Universidad de Cantabria

Departamento de Ingeniería Informática y Electrónica

Tesis Doctoral

Propiedades de Distancia y Simetría en Grafos y su Aplicación a Redes de Interconexión y Códigos

Presentada por Cristóbal Camarero Coterillo.

Dirigida por Carmen Martínez Fernández y Ramón Beivide Palacio.

Santander, Marzo de 2015.

Agradecimientos

Agradezco a mis directores Carmen Martínez y Ramón Beivide por su apoyo durante todos estos años. Esta tesis a sido financiada por la Universidad de Cantabria en 2011 desde Junio hasta Noviembre y por el ministerio de España desde entonces hasta Mayo de 2015 bajo las ayudas para la formación del profesorado universitario, referencia AP2010-4900.

Resumen

La topología de una red de interconexión es el grafo de sus elementos encaminadores o *routers*. Es decir, los vértices del grafo representan *routers* y las aristas representan conexiones. Las topologías actualmente usadas en los grandes supercomputadores se pueden dividir en dos familias: las que usan *routers* con grado moderado y las que usan *routers* de alto grado. El objetivo de esta tesis es proponer topologías para ambas familias que posean mejores propiedades que las actuales.

* * * * *

La familia de topologías de grado moderado consta de toros de entre 3 y 6 dimensiones. Entre las mayores máquinas existentes se encuentran el Cray XK7, el K computer y varios Blue Gene/Q. En esta tesis se proponen variantes de los toros que, con los mismos recursos, alcanzan mayor rendimiento. Una forma de mejorar las distancias en el toro es la introducción de *twists* en los enlaces periféricos; su generalización lleva a la definición de *lattice graphs*, que abarcan realmente todos los grafos de Cayley sobre grupos Abelianos. En el caso tridimensional, uno puede fijarse en las *lattices* cúbicas que se usan en cristalografía. Al usarse éstas como base de un *lattice graph* se obtienen buenas propiedades, tales como una pequeña distancia media y muchas simetrías. Dichos cristales pueden generalizarse a cualquier dimensión; esto es importante, ya que existen máquinas cuya topología es un toro de 6 dimensiones. Entre las propiedades de estos cristales destaca la simetría; se comprueba que los *lattice graphs* simétricos tienen mejor rendimiento que los asimétricos. En contraposición, existen bastantes implementaciones con lados de tamaño diferente, es decir, asimétricos, que obtendrían grandes mejoras con la adopción de estas técnicas.

* * * * *

Dentro de la familia de topologías de *routers* de alto grado más usadas se encuentran los *fat trees*, las redes de Clos y más recientemente, las dragonflies. En esta tesis nos hemos centrado en está última. Las dragonflies se definen jerárquicamente como grupos de *routers* que están fuertemente conectados dentro del grupo y entre el conjunto de todos los grupos. En concreto, cada grupo forma un grafo completo de *routers* y la red total es un grafo completo de grupos. Esta definición hace muy fácil su realización física en armarios. Tienen unas propiedades de distancia bastante buenas, con lo que a pesar de que se conocen familias con mejores distancias, su simplicidad de uso y su bajo coste las hace muy atractivas. Una topología más clásica, que se ha usado en redes de interconexión, es el grafo de Hamming. En esta memoria se explica cómo un grafo de Hamming puede verse como una dragonfly con gran trunking global y como ciertas propiedades de los grafos de Hamming pueden extenderse a otras dragonflies. En concreto, en el grafo de Hamming es muy fácil obtener un encaminamiento libre de deadlock simplemente fijando un orden entre las dimensiones. En dragonflies con cierto trunking global puede obtenerse un encaminamiento similar.

* * * * *

El problema de buscar *lattice graphs* con propiedades de distancia óptimas resulta ser equivalente al problema de encontrar buenos códigos sobre el espacio de Lee. Así que algunos resultados se vuelven mucho más interesantes desde una perspectiva de teoría de códigos. En esta tesis se construyen varios códigos cuasi-perfectos, que se pueden por tanto ver como topologías casi óptimas. Aparece el problema de que existen sólo para cardinales que sean cuadrados de primos, lo que dificulta un uso claro como redes de interconexión. Existe una conjetura por Golomb y Welch que dice que no existen códigos perfectos para radio mayor o igual que 2 y dimensión mayor o igual que 3 y los últimos resultados para dimensiones grandes tienen ya más de 30 años. En esta tesis se construyen códigos cuasi-perfectos para dimensiones arbitrariamente grandes que alcanzan la mitad de la densidad de los posibles códigos perfectos.

