (q)*circulant* matrix [250].

We denote the *inverse permutation* matrix by Γ, which is a $(-1)circulant$, that is, $\Gamma = (-1)circ(1, 0, ..., 0)$. An important property of Γ is that $\Gamma^2 = I$, where I is the identity matrix.

We use F to denote the *Fourier* matrix, defined by its conjugate transpose

$$F^* = \frac{1}{\sqrt{n}} \left[w^{(i-1)(j-1)} \right], 1 \leq i, j \leq n,$$

where w stands for the n^{th} root of the unity [75]. Two important properties of F are: $F^* = \overline{F}$ and $FF^* = I$.

Other definitions and notations will follow in the places where they are used.

4.2 General graph theory considerations

We observe that the adjacency matrix of the Knödel graphs is a (-1)*circulant* matrix, called also a *retrocirculant* [5], where all the rows are circular permutations of the first row toward left. For example, the adjacency matrix of $W_{3,2^3}$ is:

$$A_{W_{3,2^3}} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad (58)$$

Some general remarks can be made about the spectra of $W_{g,n}$:

- All eigenvalues are real since the adjacency matrix is real and symmetric [50].
- The maximum eigenvalue is $\lambda_n = g$, since $W_{g,n}$ is regular of degree g [40].
- All eigenvalues are symmetric with respect to zero [236] since the Knödel graph is bipartite; also, its characteristic polynomial has the form:

$$P(\lambda) = \lambda^n + a_2\lambda^{n-2} + ... + a_{n-2}\lambda^2 + a_n \qquad (59)$$

- In particular, for $W_{d,2^d}$, the number of distinct eigenvalues is at least $\lceil \frac{d}{2} \rceil + 2$ since the diameter is $\lceil \frac{d}{2} \rceil + 1$ [71].

4.3 Computing the spectrum of $W_{d,2^d}$

Davis [75] showed that a matrix A is (-1)*circulant* if and only if $A = F^*(\Gamma\Lambda)F$, where $\Lambda = diag(\gamma_1, \gamma_2, ..., \gamma_n)$. This is equivalent to $FAF^* = \Gamma\Lambda$, which means that A and $\Gamma\Lambda$ have the same eigenvalues [75]. The second term is a *PD-matrix*, defined as a product of two matrices, P and D, where P is a permutation matrix and D is a diagonal matrix. The characteristic polynomial of a PD-matrix can be computed by decomposing the permutation P in prime cycles of total length n [75]. Since Knödel graphs adjacency matrices are (-1)*circulants*, the problem reduces to that of determining the values of $\gamma_1, \gamma_2, ..., \gamma_n$. Since $\Gamma\Lambda$ has the form:

$$\Gamma\Lambda = \begin{pmatrix} \gamma_1 & 0 & \dots & 0 \\ 0 & 0 & \dots & \gamma_n \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \gamma_2 & \dots & 0 \end{pmatrix}, \tag{60}$$

we can compute $FAF^* = [c_{ij}] = \Gamma\Lambda$ and identify the terms $c_{11} = \gamma_1$, $c_{2n} = \gamma_n$, $\dots$, $c_{n2} = \gamma_2$. In order to compute the triple matrix multiplication FAF^*, we note that:

$$F = \overline{F}^* = \frac{1}{\sqrt{n}}\left[w^{-(i-1)(j-1) \bmod n}\right] \tag{61}$$

Since $w^n = 1$ we may skip the modulo operations in the powers. Also, in order to avoid confusion with the summation indices, we emphasize the matrix indices. That is, $[a]_{i,j}$ means that i is the row index and j is the column index, both varying from 1 to n.

$$\begin{aligned} FAF^* =& \frac{1}{n}\left[w^{-(i-1)(k-1)}\right]_{i,k}\left[a_{k,m}\right]_{k,m}\left[w^{(m-1)(j-1)}\right]_{m,j} = \\ & \frac{1}{n}\left[\sum_{k=1}^{n} w^{-(i-1)(k-1)}a_{k,m}\right]_{i,m}\left[w^{(m-1)(j-1)}\right]_{m,j} = \\ & \frac{1}{n}\left[\sum_{k=1}^{n} w^{-(i-1)(k-1)}a_{1,m+k-1}\right]_{i,m}\left[w^{(m-1)(j-1)}\right]_{m,j} \end{aligned} \tag{62}$$

Since in the first row of the adjacency matrix only d elements are nonzero, we can change the variable of summation in the first term of (62): $k \rightarrow r$, where $1 \leq r \leq d$.

Therefore:

$$\begin{aligned}FAF^* =& \frac{1}{n}\left[\sum_{r=1}^{d} w^{-(i-1)(2^r-m)}\right]_{i,m}\left[w^{(m-1)(j-1)}\right]_{m,j} = \\ & \frac{1}{n}\left[\sum_{m=1}^{n}\left(\sum_{r=1}^{d} w^{-(i-1)(2^r-m)}\right) w^{(m-1)(j-1)}\right]_{i,j} = \\ & \frac{1}{n}\left[\sum_{m=1}^{n}\sum_{r=1}^{d} w^{-(i-1)(2^r-m)} w^{(m-1)(j-1)}\right]_{i,j} = \\ & \frac{1}{n}\left[w^{-(j-1)}\sum_{m=1}^{n} w^{m(j+i-2)}\sum_{r=1}^{d} w^{-2^r(i-1)}\right]_{i,j}\end{aligned} \tag{63}$$

Thus, for the general term of FAF^* we obtain:

$$c_{i,j} = \frac{w^{-(j-1)}}{n}\sum_{m=1}^{n} w^{m(j+i-2)}\sum_{r=1}^{d} w^{-2^r(i-1)} \tag{64}$$

The general term of the $\Gamma\Lambda$ matrix from (60) can be expressed as follows:

$$\gamma_p = c_{n-p+2,p} = \frac{1}{n}\left(w^{-(p-1)}\sum_{m=1}^{n} w^{mn}\sum_{r=1}^{d} w^{-2^r(n-(p-1))}\right) \tag{65}$$

But $\sum\limits_{m=1}^{n} w^{mn} = n$ and $w^{-2^r(n-(p-1))} = w^{2^r(p-1)}$. Thus,

$$\gamma_p = w^{-(p-1)}\sum_{r=1}^{d} w^{2^r(p-1)} \tag{66}$$

On the other hand, the Γ matrix corresponds to the permutation:

$$\pi(\Gamma) = \begin{pmatrix} 1 & 2 & 3 & \dots & n/2+1 & \dots & n \\ 1 & n & n-1 & \dots & n/2+1 & \dots & 2 \end{pmatrix}$$

This permutation can be decomposed in n/2+1 prime cycles of total length n [75, 5]: $(1)(2,n)\dots(n/2,n/2+2)(n/2+1)$. Thus, the characteristic polynomial will be:

$$P(\lambda) = (\lambda-\gamma_1)\left(\lambda^2-\gamma_2\gamma_n\right)\left(\lambda^2-\gamma_3\gamma_{n-1}\right)\dots\left(\lambda^2-\gamma_{n/2}\gamma_{n/2+2}\right)\left(\lambda-\gamma_{n/2+1}\right) \tag{67}$$

The eigenvalues set will be:

$$S = \left\{\gamma_1, \sqrt{\gamma_2\gamma_n}, \sqrt{\gamma_3\gamma_{n-1}}, \dots, \sqrt{\gamma_{n/2}\gamma_{n/2+2}}, \gamma_{n/2+1}\right\} \tag{68}$$

As expected, since $W_{d,2^d}$ is regular of degree d, for the first eigenvalue we obtain:

$$\gamma_1 = \sum_{r=1}^{d} 1 = d \tag{69}$$

Aitken proved in [5] that, for a (-1)*circulant*, $\gamma_{n/2+1} = a_1 - a_2 + a_3 - ... - a_n$, where $(a_1, a_2, ..., a_n)$ are the values of the first row of adjacency matrix. Thus:

$$\gamma_{n/2+1} = \sum_{i=1}^{n} (-1)^{i+1} a_i = \sum_{j=1}^{d} (-1)^{2^j+1} = -d \tag{70}$$

For the rest of the eigenvalues we have to evaluate products of the form: $\gamma_t \gamma_{n-t+2}$, $2 \leq t \leq n/2$. From (66) we have:

$$\begin{aligned}\gamma_t \gamma_{n-t+2} = \left(w^{-(t-1)} \sum_{r=1}^{d} w^{2^r(t-1)} \right) \left(w^{-(n-t+1)} \sum_{r=1}^{d} w^{2^r(n-t+1)} \right) = \\ \left(w^{-(t-1)} \sum_{r=1}^{d} w^{2^r(t-1)} \right) \left(w^{(t-1)} \sum_{r=1}^{d} w^{2^r(n-t+1)} \right) = \\ \sum_{r=1}^{d} w^{2^r(t-1)} \sum_{r=1}^{d} \overline{w}^{2^r(t-1)} = \sum_{r=1}^{d} w^{2^r(t-1)} \overline{\sum_{r=1}^{d} w^{2^r(t-1)}} = \\ \left\| \sum_{r=1}^{d} w^{2^r(t-1)} \right\|^2\end{aligned} \tag{71}$$

This confirms the well-known fact that all eigenvalues are real. Thus, the spectrum of $W_{d,2^d}$ is the set:

$$S\left(W_{d,2^d}\right) = \{\pm d\} \cup \left\{ \pm \left\| \sum_{r=1}^{d} w^{2^r(t-1)} \right\| \right\} \tag{72}$$

where $2 \leq t \leq n/2$.

4.4 Observations

A. Not all eigenvalues are distinct. We can show that at most $(n-4)/2$ of them are distinct. If we decompose the norm from (72) in its trigonometric form we obtain:

$$\left\| \sum_{r=1}^{d} w^{2^r(t-1)} \right\|^2 = \left(\sum_{r=1}^{d} \cos \frac{2\pi}{2^d} 2^r (t-1) \right)^2 + \left(\sum_{r=1}^{d} \sin \frac{2\pi}{2^d} 2^r (t-1) \right)^2 \tag{73}$$

We observe that this norm evaluates to the same value for $t = n/4 + 1 - k$, and $t = n/4 + 1 + k$:

$$
\begin{gathered}
\left\|\sum_{r=1}^{d} w^{2^r(n/4+1-k-1)}\right\|^2 = \\
\left(\sum_{r=1}^{d} \cos\frac{2\pi}{2^d}2^r\left(\frac{2^d}{4}-k\right)\right)^2 + \left(\sum_{r=1}^{d} \sin\frac{2\pi}{2^d}2^r\left(\frac{2^d}{4}-k\right)\right)^2 = \\
\left(\sum_{r=1}^{d} \cos\frac{2\pi}{2^d}2^r\left(\frac{2^d}{4}+k\right)\right)^2 + \left(\sum_{r=1}^{d} \sin\frac{2\pi}{2^d}2^r\left(\frac{2^d}{4}+k\right)\right)^2 = \\
\left\|\sum_{r=1}^{d} w^{2^r(n/4+1+k-1)}\right\|^2
\end{gathered}
\tag{74}
$$

The computations for particular cases yield the claim that these are the only overlapping eigenvalues.

B. To our knowledge, there is no closed form for the sum from (71). Nevertheless, computations for particular cases suggest that for the particular value $t = 2^d/4 + 1$, the sum can be evaluated to a closed form:

$$
\begin{gathered}
\left\|\sum_{r=1}^{d} w^{2^r(2^d/4)}\right\|^2 = \left(\sum_{r=1}^{d} \cos\frac{\pi}{2}2^r\right)^2 + \left(\sum_{r=1}^{d} \sin\frac{\pi}{2}2^r\right)^2 = \\
\left(-1+1+\sum_{r=3}^{d} \cos\frac{\pi}{2}2^r\right)^2 + \left(0+\sum_{r=2}^{d} \sin\frac{\pi}{2}2^r\right)^2 = (d-2)^2
\end{gathered}
\tag{75}
$$

Thus, for $W_{d,2^d}$, the spectrum from (72) can be written as:

$$
S_{W_{d,2^d}} = \{d, (d-2)\} \cup \left\{\left\|\sum_{r=1}^{d} w^{2^r(t-1)}\right\|\right\} \tag{76}
$$

where $2 \leq t \leq n/4$, and the last set has multiplicity two.

C. Note that in formulas (62)–(71) we didn't make any assumptions regarding the number of vertices n nor the degree d. Therefore, the result from (72) can be extended in a similar manner for Knödel graphs with arbitrary degree g and number of vertices n, $W_{g,n}$:

$$
S_{W_{g,n}} = \{g\} \cup \left\{\left\|\sum_{r=1}^{g} w^{2^r(t-1)}\right\|\right\} \tag{77}
$$

where $2 \leq t \leq n/2$.

For example, for $W_{2,2^k}$, which are cycles C_{2^k} of length 2^k, applying (77), we obtain the spectrum:

$$S_{C_{2^k}} = \{\pm k\} \cup \left\{\pm \left\|w^{2(t-1)} + w^{4(t-1)}\right\|\right\} \tag{78}$$

where $2 \leq t \leq 2^{k-1}$. The norm from (78) can be evaluated to $2\cos 2\pi(t-1)/2^k$ as follows:

$$\begin{aligned}
&\left\|w^{2(t-1)} + w^{4(t-1)}\right\|^2 = \\
&\left(\cos\frac{2\pi}{2^k}2\,(t-1) + \cos\frac{2\pi}{2^k}4\,(t-1)\right)^2 + \\
&\left(\sin\frac{2\pi}{2^k}2\,(t-1) + \sin\frac{2\pi}{2^k}4\,(t-1)\right)^2 = \\
&4\left(\cos\frac{2\pi}{2^k}\,(t-1)\right)^2
\end{aligned} \tag{79}$$

Thus, we meet the well-known result [40] that the spectrum of a cycle of length n is the set:

$$S_{C_n} = \left\{ 2\cos\frac{2\pi j}{n} \,\middle|\, 1 \leq j \leq n \right\} \tag{80}$$

4.5 The number of spanning trees in Knödel graphs

An immediate consequence of the knowledge of the spectra of Knödel graphs $W_{g,n}$ is a $O(ng)$ formula for the number of spanning trees. It is well known that, given a regular graph G on n vertices and degree k, the number of spanning trees can be expressed as:

$$\kappa(G) = \frac{1}{n}\prod_{t=1}^{p-1}(k-\lambda_t)^{m_t}, \tag{81}$$

where λ_t are the eigenvalues, m_t their multiplicities, and p the number of distinct eigenvalues [40]. Thus, for the particular case in which the degree is d and the number of vertices is 2^d, using (76) we obtain:

$$\kappa\left(W_{d,2^d}\right) = \frac{d(2d-2)}{2^{d-2}}\prod_{t=2}^{2^{d-2}}\left(d^2 - \left\|\sum_{r=1}^{d} w^{2^r(t-1)}\right\|^2\right)^2 \tag{82}$$

If we further decompose the norm from (82) in its trigonometric form, we obtain:

$$\left\|\sum_{r=1}^{d} w^{2^r(t-1)}\right\|^2 =$$
$$\left(\sum_{r=1}^{d}\cos\frac{2\pi}{2^d}2^r(t-1)\right)^2 + \left(\sum_{r=1}^{d}\sin\frac{2\pi}{2^d}2^r(t-1)\right)^2 =$$
$$d + \sum_{i=1}^{d}\sum_{j=i+1}^{d}\cos\frac{2\pi}{2^d}\left(2^i - 2^j\right)(t-1) \tag{83}$$

Substituting this result in (82) and changing the variable $t \rightarrow t+1$ we obtain the number of spanning trees in $W_{d,2^d}$:

$$\kappa\left(W_{d,2^d}\right) = \frac{2d(d-1)}{2^{d-2}}\prod_{t=1}^{2^{d-2}-1}\left(d^2 - d - \Phi(t)\right)^2 \tag{84}$$

where:

$$\Phi(t) = \sum_{i=1}^{d}\sum_{j=i+1}^{d}\cos\frac{2\pi}{2^d}\left(2^i - 2^j\right)t \tag{85}$$

In general, for Knödel graphs having arbitrary degree g and arbitrary number of vertices n, $W_{g,n}$, according to (77), the number of spanning trees can be expressed as follows:

$$\kappa\left(W_{g,n}\right) = \frac{2g}{n}\prod_{t=2}^{n/2}\left(g^2 - \left\|\sum_{r=1}^{g} w^{2^r(t-1)}\right\|^2\right) \tag{86}$$

A straightforward upper bound for the number of spanning trees of Knödel graphs $W_{g,n}$ can be obtained by cancelling the norm from (86):

$$\left\|\sum_{r=1}^{g} w^{2^r(t-1)}\right\|^2 = 0 \tag{87}$$

Therefore, for $\kappa\left(W_{g,n}\right)$ we obtain the upper bound:

$$\kappa\left(W_{g,n}\right) \leq \frac{2g^{n-1}}{n} \tag{88}$$

Since, for Knödel graphs $W_{g,n}$, the degree of a vertex g is upper bounded by $\lfloor \log_2 n \rfloor$ (see (56)), the bound from (88) can be expressed as follows:

$$\kappa\left(W_{g,n}\right) \leq \frac{2\left\lfloor \log_2 n \right\rfloor^{n-1}}{n} \tag{89}$$

Chapter 5

The global fault tolerance of interconnection networks

Fault-tolerance in interconnection networks is an important research topic. The more complex the systems become, the more likely it is to be vulnerable to faults, making fault-tolerance design even more worthy of investigation. Nowadays, we are experiencing an exponential growth of the interconnection networks in both size and complexity. Distributed computing, supercomputing, and real-time applications are areas where fault-tolerance design becomes crucial.

Fault tolerance in computer systems has been described by Avizienis in [12] as "the ability to execute specified algorithms correctly regardless of hardware failures and program errors". This is a fairly general concept, which is hard to apply in complex systems involving both hardware and software modules that work together to a specific task. Hayes brought this definition to a more concrete level in [163]. He defined the modules, generally called facilities, as nodes in a graph and the connections between the modules as edges in graph. The problem of the fault-tolerant system design reduces now to that of designing a fault-tolerant graph, which underlies the system, seen as an interconnection network.

Following the Hayes model, there are two main trends in fault-tolerant design of interconnection networks. In one such model, the faulty part of the network is simulated by the remaining healthy part, which slows-down the system in case of a failure (see [162, 177, 204] for example). We have to mention here two seminal papers by Harary and Hayes which treat the cases of edge failures [135] and node failures

[136] separately. They measured the fault-tolerance using two metrics: the *k-edge fault-tolerance* of a graph G, shortly *k-EGT(G)* [135], and the *k-node fault-tolerance* of a graph G, shortly *k-NFT(G)* [136].

Definition 13. *A graph or multigraph G^* is k-edge fault-tolerant with respect to a graph G if every graph obtained by removing any k edges from G^* contains G.*

Definition 14. *A graph or multigraph G^* is k-node fault-tolerant with respect to a graph G if every graph obtained by removing any k nodes from G^* contains G.*

Note that, in both above definitions, graph G^* can be reduced to a minimum number of edges, respectively nodes, in which case we obtain optimal *k-NFT(G)*, respectively optimal *k-EFT(G)* graphs. Therefore, in each case, for every graph G, we can define a scalar representing the minimum number of edges, respectively nodes, of G^*.

The other trend is concerned with adding spare nodes or edges, such that, in case of a failure, the system can be reconfigured to simulate a desired topology that maintains the performance of the system [233, 234, 10, 6, 45, 254]. Most of the previous work in fault tolerance has been done around particular network topologies like arrays [10, 64], trees [84, 217], meshes [45, 188, 254, 246], hypercubes [161, 209, 11, 162, 54, 53] , butterfly networks [179], recursive circulants [247], etc. A more general approach can be found in [85, 14]. For a survey of some of the parameters used in network fault tolerance design as Menger number, Rabin number, fault-tolerant diameter, wide-diameter, restricted connectivity, and (l,w)-dominating number we refer the reader to [257]. Note that global characterisations of interconnection networks from other perspectives (robustness and resilience for example) also exist in the literature. For an overview of most of these parameters we refer the reader to [42].

We bring the notion of fault tolerance of a network to a higher level, independent of the two models described above, faulty nodes or faulty edges, and independent of a particular network topology. Our goal is to find a measure of the fault tolerance of a network that would allow us to compare two networks having different topologies in terms of their ability to support faults. This concept is called the *global fault tolerance*. In this chapter we describe such a metric and we analyze some of its properties. We exemplify the metric on two comparative analyzes:

- Three families of infinite minimum broadcast graphs: the hypercube, the recursive circulant and the Knödel graph.

- Five families of hypercubic graphs: butterfly, wrapped butterfly, shuffle exchange, de Bruijn, and cube connected cycles.

5.1 A metric for the global fault tolerance

Throughout this chapter we model an interconnection network as an undirected unweighted graph $G = (V, E)$, where V represents the set of nodes and E represents the set of edges. The length of a minimum path between two nodes $i, j \in V$ equals the number of edges in that path. We assume that the graphs are simple and connected, but the results can also be extended to disconnected multigraphs.

We denote by A_G the adjacency matrix of a graph G, having the elements defined as usual: $A_G[i, j] = 1$ if $(i, j) \in E$ and 0 otherwise. We denote by $dist_G(i, j)$ the distance between vertices i and j in G, and by $D(G)$ the diameter of G. We denote by K_n the complete graph on n vertices and by P_n the path on n vertices.

In order to introduce the global fault-tolerance we use a constructive approach. Assume that a message is travelling from vertex i to vertex j. The main point of a fault-tolerant design is to be able to reroute this message in case of a node or an edge failure. For rerouting, one must have this new route available. The more such routes are available, the more tolerant the graph is to faults, with respect to the particular route between vertices i and j. Therefore, for a given topology and a given pair of points, say i and j, we would like to know how many routes are available, in order to see how fault tolerant is this particular route.

There are two issues regarding this number. The first issue concerns the length of such paths. Since the routing algorithms usually consider the shortest path between two nodes, our metric for the global fault-tolerance takes into account only the minimum length paths. Another concern involves the relationship between these paths. Our metric takes into account all different paths rather than disjoint paths alone. The reason for our choice is a computational one: the maximum number of vertex/edge disjoint paths in directed/undirected graphs has been proven to be NP-complete [180]. On the other hand, the number of different walks of length p between nodes i and j in a graph G is given by the $[i, j]$ entry of the matrix A_G^p [40], which can be computed in polynomial time.