University of Cantabria

Department of Computer Science and Electronics

Doctoral Thesis

Distance and Symmetry Properties of Graphs and their Application to Interconnection Networks and Codes

Presented by Cristóbal Camarero Coterillo.

Advised by Carmen Martínez Fernández and Ramón Beivide Palacio.

Santander, March 2015.

Abstract

The topology of a interconnection network is the graph of its routers. Thus, the vertices of the graph model the routers and the edges model the network links. The topologies that are being currently used in large supercomputers can be classified into two families: the ones that use routers with moderate radix and the ones using high-radix routers. The objective of this thesis is to define topologies for both families that exhibit better properties than the actual ones.

* * * * *

The family of moderate radix topologies consists on tori with among 3 and 6 dimensions. The largest machines based on tori are the Cray XK7, the K computer and a variety of Blue Gene/Q systems. In this thesis several variants of tori are proposed that can achieve greater performance with the same cost. An idea to improve torus' distance is the introduction of twists in the peripheral links; the generalization of this idea brings the definition of lattice graphs, which actually contains all Cayley graphs over Abelian groups. In the three-dimensional case, special attention can be devoted to the cubic lattices used in crystallography. When these are used to define lattice graphs, good properties are obtained, such as small average distance and many symmetries. In fact, they can be generalized to any number of dimensions, which is necessary to match the 6 dimensions of some actual machines. From the properties of these crystal lattice graphs the symmetry is very notable; it is obtained that symmetric lattice graphs have better performance than the asymmetric ones. This confronts with many implementations of tori with mixed-radix, which would obtain great improvements by adopting these techniques.

* * * * *

Among the most used topologies for the family of high-radix routers there are the fat-trees, the Clos networks, and more recently, the dragonfly networks. This thesis focuses on dragonfly networks. Dragonflies are defined in a hierarchic way as groups of routers with a strong connectivity inside the group and among the collection of all the groups. More specifically, each group is a complete graph of routers and the whole network is a complete graph of those groups. This definition makes very easy to implement them in racks. Although there are known families of graphs with better distance properties, the simplicity of use and low cost of the dragonflies makes them very attractive. A very classic topology, which has been proposed for interconnection networks, is the Hamming graph. In this thesis, it is explained how Hamming graphs can be seen as a dragonfly with large global trunking and that some properties of the Hamming graphs can be extrapolated to dragonflies. Specifically, the Hamming graph has an easy deadlock-free routing that consists simply in fixing an order among the dimensions. In dragonflies with some global trunking a similar routing is shown to be possible.

* * * * *

The problem of finding lattice graphs with optimal distance properties is actually equivalent to the problem of finding good codes over the Lee space. This makes some results more attractive when seen from the coding theory perspective. In this thesis several quasi-perfect codes are built, which can then be seen as nearly optimal lattice graphs. They only exist when the number of vertices is the squares of a prime, which is an obstacle to their implementation. A conjecture was made by Golomb and Welch stating that there are not perfect codes for radius greater or equal to 2 and dimension greater or equal to 3, and the last results concerning large dimensions were made more than 30 years ago. In this thesis quasi-perfect codes are built for arbitrarily large dimensions that reach half the density of the density of potential perfect Lee codes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The purpose of this introductory chapter is to establish the context of this thesis, to outline the results it contains and to introduce basic concepts to be used along this thesis.

For that, Section 1.1 tries to motivate the study by showing the importance of the topology of the interconnection networks. Then, Section 1.2 establishes some important problems that this thesis has tried to solve. In Section 1.3 a collection of works that are related to these problems are considered. Later, Section 1.4 makes a summary of the results. Afterwards, Section 1.5 outlines the contents of each chapter. Finally, Section 1.6 introduces the notation and basic theory that is used along the thesis.

1.1 Motivation

Since the introduction of packet switching by P. Baran [Bar64], interconnection networks have played an increasingly important role in Computer Science and Engineering. Computer networks were initially used for defense applications in which reliability and availability were fundamental issues. Notwithstanding, interconnection networks quickly became popular on the fields of distributed systems and High-Performance Computing (HPC). Nowadays, most computer systems exploit the concept of parallelism and consequently, networks have become strategic and pervasive.

A look to the interconnection networks of top current supercomputers shows a dichotomy: in one group there are the Cray XK7, the K computer and several Blue Gene/Q, whose topologies are tori between 3 and 6 dimensions—they are topologies with *moderate degree*; the other group contains machines as Tianhe-2, the Cray XC30 (a.k.a. CASCADE) and Stampede, using high-radix routers. The second group is classically composed of Clos networks[1] (*e.g.,* Tianhe-2 [LPW+15] and Stampede). The topology of Cray XC30 is based on dragonflies, which is a more modern approach. This thesis discusses both groups of machines, although more attention is paid to the ones with moderate degree.

Next, some basic concepts are introduced in order to facilitate further discussion. An *interconnection network* is defined by its topology, routing, flow control and deadlock avoidance mechanisms, along with other technological aspects such as the used media and router design. However, very often the *topology* and *network* terms are used interchangeably in the literature. The *topology* of a network defines how the different routers are connected. An *indirect* topology (or network) employs transit routers, to which no computing node is connected. Typical examples of these are the tree and folded Clos

[1] Although they call them *fat-trees*. But that differs from the original meaning of fat-tree, which is a tree where the links are ticker towards the root [Lei85].

topologies. Conversely, a *direct* topology does not employ transit routers, so each network router has one or more compute nodes directly connected to it. When all the network links are point-to-point, as often occurs today in HPC and datacenter networks, the topology can be completely defined using a graph. The graph degree, Δ, is determined by the radix of the network routers, not considering the connections to the compute nodes. Frequent direct topologies proposed for HPC and datacenters are those based on meshes, tori, dragonflies [KDSA08] and Hamming graphs (also known as flattened butterflies [KDA07]). Some important issues of the network topology are its *scalability*, to be able to make large machines—which in the graph theory literature is manifested as the *degree-diameter problem*—and its *symmetry* properties—vertex- and edge-transitivity—which guarantee a *balanced* resource usage, as well as the simplicity of its deadlock avoidance mechanisms.

Before the use of tori, ring topologies have been widely employed in different domains. Token Ring [IEE89], MetaRing [CO93] and FDDI [IEE91] are good examples for local area networks. More recently, on the VLSI domain, several current microprocessors [SKS+11, SCS+08], use ring networks to interconnect their functional units. Although rings are cheap and symmetric, they exhibit poor reliability and performance. Hence, the idea to add connections to a base ring has been deeply studied and applied.

Two-dimensional tori are a natural evolution of rings. The torus is, together with the mesh, the most popular two-dimensional topology. However, first significant developments in parallel supercomputing were not based on tori. The SOLOMON, as described in a 1962 paper [SBM62], used a two-dimensional mesh but its successor, the ILLIAC IV described in [BBK+68], added wraparound links to the mesh to form a two-dimensional twisted torus. Such a twisted torus is, in fact, a circulant graph (specifically a chordal ring), in which the Hamiltonian embedded ring was used for control and command purposes. Notwithstanding, standard tori have became more popular than their twisted counterparts and several current supercomputers are tori-based [ABC+05, Cra, ASS09].

Powerful supercomputers such as Cray XK7, IBM BluGene/Q and K computers use moderate degree networks. The Cray employs a 3D torus whereas Blue Gene uses a 5D one [Cra, CEH+11]. The K computer employs small 3D meshes (that can also be seen as 4×3 tori) connected by a bigger 3D torus [ASS09]. All these topologies are mixed-radix tori, as they have dimensions of different sizes. For example, a configuration for a Cray Jaguar can be $25 \times 32 \times 16$ and a Blue Gene configuration $16 \times 16 \times 16 \times 12 \times 2$. The topology of the $88,128$-node K computer installed at Riken, is equivalent to a $17 \times 18 \times 24$ torus connecting 3D meshes of 12 nodes. Mixed-radix tori are not edge-symmetric, which can lead to unbalanced use of their network links. However, these big systems are typically divided into smaller partitions, which enable them to be used by multiple users. Hence, providing symmetry is, at least in typical network partitions, an advisable design goal.