Following the above framework, in a graph G we associate to each pair of nodes

(i, j) a number $\varphi_G(i, j, p)$ representing the number of different paths of length p between i and j in G. Considering p as the minimum distance between i and j, we obtain a characterisation of the fault tolerance of G relative to pair (i, j), denoted here by $FT_{i,j}(G)$:

$$FT_{i,j}(G) = \varphi_G(i, j, dist_G(i, j)) \tag{90}$$

Summing these values for all possible pairs of vertices in G we obtain a global characterisation of the fault tolerance of G, denoted here by $FT(G)$:

$$FT(G) = \sum_{i \neq j \in G} \varphi_G(i, j, dist_G(i, j)) \tag{91}$$

Unfortunately, this value is not normalized and we cannot use it to compare two graphs with different number of vertices. Nevertheless, we encounter two extreme cases: the complete graph and the path graph. Indeed, the complete graph is the most fault-tolerant one and the path graph is the least fault-tolerant one. We would like to attain the maximum global fault-tolerance for the complete graph and the minimum global fault-tolerance for the path graph. Trying to meet these constraints, we define the global fault-tolerance of a graph as follows.

Definition 15. *The global fault-tolerance of a connected unweighted undirected graph G on $n \geq 2$ vertices, denoted by $GFT(G)$, is defined as follows:*

$$GFT(G) = \frac{\sum\limits_{i \neq j \in G} \varphi_G(i, j, dist_G(i, j))}{\sum\limits_{i \neq j \in K_n} \varphi_{K_n}(i, j, dist_G(i, j))} \tag{92}$$

The meaning of the terms from Definition 15 is as follows:

- $\sum\limits_{i \neq j \in G} \varphi_G(i, j, dist_G(i, j))$ represents the sum of all different paths of minimum length between any two vertices in graph G.
- $\sum\limits_{i \neq j \in K_n} \varphi_{K_n}(i, j, dist_G(i, j))$ represents the sum of all different paths of length $dist_G(i, j)$ between all vertex pairs (i, j) in the complete graph on n vertices, K_n.

By default we consider that the global fault-tolerance of the one-node graph is 1, since this represents the complete graph on one vertex. Note that $\varphi_{K_n}(i, j, dist_G(i, j))$ represents the number of walks of length $dist_G(i, j)$ between vertices i and j in K_n, regardless of the fact that the minimum distance is computed in G. This allows us to compare how many paths of same length between i and j are available in G compared to K_n, and to scale the value of $GFT(G)$ to that of a complete graph.

For the summation term in the scaling coefficient a closed formula has been computed in [44] (see also the notes of Emeric Deutsch in Sloane's Encyclopedia[1], entries A015521, A015531, and related):

$$\varphi_{K_n}(i,j,p) = \frac{(n-1)^p - (-1)^p}{n} \tag{93}$$

for $i \neq j$ and $1 \leq p \leq n-1$.

In what follows we give a few properties of the global fault-tolerance.

Property 22. *For any simple graph G, $GFT(G) \leq 1$.*

This is because

$$\sum_{i \neq j \in G} \varphi_G(i,j,dist_G(i,j)) \leq \sum_{i \neq j \in K_n} \varphi_{K_n}(i,j,dist_G(i,j)) \tag{94}$$

for any simple graph.

Property 23. *For any simple graph G on n vertices,*

$$GFT(G) \geq \frac{n^2(n-1)(n-2)^2}{2(n-1)^{n+1} - 2(n-1)^3 - n(n-2)^2} \tag{95}$$

This is because

$$\begin{aligned} &\min\left\{\sum_{i \neq j \in G} \varphi_G(i,j,dist_G(i,j))\right\} = \\ &= \sum_{i \neq j \in P_n} \varphi_{P_n}(i,j,dist_{P_n}(i,j)) = \\ &= n(n-1) \end{aligned} \tag{96}$$

and

$$\begin{aligned} &\max\left\{\sum_{i \neq j \in K_n} \varphi_{K_n}(i,j,dist_G(i,j))\right\} = \\ &= 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} \frac{(n-1)^{j-i} - (-1)^{j-i}}{n} \leq \\ &\leq \frac{2(n-1)^{n+1} - 2(n-1)^3 - n(n-2)^2}{n(n-2)^2} \end{aligned} \tag{97}$$

[1] http://www.research.att.com/~njas/sequences/

We can see that the lower bound from Property 23 tends to 0 as n increases, scaling the global fault tolerance to the $(0, 1)$ interval.

Property 24. *For any tree T on n vertices, $GFT(T) \geq GFT(P_n)$.*

This is due to the fact that the number of minimum length paths between any two vertices in a tree is one. Therefore,

$$\sum_{i,j \in T} \varphi_T\left(i, j, dist_T\left(i, j\right)\right) = \sum_{i,j \in P_n} \varphi_{P_n}\left(i, j, dist_{P_n}\left(i, j\right)\right) \tag{98}$$

and the denominator from (92) can only decrease, since

$$\sum_{i,j \in T} dist_T\left(i, j\right) \leq \sum_{i,j \in P_n} dist_{P_n}\left(i, j\right) \tag{99}$$

A simple algorithm to compute *GFT(G)* can be formally described as follows:

---

input: a graph G on $n \quad > 2$ vertices, given by A_G
output: $GFT(G)$
compute all node pairs minimum distance
sort the node pairs by the minimum distance
group the pairs with the same minimum distance in $D(G)$ lists: $L[D(G)]$
$B = A_G$
$\phi_G = \phi_{Kn} = 0$
for $k = 1$ to $D(G)$
  for each pair (i, j) in $L[k]$
    $\phi_G = \phi_G + B[i, j]$
  end for
  $\phi_{Kn} = \phi_{Kn} + \frac{(n-1)^k - (-1)^k}{n} *$ size of $L[k]$
  $B = B * B$;
end for
$GFT(G) = \frac{\varphi_G}{\varphi_{K_n}}$

Here, a question naturally arises: How much computational effort do we need in order to compute the global fault tolerance? Since the graph is unweighted, the distances can be computed in $O\left(m\, n\right)$, where m is the number of edges, and sorted in $O\left(n^2 \log n\right)$. The numerator from Definition 15, $\sum_{i,j \in G} \varphi_G\left(i, j, dist_G\left(i, j\right)\right)$, is based on matrix multiplications. The best known algorithm for matrix multiplication works in $O(n^{2.376})$ [69]. Since we need to compute $A_G^{D(G)}$, where $D(G)$ is the diameter of G, the time complexity of computing this term is in $O(D(G)\,n^{2.376})$. Therefore, the worst case complexity is in $O(n^{3.376})$. The space complexity is $O(n^2)$ due to the use of the adjacency matrix.

From Property 23 we can see that, for graph topologies close to a path, the ratio becomes exponential and, for large values of n, it becomes hard to be computed numerically. In order to avoid this difficulty we can further scale the global fault-tolerance using a logarithm factor:

$$LGFT(G) = \frac{\log \sum\limits_{i \neq j \in G} \varphi_G(i, j, dist_G(i, j))}{\log \sum\limits_{i \neq j \in K_n} \varphi_{K_n}(i, j, dist_G(i, j))} \tag{100}$$

where *LGFT(G)* stands for *logarithmic global fault-tolerance* of graph G.

5.2 Comparative analyses

In this section we first apply the metric proposed in Section 5.1 for the global fault tolerance to a class of graphs called *minimum broadcast graphs.* The main reason to choose this class of graphs is that, in order to build a super computer for example, one needs a topology which can offer support for the communication primitives: routing (one-to-one transmission), broadcasting (one-to-all transmission), and gossiping (all-to-all transmission). There are two main constraints for such a topology: the time needed to accomplish one primitive and the cost of the topology, which becomes a crucial issue as we are scaling down the circuits. As we saw in the introduction, minimum broadcast graphs are topologies that meet two constraints: they offer support for broadcasting in $\lceil \log n \rceil$ time and have the minimum number of edges. Minimum broadcast graphs are generally hard to obtain for an arbitrary number of vertices. Nevertheless, three infinite families of minimum broadcast graphs can be defined for 2^k vertices: the hypercube, the recursive circulant [224], and the Knödel graph [191]. All of them offer support for the three communication primitives in $O(\log n)$ time [203, 224, 151], and have the minimum number of edges $n \log n$.

One can find different representations of an algebraic form for these three topologies in the literature. We choose the following representations, which allow us to compare them in terms of their internal structure.

Definition 16. *The hypercube of degree k is the graph* $Q_k = (V, E)$ *on* $n = 2^k$ *vertices, with the set of vertices* $V = \{0, 1, ..., 2^k - 1\}$*, and the set of edges:*

$$E = \{(i,j) \,|\, j - i = 2^r, 0 \le i < j \le n-1, 0 \le r \le k-1\} \tag{101}$$

Definition 17. *The recursive circulant of degree k and order n is the graph* $G_{n,k} = (V, E)$ *with the set of vertices* $V = \{0, 1, ..., n-1\}$*, and the set of edges:*

$$E = \{\, (i,j) \,|\, i + k^r \equiv j (\text{mod} n),\ 0 \le i,j \le n-1,\ 0 \le r \le \lceil \log_k n \rceil \} \tag{102}$$

Definition 18. *The Knödel graph of degree k and even order n and is the graph* $W_{n,k} = (V, E)$ *with the set of vertices* $V = \{0, 1, ..., n-1\}$*, and the set of edges:*

$$E = \{\, (i,j) \,|\, i + j \equiv 2^r - 1 (\text{mod} n),\ 0 \le i,j \le n-1,\ 1 \le r \le k\} \tag{103}$$

It turns out that Q_k, $G_{2^k,4}$, and $W_{2^k,k}$ are all minimum broadcast graphs on 2^k vertices and $k2^{k-1}$ edges [95, 224, 99]. Moreover, they are regular graphs of degree k, can be defined as Cayley graphs, and they are nonisomorphic for $k > 3$ [105]. One of the most interesting facts regarding the three families of graphs is their diameter: $D(Q_k) = k$, $D(G_{2^k,4}) = \lceil (3k-1)/4 \rceil$ [224], and $D(W_{2^k,k}) = \lceil (k+2)/2 \rceil$ [109]. A comprehensive comparison regarding other network parameters of these three topologies is given in [108], which we reproduce here (Table 7).

Table 7: Comparison of the main network parameters between hypercubes, recursive circulants, and Knödel graphs.

Properties		H_k	$G(2^k;4)$	$W_{k,2^k}$
Number of vertices		2^k	2^k	2^k
Vertex-connectivity		k	k	$\in \left(\frac{2k}{3}, k\right]$
Edge-connectivity		k	k	k
Degree		k	k	k
Diameter		k	$\left\lceil \frac{3k-1}{4} \right\rceil$	$\left\lceil \frac{k+2}{2} \right\rceil$
Vertex-transitivity		Yes	Yes	Yes
Edge-transitivity		Yes	No	No
Spanning subgraph	Hamiltonian cycle	Yes	Yes	Yes
	Binomial Tree	Yes	Yes	Yes

In Figure 5.2 we present the logarithmic global fault tolerance of the three families of graphs against the number of vertices. We have computed the $LGFT$ for up to 2^{12}

Table 8: Logarithmic global fault tolerance values for three families of minimum broadcast graphs: hypercubes, recursive circulants, and Knödel graphs.

Dimension	Hypercube	Recursive circulant	Knödel graph
2	0.73836	0.73836	0.73836
3	0.60000	0.62763	0.65129
4	0.51435	0.53627	0.55829
5	0.45525	0.46962	0.52320
6	0.41129	0.44440	0.46121
7	0.37686	0.40249	0.44425
8	0.34890	0.36493	0.39954
9	0.32559	0.33753	0.38652
10	0.30578	0.32899	0.35061
11	0.28866	0.30780	0.34330

vertices and we represent the Ox values also on a logarithmic scale ($n = 2^{degree}$) (see the values in Table 8). We can see that the Knödel graph is the most fault tolerant, according to our metric, followed by the recursive circulant and the hypercube. Also, it can be seen that the fault tolerance constantly decreases as the number of vertices increases. This is due to the fact that the global fault tolerance decreases if the density of the network increases. An interesting fact is that the GFT behaviour captures the diameter property. For example, for the Knödel graph, the diameter increases with step two in terms of the graph degree. This can be seen as inflexion points on the upper curve. The same thing can be observed in the curve representing the recursive circulant, for which the diameter increases with step 4. As a last remark, we can see that, for degree 3, all of them have the same GFT since they are isomorphic when $n = 8$.

The second comparative analysis focuses on five families of hypercubic graphs: butterfly (BF), wrapped butterfly (WBF), shuffle exchange (SE), de Bruijn (DB), and cube connected cycles (CCC). All of them have good communication properties (routing, broadcasting, gossiping, etc.) and have been widely studied in the literature. We refer to [203] for a detailed description of their communication properties. The results are presented in Figure 8 for six dimensions (3 to 8) using again the logarithmic global fault tolerance. One can see that de Bruijn graph has the biggest global fault tolerance since the cube connected cycles has the smallest global fault tolerance.

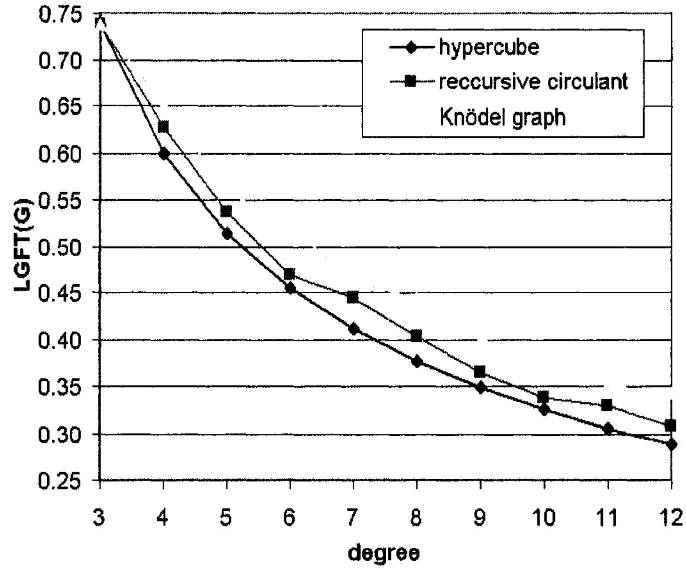

Figure 7: The logarithmic fault tolerance of three infinite families of minimum broadcast graphs

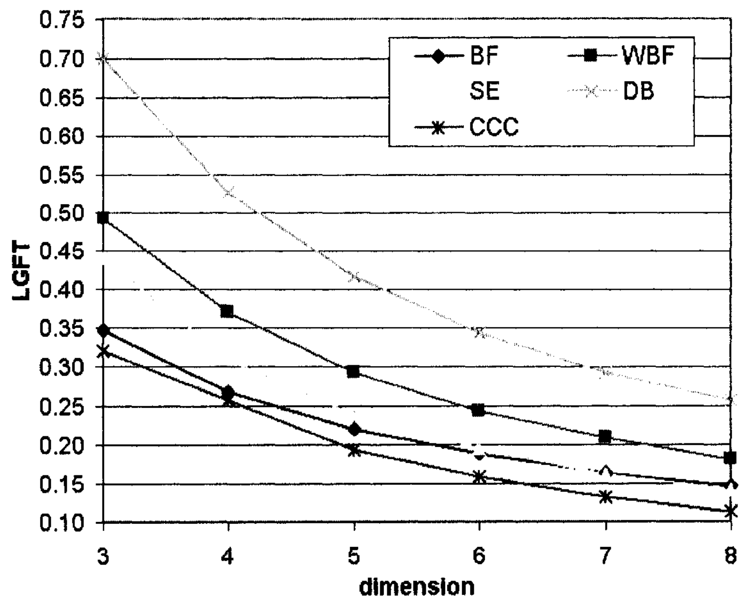

Figure 8: The logarithmic fault tolerance of five families of hypercubic graphs (BF – butterfly, WBF – wrapped butterfly, SE – shuffle exchange, DB – deBruijn, CCC – cube connected cycles)

5.3 Observations

In this chapter we introduce a new concept in fault tolerance: the global fault tolerance of a network. We define it as the number of all different minimum length paths between each pair of vertices. We scale this metric to the $(0, 1)$ interval dividing it by the corresponding number of paths in the complete graph.

We use this metric in order to compare three infinite families of minimum broadcast graphs, namely the hypercube, the recursive circulant, and the Knödel graph, which are the best candidates for a supercomputer architecture. We show that, not only in terms of diameter, but also in terms of global fault tolerance, there are better architectures than the hypercube.

The ideal candidate for such a metric would be the maximum number of disjoint paths, but, since computing this number is NP complete [180], we use the number of different minimum length paths. Surely, this comes at a price: the global fault tolerance cannot capture some of the network characteristics, especially for small density networks.

For example, a cycle of odd length will clearly have a different GFT compared to a cycle of even length. This is due to the fact that in a cycle of odd length, the number of different minimum length paths is one, since for a cycle of even length the number of different paths is two. Using the number of disjoint paths we always have two disjoint paths between any pair of vertices. Nevertheless, this behaviour tends to vanish for higher density graphs.

On the other hand, it may happen that more than one minimum path between the same pair of vertices goes through the same vertex (edge). If this vertex (edge) becomes faulty, all the counted minimum paths will be unavailable. Although this might be seen as a major drawback, in most cases, our metric is still able to capture this effect. For example, we chose two regular networks with the same number of vertices, same degree and same density, but obviously with different fault tolerances (network A in Figure 9 and network B in Figure 10). For these networks we computed the GFT and the $LGFT$ in Table 9, and it can be seen that, as expected, the first network has a higher global fault tolerance. Note that the global fault tolerance can be also computed for multigraphs with disconnected components. Finally, the algebraic methods used to compute the global fault tolerance suggest that this parameter could be analytically computed for highly regular network topologies.

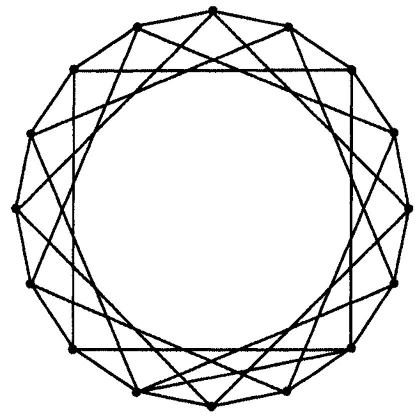

Figure 9: A regular network on 16 vertices having a high global fault tolerance.

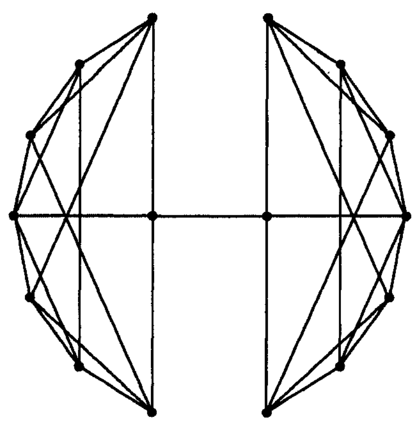

Figure 10: A regular network on 16 vertices having a low global fault tolerance.

Table 9: The global fault tolerance and the logarithmic global fault tolerance for networks from Figures 9 and 10.

Network	GFT	$LGFT$
A	0.035	0.62
B	0.001	0.47

Chapter 6

Conclusions

Interconnection networks are used both in parallel processing and in communication systems to exchange information. Their exponential growth challenges the research community to study the communication primitives, or more specifically routing, broadcasting, multicasting, and gossiping.

In this thesis we study interconnection networks mainly from the perspective of broadcasting. Our studies converge from two directions. One of them is concerned with aspects of the minimum broadcast time problem. Here we describe two new properties of the minimum broadcast time function, one of them yielding a heuristic for solving the minimum broadcast time problem. The main characteristic of this algorithm is its original iterative approach: at each iteration it tries to find a better solution based on the previous one. This leads to the possibility of finding an optimal solution for small instances which involve a reasonable number of iterations.

There is a considerable amount of literature dedicated to the NP-completenes, approximability, and inapproximability of the minimum broadcast time problem, but no results regarding the size of this problem. We estimate its size by mapping the set of the broadcast schemes into the set of rooted spanning trees, and further into the set of spanning trees. We give tight upper and lower bounds for the cardinality of the set of broadcast schemes, along with some results regarding particular topologies. More precisely, we provide exact values for complete bipartite graphs and an upper bound for regular graphs.

Another direction of our study consists of fundamental properties of interconnection networks and properties with implications in communication primitives, with a special emphasis on broadcasting. In this category we present two studies. In the first study we introduce a new metric for the global fault tolerance of a network. Here, the goal is to define a metric that would allow us to compare two networks in terms of their tolerance to faults, regardless of the type of fault (edge or node fault) and regardless of their size and density. We identify an ideal candidate for such a metric based on the maximum number of vertex disjoint paths between each pair of vertices. Since computing this number is NP-complete, we propose a surrogate based on the number of shortest paths between each pair of vertices, which is computable in polynomial time. We apply this metric to compare two classes of graphs in terms of their global fault tolerance: three infinite families of minimum broadcast graphs and five infinite families of hypercubic graphs.

The second study is dedicated to the spectra of Knödel graphs. This family of graphs gain the attention of the scientific community due to their good communication properties: logarithmic routing, minimum broadcast graphs and minimum gossip graphs. Knödel graphs appear to be the main countercandidate of the hypercube for regular network constructions. Perhaps one of the most interesting properties of the Knödel graphs is their diameter, which is half the diameter of the hypercube. On the other hand, the graph spectrum is a well known fundamental characteristic with wide applications in graph theory. As an application of the spectra of Knödel graphs we give a $O(n \log n)$ formula for the number of spanning trees.

The wide area covered by these studies inevitably generated a long list of references. Nevertheless, we did not mention numerous good papers, especially from broadcasting, due to specific graph topologies implied (directed graphs, ad-hoc or mobile networks) or because of different communication models. Also, we apologize for ignoring some seminal papers in fault tolerance or spectra of graphs fields.

As a general remark, we tried to draw attention to the broadcasting communication primitive and the two main related problems: minimum broadcast time and minimum broadcast graph. We found the hardness of these problems challenging and we tried to find new approaches, at a fundamental level, rather then improving previous results.

There still remain enough open questions, perhaps the most challenging being the conjecture that we made in Chapter 2 regarding the optimal broadcast schemes. This could also bring some light about the rate of convergence to an optimal solution of the iterative algorithm for the minimum broadcast time problem. The algebraic methods used in the computation of global fault-tolerance suggest possible closed formulas for some types of highly symmetric graphs (torus or hypercube, for example). Also, the cosine function used to describe the spectra of Knödel graphs implies errors for numerical computation. Minimizing these errors is also a challenging task.