Other two-dimensional topologies based on rings have been explored time ago as alternatives to tori. For example, the diagonal toroidal mesh, which is isomorphic to the Kronecker product of two cycles, has been considered as a substitute to their Cartesian product—*i.e.,* to tori—claiming for some advantages, specially in the case of mixed-radix topologies [TP94, Pea96]. Other Cayley graphs, such as circulants, chordal rings and Gaussian networks have been previously studied, [FYAV87, BW85, MBS+08, LY10]. Some variations of the 3D tori are given in [CMV+10]. That work starts with mixed-radix tori, then by introducing twists the distances are reduced and certain symmetries are obtained.

Recently, dragonfly networks have appeared [KDSA08]. These topologies have great connectivity, low cost, and they enjoy a natural layout into the physical racks containing the compute boards. This explains the celerity with one of its variants has been used as

the topology of one of the top 10 supercomputers—the CASCADE computer.

1.2 Goals

The general objective of this thesis is to find good network topologies for HPC systems. For the case of moderate degree networks, the only topologies in use are tori; indeed, in most cases they are unbalanced mixed-radix tori. Their twisted alternatives have been considered only for two and three dimensions. Thus, an important problem is to make twists in tori of arbitrary number of dimensions, or at least, up to the 6 dimensions that are currently used in real supercomputers. Moreover, those mixed-radix torus implementations make unbalanced use of the links, which reduce performance. Hence, these multidimensional twists should be done in a way that balances the load. Others aspects that have been addressed are routing algorithms, packaging and upgrading.

Respect to the high-degree networks, it is notable the recent development of the promising dragonfly topologies. It was preceded by the reinvention of the Hamming graph with the name of flattened butterfly. The Hamming graph has been deeply studied, under many different points of view. On the contrary, dragonflies are novel and many aspects of them can be studied. Furthermore, Hamming graphs and dragonflies possess several similarities that are worth of study.

1.3 Related Work

For moderate degree topologies, many works have introduced twists in tori; some of them explicitly calling them twists. The first use of twisted tori we know was in the implementation of the Illiac IV computer [BBK+68]. Later, twisted tori were proposed for design of VLSI systems [Seq81, Mar81]. In [TP94], 2D tori with diagonal links were shown to have better distance properties than normal tori. Afterwards, Pearlmutter in [Pea96] realized that these tori with diagonal links can be seen as twisted tori with orthogonal links. The optimal twisted tori for each diameter was found in [BHBA91]. In [MBS+08] Gaussian graphs were defined over the ring of Gaussian integers; they can be seen as bidimensional twisted tori where the size and twist depend on the generator. In [CMV+10] some 2D and 3D mixed-radix tori were studied whose sides followed the proportions 2:1 and 2:1:1, respectively. They include some results on the implications of symmetry, although only a few topologies were considered. In [Fio95] circulant graphs were generalized to more dimensions, providing most of the mathematical background used in our work of lattice graphs (Chapter 2).

Examples of high degree networks are fat trees, Clos networks, Hamming graphs and dragonflies. Only the last two are studied in this thesis. The use of Hamming graphs as interconnection network began in 1984 with [BA84] under the name of generalized hypercubes—the binary hypercubes were already been used for some time—and they have been reinvented several times since then. Bhuyan and Agrawal proposed in [BA84] the Hamming graph with the name of *generalized hypercube*. They analyzed mostly distance properties and latencies, together with some comments about routing and fault tolerance. Later, LaForge *et al.* studied extensively the fault tolerance of Hamming graphs, using the name of K-cubes [LKF03]. Kim *et al.* found the Hamming graph as the result of applying a flattening operation to the butterfly network and they give it the name of *flattened butterfly* [KDA07]. Dragonfly networks were recently introduced by Kim *et*

al. [KDSA08]. Many works have proposed routing mechanisms for dragonfly networks to tolerate adverse traffic patterns, taking into account possible transient traffic and implementation costs [JKD09, GVB$^+$12b, GVB$^+$13c]. Industrial implementations of the dragonfly topology have been the IBM PERCS [AAC$^+$10] and Cray Cascade [FBR$^+$12].

1.4 Results

The results obtained in this thesis can be organized in three domains: i) the ones for moderate degree, ii) the ones for high-degree and iii) some results in coding theory.

i) For moderate degree networks the problem has been attacked using lattices and linear algebra.

- *Lattice graphs* are defined, which englobe multidimensional tori with multiple twists. This family hence generalizes tori and multitude of their variants.
- The impact of symmetry on the performance of lattice graphs is proved to be very high, in accordance with [CMV$^+$10]. This has been done analytically and confirmed by experimentation.
- The symmetric lattice graphs of dimension 2 and 3 have been characterized; they are still a large family that include Gaussian graphs, Kronecker product of cycles and the three-dimensional proposal of [CMV$^+$10] (the prismatic doubly twisted torus) among many others.
- The lattice graphs based on the cubic lattices inherited from crystallography have been deeply studied. These crystal lattice graphs are shown to be 3D symmetric graphs and to have very good distance properties. Moreover, analogous lattice graphs are defined for any dimension.
- The diameter and average distance have been exactly determined for 2D lattice graphs and for 3D crystal graphs by different methods.
- Lattice graphs have a matricial representation, which gives a compact way to express characterizations and an immediate way to find natural subgraphs. This has allowed to define a lift operator to construct lattice graphs containing the desired lattice graphs as subgraphs.
- Several routing algorithms have been yield. For 2D lattice graphs it is given a routing algorithm based on lattice reduction that can be understood in a simple geometrical way by means of tessellations. For the special case of the rectangular twisted torus (RTT) an algorithm is given, which is more elegant than the general one for 2D lattice graphs. For greater dimension, a general hierarchical algorithm is established that has better complexity for the case of crystal lattice graphs than the best general known algorithms.
- For the physical deployment of lattice based networks, two strategies are given. One is based on the generic approach that Cray uses in the layout of its tori. The other one is based on the Blue Gene technology, where there is additional hardware to make tori partitions.
- Simulations show that lattice graphs outperform some tori currently in use.

ii) For networks of high degree our focus has been on the recent dragonfly topologies.

- Dragonflies has been defined in the literature in an informal way; some effort has been done in this work to clear up many aspects.
- Global trunking—several global links between every pair of groups of routers—in dragonflies is studied. They are shown to include the Hamming graphs as the extremal case of a dragonfly with maximum global trunking.
- The global trunking alternatives are thoroughly studied, giving the conditions to make a balanced use of links.
- Several possible arrangements for global links are studied, coming to the conclusion that they have a very small impact on performance. This contrasts with the case of lattice graphs, where the symmetry is very important. However, symmetries can allow for specific mechanisms, like some routing algorithms.
- A deadlock-free routing is developed that does not require the use virtual channels but a few symmetries in the topology are needed. Experiments show that it has similar performance that known routing algorithms when given the same resources.
- Finally, some remarks are given for three-level dragonflies. Note that three are enough levels to reach astronomical amounts of compute nodes.

iii) There have been some interesting results on coding theory that derivate from our study of lattice graphs.

- It is described how lattice-graphs and linear Lee-codes are related. A graph with good distance properties will give a good linear Lee code with good correction and covering properties and *vice versa*.
- A characterization is done of t-quasi-perfect codes given by ideals of Gaussian and Eisenstein–Jacobi integers. Decoding algorithms are given and a comparison is made with other 2D t-quasi-perfect Lee codes.
- 2-quasi-perfect Lee codes are built for arbitrarily large dimension. This result has importance in coding theory, where the Golomb–Welch conjecture says that there are not perfect codes and the ones obtained in this thesis are very close to be perfect. Furthermore, the graphs associated to these codes are Ramanujan graphs; these graphs have found uses in many different areas, so the impact of the result could have more extension.

1.5 Organization

Many of the chapters and sections of this thesis are adaptations of articles published on journals and conferences. A list of these publications can be found at the end of the thesis, in Section 6.3.

Chapter 2 presents the results on lattice graphs, which correspond to moderate degree network models. It starts with the necessary linear algebra to define lattice graphs and stablishes its fundamental properties. Then, the crystal lattice graphs and its derivatives are studied. The problem of routing is studied both in general and in specific cases. Some ways to make the physical layout are devised.

Chapter 3 presents the results on dragonfly networks with global trunking, which correspond to a model for high degree networks. This chapter begins with an introduction

to Hamming graphs, its properties and how they were rediscovered several times with different names. Then the dragonfly topology is defined in a more precise way, introducing the concept of *global link arrangements*. Global trunking is introduced, which allows to see the Hamming graph as a dragonfly with large global trunking. This fact hints to the great size of this family of graphs and motivates to study if some properties of the Hamming graph can be translated to general dragonflies. Afterwards, it is seen that indeed the Hamming graphs allow for deadlock-free routing without VCs and dragonflies with some global trunking allow for similar routings. Finally, some remarks on 3-level dragonflies are done.