Bibliography

[1] D. Agrawal, A. E. Abbadi, and R. C. Steinke. Epidemic algorithms in replicated databases, in *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Tucson, AZ, pp. 161–172, 1997.

[2] R. Ahlswede, L. Gargano, H.S. Haroutunian and L.H. Khachatrian. Fault-tolerant minimum broadcast networks, *Networks*, 27:293–307, 1996.

[3] R. Ahlswede, H.S. Haroutunian, and L.H. Khachatrian. Messy broadcasting in networks, in *Communications and Cryptography*, eds. R.E. Blahut, D.J. Costello Jr., U. Mauter, and T. Mittelholzer (Kluwer, Boston/Dordrecht/London), pp. 13–24, 1994.

[4] B. Aiello, F. T. Leighton, B. M. Maggs, and M. Newman. Fast algorithms for bit-serial routing on a hypercube, *ACM Symposium on Parallel Algorithms and Architectures*, pp. 55–64, 1990.

[5] A. C. Aitken. Two notes on matrices, In *Proc. Glasgow Math. Assoc.*, pp. 109–113, 1961-1962.

[6] M. Ajtai, N. Alon, J. Bruck, R. Cypher, C. T. Ho, M. Naor, and E. Szemerédi. Fault tolerant graphs, perfect hash functions and disjoint paths, in *Proc. IEEE Symp. on Foundations of Computer Science*, pp. 693–702, 1992.

[7] D. J. Aldous. The random walk construction of uniform spanning trees and uniform labelled trees, *SIAM Journal on Discrete Mathematics*, 3(4):450–465, 1990.

[8] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast, *Journal of Computer and System Sciences*, 43:290–298, 1991.

[9] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. Single round simulation of radio networks, *Journal of Algorithms*, 13:188–210, 1992.

[10] N. Alon and F.R.K. Chung. Explicit construction of linear sized tolerant networks, *Discrete Mathematics*, 72(1-3):15–19, 1988.

[11] F. Annexstein. Fault tolerance of hypercube-derivative networks, in *Proc. of the* 1^{st} *Annual ACM Symp. on Parallel Algorithms and Architectures*, pp. 179–188, 1989.

[12] A. Avizienis. Fault tolerant computing – an overview, *IEEE Computer*, 4:5–8, 1971.

[13] K. Aomoto and Y. Kato, Green functions and spectra on free products of cyclic groups, *Annales de l'Institut Fourier*, 38(1):59–86, 1988.

[14] A. Bagchi, A. Bhargava, J. Hopkins, A. Chaudhary, D. Eppstein, and C. Scheideler. The effect of faults on network expansion, in *Proceedings of the* 16^{th} *ACM Symposium on Parallelism in Algorithms and Architectures, Barcelona, Spain*, pp. 286–293, 2004.

[15] A. Bagchi , S. L. Hakimi , J. Mitchem , and E. Schmeichel. Parallel algorithms for gossiping by mail, *Information Processing Letters*, 34(4):197–202, 1990.

[16] A. Bar-Noy, J. Bruck, C.T. Ho, S. Kipnis, and B. Schieber. Computing global combine operations in the multi-port postal model, *IEEE Trans. Par. Distr. Syst.*, 6:896–900, 1995.

[17] A. Bar-Noy, S. Guha, J. Seffi, and N. Baruch. Multicasting in heterogeneous networks, in *STOC'98*, pp. 448–453, 1998.

[18] A. Bar-Noy, C.-T. Ho. Broadcasting multiple messages in the multiport model, *IEEE Transactions on Parallel and Distributed Systems*, 10(5):500–508, 1999.

[19] A. Bar-Noy and S. Kipnis. Broadcasting multiple messages in simultaneous send/receive systems, in 5^{th} *IEEE Symp. Parallel, Distributed Processing*, pp. 344–347, 1993.

[20] A. Bar-Noy and S. Kipnis. Designing broadcasting algorithms in the postal model for message-passing systems, *Mathematical Systems Theory*, 27(5):431–452, 1994.

[21] A. Bar-Noy and S. Kipnis. Multiple message broadcasting in the postal model, *Networks*, 29(1):1–10, 1997.

[22] A. Bar-Noy, S. Kipnis, and B. Schieber. Optimal multiple message broadcasting in telephone-like communication systems, in *Proc. Sixth Symp. Parallel and Distributed Processing*, pp. 216–223, 1994.

[23] M. Barreto, P. Navaux, and M. Rivière. DECK: A new model for a distributed executive kernel integrating communication and multithreading for support of distributed object oriented application with fault tolerance support, in *Anales del Congresso Argentino de Ciencias de la Computación (Neuquén, AR)*, 2:623–637, 1998.

[24] M. Barreto, F. Oliveira, R. Ávila, and P. Navaux. DECK: an environment for parallel programming on clusters of multiprocessors, in *Proceedings of the Brazilian Symposium on Computer Architecture and High Performance Computing (UFSCar)*, São Pedro, Brazil, pp. 321–329, 2000.

[25] R. Bar-Yehuda, O. Goldreich, A. Itai. Efficient emulation of single-hop radio network with collision detection on multi-hop radio network with no collision detection, *Distributed Computing*, 5:67–71, 1991.

[26] D. Barth and P. Fraigniaud. Approximation algorithms for structured communication problems, in *the* 9^{th} *ACM Symposium on Parallel Algorithms and Architectures (SPAA '97)*, pp. 180–188, 1997. (Tech. Rep. LRI-1239, Univ. Paris-Sud)

[27] B. Beauquier, S. Perennes, and O. Delmas. Tight Bounds for broadcasting in the linear cost model, *Journal of Interconnection Networks*, 2(2):175–188, 2001.

[28] R. Beier and J. F. Sibeyn. A powerful heuristic for telephone gossiping, in the *7-th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2000)*, L'Aquila, Italy, 17–36, 2000.

[29] A. Bellaachia and A. Youssef. Personalized broadcasting in banyan-hypercube networks, *Fourth International Conference on Computer Communications and Networks (ICCCN '95)*, pp. 470–474, 1995.

[30] S. Berg. Random contact process, snowball sampling and factorial series distributions, *J. Appl. Probl.*, 20:31–46, 1983.

[31] C. Berge. *Graphs and Hypergraphs*, North-Holland Publishing Company, 1976.

[32] J.-C. Bermond, A. Ferreira, S. Pérennes, and J. G. Peters. Neighbourhood broadcasting in hypercubes, *Technical Report TR2004-12*, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, September 2004.

[33] J. C. Bermond and P. Fraigniaud. Broadcasting and NP-completeness, *Technical report CMPT-TR-92-01*, Simon Fraser University, 1992.

[34] J.-C. Bermond, P. Fraigniaud, and J. Peters. Antepenultimate broadcasting, *Networks*, 26:125–137, 1995.

[35] J.C. Bermond, H. A. Harutyunyan, A. L. Liestman, and S. Perennes. A note on Dimensionality of modified Knödel graphs, *International Journal of Foundations of Computing Science*, 8(2):109–116, 1997.

[36] J.-C. Bermond, P. Hell, A.L. Liestman, and J. Peters. Sparse broadcast graphs, *Discrete Appl. Math.*, 36:97–130, 1992.

[37] J-C. Bermond, P. Hell, A. L. Liestman, and J. G. Peters. Broadcasting in bounded degree graphs, *SIAM J. Discr. Math.*, 5(1):10–24, 1992.

[38] J.-C. Bermond and C. Peyrat. Broadcasting in deBruijn networks, in *Proc.* 19^{th} *Southeastern Conference on Combinatorics, Graph Theory and Computing, Congressus Numerantium*, 66:283–292, 1988.

[39] P. Berthome, A. Ferreira, and S. Perennes. Optimal information dissemination in star and pancake networks, *IEEE Transactions on Parallel and Distributed Systems*, 7(12):1292–1300, 1996.

[40] N. L. Biggs. *Algebraic Graph Theory*, Cambridge University Press, 1974.

[41] B.D. Birchler, A. Esfahanian, and E. Torng. Information dissemination in restricted routing networks, in *Proceedings of the International Symposium on Combinatorics and Applications*, pp. 33–44, 1996.

[42] U. Brandes and T. Erlebach. Network Analysis: Methodological Foundations, *Lecture Notes in Computer Science*, Vol. 3418, 2005.

[43] A. Broder. Generating random spanning trees, *Foundation of Computer Science*, pp. 444–447, 1989.

[44] R. A. Brualdi and H. J. Ryser. *Combinatorial matrix theory*, Cambridge Univ. Press, Cambridge, 1991.

[45] J. Bruck, R. Cypher and C.-T. Ho. Fault-tolerant meshes with small degree, *ACM Symposium on Parallel Algorithms and Architectures*, pp. 1–10, 1993.

[46] J. Bruck and C.T. Ho. Efficient global combine operations in the multi-port message passing systems, *Parallel Processing Letters*, 3:335–346, 1993.

[47] L. Caccetta and W. F. Smyth. Graphs of maximum diameter, *Discrete Mathematics*, 102:121–141, 1992.

[48] V. R. Cane. A note on the size of epidemics and the number of people hearing a rumour, *Journal of the Royal Statistical Society, Series B*, 28(3):487–490, 1966.

[49] A. Cayley. A theorem on trees, *Quarterly Journal of Mathematics*, 23:376–378, 1889.

[50] F. Chatelin and M. Ahués. *Eigenvalues of Matrices*, Chichester, Wiley, New York, 1993.

[51] S.C. Chau and A.L. Liestman. Constructing minimal broadcast networks, *J. Combin. Inform. System. Sci.*, 10:110–122, 1985.

[52] X. Chen. An upper bound for the broadcast function $B(n)$, *Chinese J. Comput.*, 13:605–611, 1990.

[53] J. Chen, I. Kanj, and G. Wang. Hypercube network fault tolerance: a probabilistic approach, *Journal of Interconnection Networks*, 6(1):17–34, 2005.

[54] J. Chen, G. Wang, and S. Chen. Routing in hypercube networks with a constant fraction of faulty nodes, *Journal of Interconnection Networks*, 2(3):283–294, 2001.

[55] P. Chinn, S. Hedetniemi, and S. Mitchell. Multiple message broadcasting in complete graphs, in *Proc. Tenth Southeastern Conf. on Combinatorics, Graph Theory and Computing*, Utilitas Mathematica, Winnipeg, pp. 251–260, 1979.

[56] I. Chlamtac and S. Kutten. Tree-based broadcasting in multihop radio networks, *IEEE Trans. on Communications*, 36(10):1209–1223, 1987.

[57] I. Chlamtac and S. Kutten. On broadcasting in radio networks – problem analysis and protocol design, *IEEE Trans. on Communications*, 33:1240–1246, 1985.

[58] I. Chlamtac and S. Kutten. A spatial-reuse TDMA/FDMA for mobile multi-hop radio networks, in *Proceedings IEEE INFOCOM*, pp. 389–94, Washington, DC, 1985.

[59] I. Chlamtac and O. Weinstein. The wave expansion approach to broadcasting in multihop radio networks, *IEEE Trans. on Communications*, 39:426–433, 1991.

[60] M. Chlebus, L. Gasieniec, A. Ostlin, and J.M. Robson. Deterministic broadcasting in radio networks, in *Proc. of the* 27^{th} *International Colloquium on Automata, Languages and Programming*, pp. 717-728, 2000.

[61] M. Chrobak, L. Gasieniec, and W. Rytter. Fast broadcasting and gossiping in radio networks, in *Proc. of the* 41^{st} *Annual IEEE Conference on Foundations of Computer Science*, pp. 575-581, 2000.