Chapter 4 shows the relation between lattice graphs and linear Lee codes. Then, quasi-perfect codes are built first for the bidimensional case and later 2-quasi-perfect codes for large dimensions. The latter are related to a conjecture by Golomb and Welch that says that perfect Lee codes do not exist for high dimension. The codes obtained are very close to be perfect.

Chapter 5 explains the evaluation infrastructure and the experiments carried out. These experiments are: study of symmetry in lattice networks, mapping applications on lattice networks, and study of symmetry in dragonfly networks and evaluation of the deadlock-free routing for dragonflies with global trunking.

Chapter 6 closes the document giving some conclusions about the realized work, stating some lines of future work and listing the publications written during this time.

Finally, the Appendix A contains the proof of the characterization of symmetric lattice graphs for dimensions 2 and 3.

1.6 Fundamentals on Graphs and Networks

This chapter introduces basic concepts to be used along this thesis. Subsection 1.6.1 presents Cayley graphs over Abelian groups and gives some examples as Hamming graphs and circulant graphs. Symmetry aspects are introduced in Subsection 1.6.2, defining the group of automorphisms and giving some notes of symmetry in Cayley graphs. Subsection 1.6.3 presents the degree-diameter problem together with graph expansion properties and their impact on the performance of the networks they model. Finally, Subsection 1.6.4 describes the routing problem, with the concepts of routing record, routing by tables, deadlock-freedom, routing randomization and head of line blocking.

Before beginning with those concepts some notation must be introduced.

Notation 1.6.1. *The following notation will be used throughout this thesis:*

- *Lower case letters can denote integers (a, b, ...), vertices, or elements of groups.*
- *Bold font denotes integer column vectors:* $\mathbf{v}$, $\mathbf{w}$, ...
- *Capitals correspond to integral matrices (M, P, ...), graphs (G,* C_n, K_n*) and sometimes sets, groups and rings.*
- $\mathbf{e}_i$ *denotes the vector with a* 1 *in its i-th component and 0 otherwise.*
- *If G is a graph, then* $V(G)$ *is its set of vertices and* $E(G)$ *the set of edges.*
- *Graph isomorphism is denoted by* $\cong$.

- *$\mathbb{Z}^n$ denotes tuples of length n over the integers. The set of integral matrices of m rows and n columns is denoted by $\mathbb{Z}^{m\times n}$.*

- *$\arg\min_{x\in S}(f(x))$ or $\arg\min\{f(x) \mid x \in S\}$ is defined as the element x in S that minimizes the value of $f(x)$.*

- *Congruence of x and y modulo n is denoted by $x \equiv y \pmod n$, and the congruence class of x by $(x \pmod n)$. This is explicitly discriminated from the remainder of x by n using Euclidean division, denoted $\mathrm{rem}(x, n)$, this is, $0 \leq \mathrm{rem}(x, n) < |n|$ and $x \equiv \mathrm{rem}(x, n) \pmod n$.*

- *For a real number r, $[r]$ denotes the closest integer, $\lfloor r \rfloor$ denotes the greatest integer that is less or equal than r and $\lceil r \rceil$ denotes the lowest integer that is greater or equal than r.*

1.6.1 Cayley Graphs

A *graph* G is defined by a finite set of *vertices* $V(G)$ and a set of unordered pairs of vertices $E(G)$ that is called its *edge set*. An edge $e = \{x, y\} \in E(G)$ is said to *connect* x to y; then x and y are called *adjacent* vertices or *neighbours*; and e is *incident* on both x and y. The *degree* of a vertex is the number of its neighbours. If all vertices of a graph G have degree Δ then G is a Δ-regular graph. Sometimes, to avoid confusions in some places, the explicit notation $\deg(G)$ will be used. Graphs are used as a model for the topologies of interconnection networks, where routers are represented by vertices and links by edges. The compute nodes are abstracted and the constant Δ_0 will be used to represent the number of compute nodes attached to every router.

A *walk* in a graph G is a list of vertices $x_1, x_2, \ldots, x_n$ such that $\{x_i, x_{i+1}\} \in E(G)$ for $i \in \{1, \ldots, n-1\}$; x_1 and x_n are the endpoints of the walk. The *length* of a walk w is the number of edges it traverses; hence, for $w = x_1, \ldots, x_n$ the length of w is $|w| = n - 1$. A *path* is a walk composed of distinct vertices, *i.e.*, $x_i \neq x_j$ for $i \neq j$. A graph G is *connected* is for any vertices $x, y \in V(G)$ there is a path (or equivalently a walk) with x and y as its endpoints.

The *distance* between two vertices v, w in a graph is defined as the length of the shortest path between them; it will be denoted as $D(v, w)$. The *diameter* (respectively *average distance*) of a graph is the greatest (resp. average) graph distance along all pairs of different vertices. The diameter is typically denoted by k and the average distance by $\bar{k}$.

The order of an element x of a group Γ is the minimum positive integer n such that x^n is the neutral element of Γ. In symbols $ord(x) = \min\{n \in \mathbb{Z} \mid n > 0 \text{ and } x^n = 1\}$.

Given a ring $\mathbb{K}$ (as $\mathbb{Z}$ or $\mathbb{Z}[i]$), a pair of elements $x, y \in \mathbb{K}$ are said congruent modulo other element $z \in \mathbb{K}$, denoted $x \equiv y \pmod z$, if there is $q \in \mathbb{K}$ such that $x - y = zq$. Then, the quotient $\frac{\mathbb{K}}{x\mathbb{K}}$ represents the subring of $\mathbb{K}$ where two elements are identified if they are the same modulo x.

The concept of Cayley graph is key in this thesis.

Definition 1.6.2. *The Cayley graph over a group Γ and adjacency set $A \subset \Gamma$ is defined as the graph $Cay(\Gamma; A)$ with vertex set $V = \Gamma$ and edges*

$$E = \{(v, v + g) \mid v \in \Gamma, g \in A\}.$$

The considered groups will always be Abelian groups; thus the group operation will be $+$ with neutral element 0. The set of hops A cannot contain 0, since it would imply a loop in every vertex, and must satisfy $-A = A$ for the edges to be undirected.

Cayley graphs over $\mathbb{Z}_n$ are called *circulant graphs*. The name comes from the fact that they are exactly the graphs whose adjacency matrix is a circulant matrix—each row is a rotation of the preceding one by one entry. The *cycle* of length n, $C_n = Cay(\mathbb{Z}_n; \{\pm 1\})$ is a common circulant graph. Another important circulant graph is the complete graph $K_n = Cay(\mathbb{Z}_n; \mathbb{Z}_n \setminus \{0\})$.

The *Cartesian product* of two graphs G_1, G_2 is denoted by $G_1 \square G_2$. It is defined by $V(G_1 \square G_2) = V(G_1) \times V(G_2)$ and

$$E(G_1 \square G_2) = \big\{\{(x_1, y_1), (x_2, y_2)\} \mid (x_1 = x_2 \wedge \{y_1, y_2\} \in E(G_2)) \vee (y_1 = y_2 \wedge \{x_1, x_2\} \in E(G_1))\big\}.$$

Weichsel [Wei62] defined the Kronecker product of two graphs as the graph having as adjacency matrix the Kronecker product of the respective adjacency matrices of their factors. Weichsel also gave another equivalent definition, which was rewritten in [IK00] in a more natural way:

Definition 1.6.3. *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. Their Kronecker product $G_1 \times G_2$ is defined as the graph $G = (V, E)$ such that the set of vertices is the Cartesian product $V = V_1 \times V_2$ and the set of edges is $E = \big\{\big((v_1, v_2), (w_1, w_2)\big) \mid (v_1, w_1) \in E_1$ and $(v_2, w_2) \in E_2\big\}$.*

Therefore, each vertex of the product is related to a vertex in each of the original graphs, and two vertices are connected in the Kronecker product if their corresponding vertices are connected in its factors.

Note that both the Cartesian product and the Kronecker product are commutative and associative up to graph isomorphism.