[62] E. Cockayne and A. Thomason. Optimal Multi-Message Broadcasting in Complete Graphs, in *Proc. of* 11^{th} *SE Conf. Combinatorics, Graph Theory, and Computing*, pp. 181–199, 1980.

[63] C. J. Colbourn, R. P. J. Day, and L. D. Nel. Unranking and Ranking Spanning Trees of a Graph, *J. Algorithms*, 10:271–286, 1989.

[64] R. J. Cole, B. M. Maggs, and R. K. Sitaraman. Reconfiguring Arrays with Faults Part I: Worst-Case Faults, *SIAM J. Comput.*, 26(6):1581–1611, 1997.

[65] F. Comellas and C. Dalfó, Optimal broadcasting in 2-dimensional manhattan street networks, *Parallel and Distributed Computing and Networks*, 246:135–140, 2005.

[66] F. Comellas, H. A. Harutyunyan, and A. L. Liestman. Messy broadcasting in mesh and torus networks, *Journal of Interconnection Networks*, 4:37–51, 2003.

[67] F. Comellas and P. Hell. Broadcasting in generalized chordal rings, *Networks*, 42(3):123–134, 2003.

[68] F. Comellas and M. Mitjana. Broadcasting in cycle prefix digraphs, *Discrete Applied Mathematics*, 83(1-3):31–39, 1998.

[69] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions, *J. of Symbolic Computation*, 9:251–280, 1990.

[70] M. Cosnard and A. Ferreira. On the real power of loosely coupled parallel architectures, *Parallel Processing Letters*, 1:103-111, 1991.

[71] D. M. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs*, Academic Press, New York, 1980.

[72] D. J. Daley and D. G. Kendall. Epidemics and rumors, *Nature*, 204:1118, 1964.

[73] D. J. Daley and D. G. Kendall. Stochastic rumours, *J. Inst. Maths. Applics.*, 1:42–55, 1965.

[74] W. J. Dally and C. L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks, *IEEE Trans. Comp.*, C-36, N.5:547–553, 1987.

[75] P. J. Davis. *Circulant Matrices*, Chichester, Wiley, New York, 1979.

[76] O. Delmas. Communications par commutation de circuits dans les réseaux d'interconnexion, *thèse de doctorat*, Université de Nice - Sophia Antipolis, U.F.R. de Sciences, Ecole Doctorale - Sciences pour l'Ingénieur, France, January 1997.

[77] P. Diaconis. *Group Representation in Probability and Statistics*, vol. 11 in *IMS Lecture Series*, Institute of Mathematical Statistics, 1988.

[78] M. Dietzfelbinger. Gossiping and broadcasting versus computing functions in networks, *manuscript*, Technische Universität Ilmenau, 1999.

[79] K. Diks and A. Pelc. Broadcasting with universal lists, *Networks*, 27(3):183–196, 1996.

[80] M. J. Dinneen. The Complexity of Broadcasting in Bounded-degree Networks, *Computer Research and Appications*, Los Alamos National Laboratory, New Maxico 87545, U.S.A., 1994.

[81] M.J. Dinneen, M.R. Fellows, and V. Faber. Algebraic constructions of efficient broadcast networks, in *Proceedings of the* 9^{th} *AAECC Meeting, LNCS*, 539:152–158, 1991.

[82] M. J. Dinneen, J. A. Ventura, M. C. Wilson and G. Zakeri. Compound constructions of broadcast networks, *Discrete Applied Mathematics*, 93:205–232, 1999.

[83] R. Dunstan. The rumour process, *J. Appl. Prob.*, 19:759–766, 1982.

[84] S. Dutt and J. P. Hayes. On designing and reconfiguring k-fault-tolerant tree architectures, *IEEE Transactions on Computers*, 39(4):490–503, 1990.

[85] S. Dutt and J. P. Hayes. Designing fault-tolerant systems using automorphisms, *Journal of Parallel and Distributed Computing*, 12(3):249–268, 1991.

[86] R. B. Ellis. Discrete Green's functions for products of regular graphs, *arxiv.org/abs/math.CO/0309080*, 2003.

[87] M. L. Elkin and G. Kortsarz. Combinatorial logarithmic approximation algorithm for directed telephone broadcast problem, in *Proc.* 34^{th} *Annual ACM Symp. on Theory of Computing*, Canada, Montreal, pp. 438–447, 2002.

[88] M. Elkin and G. Kortsarz. Sublogarithmic approximation for telephone multicast: path out of jungle, in *Proc. of the* 14^{th} *Annual ACM-Siam Symp. on Discrete Algorithms (SODA'03)*, USA, MD, Baltimore, 2003.

[89] M. Elkin and G. Kortsarz. Logarithmic inapproximability of the radio broadcast problem, *Journal of Algorithms*, 52(1):8–25, 2004.

[90] M. Elkin and G. Kortsarz. Polylogarithmic Inapproximability of the Radio Broadcast Problem, in *Proc. of the* 7^{th} *International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pp. 105–114, Cambridge, MA, 2004.

[91] M. Elkin and G. Kortsarz. Polylogarithmic additive hardness of approximating radio broadcast problem, *SIAM Journal of Discrete Mathematics*, 19(4):881–899, 2005.

[92] M. Elkin and G. Kortsarz. An improved algorithm for radio broadcast, in *Proc. ACM-SIAM on Discrete Algorithms (SODA'05)*, Vancouver, BC, Canada, 2005.

[93] P. Eugster, S. Handurukande, R. Guerraoui, A. Kermarrec and P. Kouznetsov. Lightweight probabilistic broadcast, *ACM Transactions on Computer Systems*, 21(4):341-374, 2003.

[94] S. Even and B. Monien. On the number of rounds necessary to disseminate information, in *Proc. of the* 1^{st} *ACM Symposium on Parallel Algorithms and Architectures, SPAA'89*, pp. 318–327, 1989.

[95] A. Farley. Minimal broadcast networks, *Networks*, 9:313–332, 1979.

[96] A. Farley. Minimum-time line broadcast networks, *Networks*, 10:59–70, 1980.

[97] A.M. Farley. Broadcast time in communication networks, *SIAM J. Applied Math.*, 39(2):385–390, 1980.

[98] A. M. Farley and S. T. Hedetniemi. Broadcasting in grid graphs, in *Proc. of the* 9^{th} *Conf. Combinatorics, graph theory, and computing*, Utilitas Mathematica Publishing Inc., Winnipeg, pp. 275–288, 1978.

[99] A. Farley, S. Hedetniemi, S. Mitchel, and A. Proskurowski. Minimum broadcast graphs, *Discrete Mathematics*, 25:189–193, 1979.

[100] U. Feige, D. Peleg, P. Raghavan and E. Upfal, Randomized broadcast in networks, *Random Struct. Algorithms*, 1(4), 447–460, 1990.

[101] R. Feldmann, J. Hromkovič, S. Madhavapeddy, B. Monien, and P. Mysliwietz. Optimal algorithms for dissemination of information in generalized communication modes, *Discrete Applied Mathematics*, 53(1-3):55–78, 1994.

[102] W. Feller. *An Introduction to Probability Theory and its Applications*, 2^{nd} ed., Wiley, New York, 1971.

[103] G. Fertin. A study of minimum gossip graphs, *Discrete Math.*, 215(1–3):33–57, 2000.

[104] G. Fertin and R. Labahn. Compounding of gossip graphs, *Networks*, 36(2):126–137, 2000.

[105] G. Fertin and A. Raspaud. Families of graphs having broadcasting and gossiping properties, in *Proc. of the* 24^{th} *International Workshop on Graph-Theoretic Concepts in Computer Science (WG'98)*, Smolenice, *LNCS*, 1517:63–77, 1998.

[106] G. Fertin and A. Raspaud. Neighbourhood communications in networks, in *Proc. Euroconference on Combinatorics, Graph Theory and Applications (COMB01), Electronic Notes on Discrete Mathematics*, 10, 2001.

[107] G. Fertin and A. Raspaud. k-Neighbourhood Broadcasting, in *Proceedings of the* 8^{th} *Int. Colloquium on Structural Information and Communication Complexity (SIROCCO 2001)*, pp. 133146, 2001.

[108] G. Fertin and A. Raspaud. A survey on Knödel graphs, *Discrete Applied Mathematics*, 137(2):173–195, 2004.

[109] G. Fertin, A. Raspaud, O. Sýkora, H. Schröder, and I. Vrto. Diameter of Knödel graph, in *the* 26^{th} *International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2000), Lecture Notes in Computer Science*, 1928:149–160. Springer-Verlag, 2000.

[110] M. Flammini and S. Pérennès. Lower bounds on the broadcasting and gossiping time of restricted protocols, *SIAM Journal on Discrete Mathematics*, 17:521-540, 2004.

[111] E. Fleury. Communications, routage et architectures des machines à mémoire distribuée - Autour du routage wormhole, *thèse de doctorat*, Université de Lyon, Ecole Normale Supérieure de Lyon, France, 1996.

[112] P. Fraigniaud. Hierarchical broadcast networks, *Information Processing Letters*, 68(6):303–305, 1998.

[113] P. Fraigniaud. Approximation Algorithms for Collective Communications with Limited Link and Node-Contention, *Tech. Rep. LRI-1264*, Univ. Paris-Sud, France, 2000.

[114] P. Fraigniaud. Approximation algorithms for minimum-time broadcast under the vertex-disjoint paths mode, in *the* 9^{th} *Annual Eur. Symp. on Alg., ESA '01, LNCS*, 2161:440–451, 2001.

[115] P. Fraigniaud. Minimum-time broadcast under edge-disjoint paths modes, in *the* 2^{nd} *International Conference on Fun with Algorithms (FUN '01)*, pp. 133–148, 2001.

[116] P. Fraigniaud. A note on line broadcast in digraphs under the edge-disjoint paths mode, *Discrete Applied Mathematics*, 144(3):320–323, 2004.

[117] P. Fraigniaud and and E. Lazard. Methods and problems of communication in usual networks, *Discrete Applied Math.*, 53:79–134, 1994.

[118] P. Fraigniaud and J. G. Peters. Minimum linear gossip graphs and maximal linear (Δ, k)-gossip graphs, *Networks*, 38:150–162, 2001.

[119] P. Fraigniaud and S. Vial. Approximation algorithms for broadcasting and gossiping, *Journal of Parallel and Distributed Computing*, 43(1), 47–55, 1997.

[120] P. Fraigniaud and S. Vial. Comparison of heuristics for one-to-all and all-to all communications in partial meshes, *Parallel Processing Letters*, 9(1):9–20, 1999.

[121] H. Frank, I. Gitman, and R. van Slyke. Routing in packet switching broadcast radio networks, *IEEE Transactions on Communication*, 24:926–930, 1976.

[122] A. Frieze and G. Grimmett. The shortest-path problem for graphs with random arc-lengths, *Discr. Appl. Mathematics*, 10:57–77, 1985.

[123] S. Fujita. Neighborhood information dissemination in the star graph, *IEEE Transactions on Computers*, 49(12):1366–1370, 2000.

[124] S. Fujita, S. Perennes, and J. Peters. Neighbourhood gossiping in hypercubes, *Parallel Processing Letters*, 8(2):189–195, 1998.

[125] I. Gaber and Y. Mansour. Broadcast in radio networks, *ACM–SIAM Symposium on Discrete Algorithms*, San Francisco, CA, USA, pp. 577–585, 1995.

[126] M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*, W. F. Freeman, San Francisco, 1979.

[127] L. Gargano and U. Vaccaro. On the construction of minimal broadcast networks, *Networks*, 19:673–689, 1989.

[128] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. S. Sunderam. *PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Networked Parallel Computing*, MIT Press, Cambridge, MA, USA, 1994.

[129] C. Godsil and G. Royle. *Algebraic Graph Theory*, Springer-Verlag, New York, Berlin, Heidelberg, 2001.

[130] W. Goffman. Mathematical approach to the spread of scientific ideas–the history of mast cell research, *Nature*, 212(61):449452, 1966.

[131] W. Goffman and V. A. Newill. Generalization of epidemic theory, an application to the transmission of ideas, *Nature*, 204:225–228, 1964.

[132] W. Goffman and V. A. Newill. Communication and epidemic processes, *Proc. Royal Soc. A*, 298:316–334, 1967.

[133] M. Grigni and D. Peleg. Tight bounds on minimum broadcast networks, *SIAM J. Discr. Math.*, 4:207–222, 1991.

[134] F. Guo. Finding the Minimum Broadcast Time of Bounded Degree Networks by Backtracking, *master thesis*, University of Auckland, 2001.

[135] F. Harary and J.P. Hayes. Edge fault tolerance in graphs, *Networks*, 23:135–142, 1993.

[136] F. Harary and J.P. Hayes. Node fault tolerance in graphs, *Networks*, 27(1):19–23, 1996.

[137] F. Harary and G. Prins. The number of homeomorphically irreducible trees, and other species, *Acta Math.*, 101:141–162, 1959.

[138] F. Harary and A. J. Schwenk. The spectral approach to determining the number of walks in a graph, *Pacific Journal of Mathematics*, 80(2):443–449, 1979.

[139] H. S. Haroutunian. Minimal broadcast networks, in *Fourth International Colloquium on Coding Theory*, Dilijan, Armenia, pp. 36-40, 1991.

[140] H. A. Harutyunyan. Multiple broadcasting in modified Knödel graphs, 7^{th} *International Colloquium on Structural Information and Communication Complexity (SIROCCO2000)*, LAquila, Italy, pp. 157–166, 2000.

[141] H. A. Harutyunyan. An efficient vertex addition method for broadcast networks, in 2^{nd} *Workshop on Combinatorial and Algorithmic Aspects of Networking*, Waterloo, Canada, 2005.

[142] H. A. Harutyunyan. Minimum Multiple Message Broadcast Graphs, *Networks*, 47(4):218–224, 2006.

[143] H. A. Harutyunyan and A. L. Liestman. Messy Broadcasting, *Parallel Processing Letters*, 8(2):149–159, 1998.

[144] H. A. Harutyunyan, A. L. Liestman. More broadcast graphs, *Discrete Applied Mathematics*, 98(1–2):81–102, 1999.

[145] H.A. Harutyunyan and A.L. Liestman. Improved upper and lower bounds for k-broadcasting, *Networks* 37:94–101, 2001.

[146] H.A. Harutyunyan and A.L. Liestman. k-Broadcasting in trees, *Networks*, 38(3):163–168, 2001.

[147] H. A. Harutyunyan and A. L. Liestman. On the monotonicity of broadcast function, *Discrete Mathematics*, 262(1):140–157, 2003.

[148] H. A. Harutyunyan, A. L. Liestman, K. Makino, and T. C. Shermer. Orderly broadcasting in trees, *manuscript.*

[149] H. A. Harutyunyan and C. D. Morosan. The spectra of Knödel graphs, *Informatica an International Journal of Computing and Informatics*, 30(3):295–299, 2006.

[150] H. A. Harutyunyan and C. D. Morosan. An iterative algorithm for the minimum broadcast time problem, in *Proceedings of the Second IASTED International Conference on Communication and Computer Networks CCN04*, Cambridge, MA, USA, pp. 447-452, 2004.

[151] H. A. Harutyunyan and C. D. Morosan. On the minimum path problem in Knödel graphs, in *Proceedings of the Second International Network Optimization Conference INOC2005*, Lisbon, Portugal, pp. 43–48, 2005. (also to appear in *Networks*, 2007)

[152] H. A. Harutyunyan and C. D. Morosan. On two properties of the minimum broadcast time function, in *the* 9^{th} *International Conference on Information Visualisation (IV05)*, London, UK, pp. 523–527, 2005. (also to appear in the *International Journal of Applied Mathematics and Computer Science, Graph Theory and its Applications*)

[153] H. A. Harutyunyan and C. D. Morosan. The global fault-tolerance of interconnection networks, in *Proceedings of the Seventh International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2006)*, Las Vegas, Nevada, pp. 171–176, 2006.

[154] H. A. Harutyunyan and B. Shao. An efficient heuristic for broadcasting in networks, *Journal of Parallel and Distributed Computing*, 66(1):68–76, 2006.

[155] H. A. Harutyunyan and B. Shao. A Heuristic for k-broadcasting in Arbitrary Networks, *IEEE Conference on Applications of Graph Theory IV03*, London, England, pp. 287–292, 2003.

[156] H. A. Harutyunyan and B. Shao. Optimal k-broadcast in Trees, *Congressus Numerantium*, vol. 164, 2003.

[157] H. A. Harutyunyan and P. Taslakian. Orderly broadcasting in a 2D torus, in *Proceedings of the Eighth International Conference on Information Visualisation (IV'04)*, pp. 370–375, 2004.

[158] H. A. Harutyunyan and Y. Xu. Broadcast graphs on odd number of vertices, in 5^{th} *Franco-Canadian Workshop on Combinatorial Algorithms (COMAL05)*, Hamilton, Canada, 2005.

[159] T. Hart and H. A. Harutyunyan. Improved messy broadcasting in hypercube and complete bipartite graphs, *Congressus Numerantium*, 156:124–140, 2002.

[160] T. Hart and H. A. Harutyunyan. Messy broadcasting in common communication networks, in *Proceedings of the* 21^{st} *Biennial Symposium on Communications*, Kingston, Canada, pp. 534-538, 2002.

[161] J. Hastad, T. Leighton, and M. Newman. Reconfiguring a hypercube in the presence of faults, in *Proc.* 19^{th} *ACM Symp. Theory of Computing*, pp. 274–284, 1987.

[162] J. Hastad, T. Leighton, and M. Newman. Fast computation using faulty hypercubes, in *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC'89)*, pp. 251–263, 1989.

[163] J. P. Hayes. A graph model for fault-tolerant computing systems, *IEEE Trans. Comput.*, C-25:875–884, 1976.

[164] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman. A survey of gossiping and broadcasting in communication networks, *Networks*, 18, 319–349, 1988.

[165] P. Hell and K. Seyffarth. Broadcasting in planar graphs, *Australas. J. Combin.*, 17:309–318, 1998.

[166] M-C. Heydemann, N. Marlin, and S. Perennes. Cayley graphs with complete rotations, *Technical Report TR-1155*, Laboratiore the Recherche en Informatique, Orsay, 1997.

[167] M-C. Heydemann, J. Opatrny, and D. Sotteau. Broadcasting and spanning trees in de Bruijn and Kautz networks, *Discrete Applied Math.*, 27-28:297–317, 1992.

[168] C. J. Hoelting, D. A. Schoenefeld, R. L. Wainwright. A genetic algorithm for the minimum broadcast time problem using a global precedence vector, in *Proc. of the 1996 ACM symposium on Applied Computing*, Philadelphia, Pennsylvania, USA, 258–262, 1996.

[169] J. Hromkovič,C.-D. Jeschke, and B. Monien. Optimal algorithms for dissemination of information in some interconnection networks, *Algorithmica*, 10(1):24–40, 1993.

[170] J. Hromkovič, R. Klasing, B. Monien, and R. Peine. Dissemination of information in interconnection networks (broadcasting and gossiping), D.-Z. Du, F. Hsu (Eds.), *Combinatorial network theory*, Kluwer Academic Publishers, Dordrecht, 125–212, 1995.

[171] J. Hromkovič, R. Klasing, and E.A. Stöhr. Dissemination of information in vertex-disjoint paths mode, *Computers and Artificial Intelligence*, 15(4):295–318, 1996.

[172] J Hromkovič, R. Klasing, W. Unger, and H. Wagener. Optimal algorithms for broadcast and gossip in the edge-disjoint path modes, in *Proceedings of the* 4^{th} *Scandinavian Workshop on Algorithm Theory*, pp.219–230, 1994.

[173] J. Hromkovič, R. Klasing, W. Unger, H. Wagener, and D. Pardubska. The complexity of systolic dissemination of information in interconnection networks, *Canada-France Conference on Parallel and Distributed Computing*, pp. 235–249, 1994.

[174] A. Jakoby, R. Reischuk, and C. Schindelhauer. The complexity of broadcasting in planar and decomposable graphs, *Discrete Applied Mathematics*, 83:179–206, 1998.

[175] K. Jansen and H. Muller. The minimum broadcast time problem for several processor networks, *Theoretical Computer Science*, 147:69–85, 1995.

[176] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman. Advances in packet radio technology, in *Proceedings of the IEEE*, vol. 66, no. 11, 1978.

[177] C. Kaklamanis, A. R. Karlin, F. T. Leighton, V. Milenkovic, P. Raghavan, S. Rao, C. Thomborson, and A. Tsantilas. Asymptotically tight bounds for computing with faulty arrays of processors, in *Proceedings of the* 31^{st} *Annual Symposium on Foundations of Computer Science*, pp. 285–296, 1990.

[178] S. Kapoor and H. Ramesh. Algorithms for Enumerating All Spanning Trees of Undirected and Weighted Graphs, *SIAM Journal on Computing*, 24(2):247–265, 1995.

[179] A. R. Karlin, G. Nelson, and H. Tamaki. On the fault tolerance of the butterfly, in *Proceedings of the 26^{th} ACM symposium on Theory of Computing*, Montreal, QC, Canada, pp. 125–133, 1994.

[180] R.M. Karp. On the computational complexity of combinatorial problems, *Networks*, 5:45–68, 1975.

[181] R. Karp, A. Sahay, E. Santos, and K.E. Schauser. Optimal broadcast and summation in the LogP model, in *Proc. Fifth Ann. Symp. Parallel Algorithms and Architectures*, pp. 142–153, 1993.

[182] R. M. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized Rumor Spreading, *IEEE Symposium on Foundations of Computer Science*, pp. 565–574, 2000.

[183] W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics, in *Proceedings of the Royal Society of London, Series A*, 115(772):700–721, 1927.

[184] P. Kermani and L. Kleinrock. Virtual cut-through: a new computer communication switching technique, *Computer Networks*, 3:267–286, 1979.

[185] L. H. Khachatrian and H. A. Haroutunian. Minimal broadcast trees, in *XIV All Union School of Computing Networks* (in Russian), pp.36-40, 1989.

[186] L. H. Khachatrian and H. A. Haroutunian. Construction of new classes of minimal broadcast networks, in *Proc. Third International Colloquium on Coding Theory*, pp. 69–77, 1990.

[187] L. H. Khachatrian and H. S. Haroutunian. On optimal broadcast graphs, in *Fourth International Colloquium on Coding Theory*, Dilijan, Armenia, pp. 65–72, 1991.

[188] H.-C. Kim and J.-H. Park. Fault hamiltonicity of two-dimensional torus networks, in *Proc. of Workshop on Algorithms and Computation WAAC'00*, Tokyo, Japan, pp. 110–117, 2000.

[189] J.-H. Kim and K.-Y. Chwa. Optimal broadcasting with universal lists based on competitive analysis, *Networks*, 45(4):224–231, 2005.

[190] R. Klasing, B. Monien, R. Peine, and E. Stöhr. Broadcasting in Butterfly and DeBruijn Networks, *Symposium on Theoretical Aspects of Computer Science*, pp. 351–362, 1992.

[191] W. Knödel. New gossips and telephones, *Discrete Mathematics*, 13:95, 1975.

[192] G. Kortsarz and D. Peleg. Approximation algorithms for minimum-time broadcast, *SIAM Journal on Discrete Mathematics*, 8(3):401–427, 1995.

[193] D. R. Kowalski and A. Pelc. Deterministic broadcasting time in radio networks of unknown topology, in *Proceedings of the 43-rd Symposium on Foundations of Computer Science*, pp.63–72, 2002.

[194] J.-C.König and E. Lazard. Minimum k-broadcast graphs, *Discr. Appl. Math.*, 53:199–209, 1994.

[195] R. Labahn. Extremal broadcasting problems, *Discr. Appl. Math.*, 23:139–155, 1989.

[196] R. Labahn. Some minimum gossip graphs, *Networks*, 23:333–341, 1993

[197] R. Labahn. A minimum broadcast graph on 63 vertices, *Discrete Appl. Math.*, 53:247–250, 1994.

[198] H. D. Landahl. On the spread of information with time and distance, *Bull. Math. Biophys.*, 15:367–381, 1953.

[199] H. G. Landau and A. Rapoport. Contribution to the mathematical theory of contagion and spread of information, I. Spread through a thoroughly mixed population, *Bull. Math. Biophys.*, 15:173-183, 1953.

[200] E. Lazard. Broadcasting in DMA-bound bounded degree graphs, *Discr. Appl. Math.*, 37/38:387–400, 1992.

[201] S. Lee. *Information dissemination theory in communication networks: Design of c Broadcast Networks*, Ph.D. Dissertation, Dept. of Industrial and Manufacturing Engineering, Pennsylvania State University, 1999.

[202] S. Lee amd J. A. Ventura. An algorithm for constructing minimal c-broadcast networks, *Networks*, 38(1):6–21, 2001.

[203] T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, San Mateo, CA, 1992.

[204] F. T. Leighton, B. M. Maggs, and R. K. Sitaraman. On the fault tolerance of some popular bounded-degree networks, *IEEE Symposium on Foundations of Computer Science*, pp. 542–552, 1992.

[205] C. Li, T. E. Hart, K. J. Henry, and I. A. Neufeld. Average-case messy broadcasting, to appear in *Journal of Interconnection Networks*.

[206] A. Liestman and J. Peters. Broadcast networks of bounded degree, *SIAM J. Disc. Math.*, 4:531–540, 1988.

[207] A. L. Liestman and N. Pržulj. Minimum average time broadcast graphs, *Par. Proc. Lett.* 8:139–147, 1998.

[208] A. Liestman, T. Shermer, and M. Suderman. Broadcasting multiple messages in hypercubes, *International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '00)*, p. 274–281, 2000.

[209] M. Livingston, Q. Stout, N. Graham, and F. Hararay. Subcube fault-tolerance in hypercubes, *Technical Report CRL-TR12-87*, University of Michigan, Computing Research Laboratory, 1987.

[210] L. Lovász. Random Walks on Graphs: A Survey, *Combinatorics, Paul Erdös is Eighty*, (ed. D. Miklós, V. T. Sós, T. Szõnyi), János Bolyai Mathematical Society, Budapest, 2:353–398, 1996.

[211] M. Mahéo and J.-F. Saclé. Some minimum broadcast graphs, *Discrete Appl. Math.*, 53:275–285, 1994.

[212] M. Middendorf. Minimum broadcast time is NP-complete for 3-regular planar graphs and deadline 2, *Information Processing Letters*, 46(6):281–287, 1993.

[213] S. Mitchell and S. Hedetniemi. A census of minimum broadcast graphs, *J. Combin. Inform. System Sci.*, 5:141–151, 1980.

[214] I. M. Mkwawa and D. D. Kouvatsos. An optimal neighbourhood broadcasting scheme for star interconnection networks, *Journal of Interconnection Networks*, 4(1):103-112, 2003.

[215] I. M. Mkwawa and D. D. Kouvatsos. Neighbourhood communication schemes for cayley graphs of permutation groups, *Technical Report*, Department of computing, University of Bradford, 2003.

[216] I. M. Mkwawa and D. D. Kouvatsos. Broadcasting schemes for hypercubes with background traffic, *Journal of Systems and Software*, Special issue: *Performance modelling and analysis of computer systems and networks*, 73(1):3–14, 2004.

[217] H. Mori and J. Kambara. Fault tolerant tree using adaptive operational redundancy. in *Proceedings of 4^{th} International Conference on Wafer Scale Integration*, San Francisco, CA, USA, pp. 85-94, 1992.

[218] C. D. Morosan. On the number of broadcast schemes in networks, *Information Processing Letters*, 100(5):188–193, 2006.

[219] G. K. Osei and J. W. Thomson. The suppression of one rumour by another, *J. Appl. Prob.*, 14:127–134, 1977

[220] Öystein Ore. Diameters in graphs, *J. Combin. Theory*, 5:75–81, 1968.

[221] P. S. Pacheco. *Parallel Programming with MPI*, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1997.

[222] I. Pak, *personal webpage.*

[223] J.F. Palmer. The NCUBE family of parallel supercomputers, *IEEE Int'l Conf. Computer Design*, 1986.

[224] J.-H. Park and K.-Y. Chwa. Recursive circulant: a new topology for multicomputers networks. in *Proc. Int. Symp. Parallel Architectures, Algorithms and Networks ISPAN'94*, Kanazawa, Japan, pages 73–80, 1994.

[225] B. Pittel. On spreading a rumor, *SIAM J. Appl. Math.*, 47:213–223, 1987.

[226] A. Proskurowski. Minimum broadcast trees, *IEEE Trans. Comput.*, 30:363-366, 1981.

[227] K. Qiu and S.K. Das. A novel neighbourhood broadcasting algorithm on star graphs, in *Proc. of the 9^{th} International Conference on Parallel and Distributed Systems (ICPADS02)*, Taiwan, pp.37-41, 2002.

[228] A. Rapoport. Spread of information through a population with socio-structural bias. I. Assumption of transitivity, *Bull. Math. Biophys.*, 15:523-533, 1953.

[229] A. Rapoport. Spread of information through a population with socio-structural bias. II. Various models with partial transitivity, *Bull. Math. Biophys.*, 15:535-546, 1953.

[230] A. Rapoport. Spread of information through a population with socio-structural bias. III. Suggested experimental procedures, *Bull. Math. Biophys.*, 16:75-81, 1954.

[231] A. Rapoport and L. I. Rebhun. On the mathematical theory of rumor spread, *Bull. Math. Biophys.*, 14:375-383, 1952.

[232] R. Ravi. Rapid rumor ramification: approximating the minimum broadcast time, in *the* 35^{th} *Symposium on Foundation of Computer Science*, Montreal, QC, Canada, 202–213, 1994.

[233] A. L. Rosenberg. The Diogenes approach to testable fault-tolerant VLSI processor arrays, *IEEE Trans. Comput.*, C-32:21–27, 1983.

[234] A. L. Rosenberg. Fault-tolerant interconnection networks: a graph-theoretic approach, in *Proc. of* 9^{th} *Workshop on Graph-Theoretic Concepts in Computer Science*, pp. 286–297, 1983.

[235] J. de Rumeur. Communication dans les réseaux de processeurs, *Collection Etudes et Recherches en Informatique*, Masson, Paris, 1994.

[236] H. Sachs. Beziehungen zwischen den in einem graphen enthalten kreisen und seinem charakteristischen polynom, *Publ. Math. Debrecen*, 11:119–137, 1964.

[237] J.-F. Saclé. Lower bounds for the size in four families of minimum broadcast graphs, *Discrete Math.*, 150:359–369, 1996.

[238] P. Scheuermann and M. Edelberg. Optimal broadcasting in point-to-point computer networks, *Technical Report*, Northwestern University, 1981.

[239] P. Scheuermann and G. Wu. Heuristic algorithms for broadcasting in point-to-point computer network, *IEEE Transactions on Computers*, 33(9):804–811, 1984.

[240] C. Schindelhauer. Broadcasting time cannot be approximated within a factor of $57/56$-ε, *ICSI Technical Report TR-00-002*, 2002.

[241] B. Shao. On k-broadcasting in graphs, *PhD thesis*, Concordia University, Montreal, Canada, 2006.

[242] A. Shastri and S. Gaur. Multi-broadcasting in communication networks. I: Trees, *Proc. Int. Symp. on Communications (ISCOM97)*, pp. 167–170, 1997.

[243] Z. Shen. An optimal broadcasting schema for multidimensional mesh structures, in *Proceedings of the ACM symposium on Applied computing (SAC'03)*, Melbourne, Florida, ACM Press, pp. 1019–1023, 2003.

[244] P. J. Slater, E. J. Cockayne, and S. T. Hedetniemi. Information dissemination in trees, *SIAM J. Comput.*, 10(4), 692–701, 1981.

[245] M. Suderman. Multiple Message Broadcasting, *MS thesis*, Simon Fraser University, BC, Canada, 1999.

[246] M.-J. Tsai. Fault-Tolerant Routing in Wormhole Meshes, *Journal of Interconnection Networks*, 4(4):463–495, 2003.

[247] C.-H. Tsai, J. M. Tan, Y.-C. Chuang, and L.-H. Hsu. Hamiltonian properties of faulty recursive circulant graphs, *Journal of Interconnection Networks*, 3(3/4):273–290, 2002.

[248] J. A. Ventura and X. Weng. A new method for constructing minimal broadcast networks, *Networks*, 23:481–497, 1993.

[249] J. Vuillemin. A data structure for manipulating priority queues. *Comm. ACM*, 21:309–315, 1978.

[250] K. Wang. On the generalisations of circulants, *Linear algebra and its applications*, 25:197–218, 1979.

[251] M. X. Weng and J. A. Ventura. A doubling procedure for constructing minimal broadcast networks, *Telecomm. Syst.*, 3:259–293, 1995.

[252] D. B. Wilson. Generating Random Spanning Trees More Quickly than the Cover Time, *STOC 1996*, pp. 296–303, 1996.

[253] E. W. Zegura, K. Calvert, and S. Bhattacharjee. How to model an internetwork, *IEEE INFOCOM*, San Francisco, CA, USA, 1996.

[254] Li Zhang. Fault-Tolerant Meshes with Small Degree, *IEEE Transactions on Computers*, 51(5):553–560, 2002.

[255] J.-G. Zhou and K.-M. Zhang. A minimum broadcast graph on 26 vertices, *Appl. Math. Lett.*, 14:1023–1026, 2001.

[256] J. Xiao and X. Wang. A research on minimum broadcast graphs, *Chinese J. Computers*, 11:99–105, 1988.

[257] J.M. Xu. *Topological structure and analysis of interconnection networks*, Kluwer Academic Publishers, Dordrecht/Boston/London, 2001.

[258] X. Xu. Broadcast Networks of Odd Size and Minimum Broadcast Network on 127 Nodes, *Master thesis*, Concordia University, 2003.