The *n-dimensional torus* graph of sides $a_1, \ldots, a_n$ denoted by $\mathrm{T}(a_1, \ldots, a_n)$ is defined as $C_{a_1} \square \cdots \square C_{a_n}$. Similarly, the *Hamming graph* of arities $a_1, \ldots, a_n$ is defined as $K_{a_1} \square \cdots \square K_{a_n}$. Its name comes from the fact that its graph distance coincides with the Hamming distance. We will make special emphasis on this graph in Chapter 3.

Both the Cartesian and the Kronecker products of two Cayley graphs are Cayley graphs. And if the original graphs are Cayley graphs over Abelian groups then the resulting graph is again a Cayley graph over an Abelian group.

If α is a Gaussian integer, then $G = Cay(\frac{\mathbb{Z}[i]}{\alpha\mathbb{Z}[i]}; \{1, -1, i, -i\})$ is called a *Gaussian graph* [MBS+08]. The analogous graph can be made for the Eisenstein–Jacobi integers. Let ω be a root of $x^3 - 1$ other than 1. Then $\mathbb{Z}[\omega]$ is the ring of Eisenstein–Jacobi integers and for any $\alpha \in \mathbb{Z}[\omega]$, the graph $G = Cay(\frac{\mathbb{Z}[i]}{\alpha\mathbb{Z}[i]}; \{\pm 1, \pm\omega, \pm\omega^2\})$ is called an *Eisenstein–Jacobi graph* [MSBG08]. Gaussian graphs were generalized in [MBG09] as graphs over any ring from the Cayley–Dickson constructions. Gaussian graphs will be mentioned several times in Chapter 2. Later, in Chapter 4 there will be extensive work on Gaussian and Eisenstein–Jacobi graphs.

1.6.2 Symmetry

Given two graphs G and H, a *graph isomorphism* from G to H is a bijection $\phi : V(G) \to V(H)$ such that there is an edge $\{x, y\} \in E(G)$ if and only if there is the edge $\{\phi(x), \phi(y)\} \in E(H)$. When $G = H$ then ϕ is a *graph automorphism*.

Then, a graph G is said *vertex-transitive* (or *vertex-symmetric*) if for each pair of vertices $(x, y) \in V(G)$, there is an automorphism ϕ of G such that $\phi(x) = y$. In addition, G is said *edge-transitive* (or *edge-symmetric*) if for each pair of edges $(\{x_1, x_2\}, \{y_1, y_2\}) \in E(G)$, there is an automorphism ϕ of G such that $\phi(\{x_1, x_2\}) = \{\phi(x_1), \phi(x_2)\} = \{y_1, y_2\}$. Finally, G is said to be *symmetric* when it is both vertex-transitive and edge-transitive.

Clearly, vertex-transitivity implies that all vertices have the same degree.

All Cayley graphs are vertex-transitive [AK89]. This is immediate upon realizing that for any $a, b \in V(G)$, $\phi : x \mapsto b - a + x$ is an automorphism that maps a into b.

In a vertex-transitive graph all vertices have the same distance distribution. Thus, the average distance is

$$\bar{k} = \frac{1}{|V(G)| - 1} \sum_{v \in V(G)} D(o, v)$$

and the diameter is $k = \max\{D(o, v) \mid v \in V(G)\}$, where o is any vertex chosen *a priori*; typically the neutral element in the case of Cayley graphs.

On the case of Cayley graphs it is also important the concept of *linear automorphism*, which is an automorphism ϕ satisfying $\phi(x + y) = \phi(x) + \phi(y)$ for any vertices x, y.

The group of automorphisms of a graph G is denoted by $Aut(G)$, and the group of linear automorphisms by $LAut(G)$. If $LAut(G)$ acts transitively on the vertices and edges then G is *linearly symmetric*.

Symmetry can have a great impact on performance. Topologies of interconnection networks of general purpose must perform well under uniform traffic, since it determines the worst case performance (when using the best routing algorithm, see Subsection 1.6.4). Thus, let us count for every edge e the number of minimal paths that traverse it[2]. If this amount is the same for every edge, then the topology is *edge balanced*; otherwise, the edge with greatest value will be saturated first (with uniform traffic) and the other edges will not reach their full capacity. Clearly, edge-transitivity implies to be perfectly edge balanced. For the family of lattice graphs studied in Chapter 2, edge-transitivity will have a great impact on performance. In contrast, dragonflies (studied in Chapter 3) have a simple condition to be very well edge-balanced, and looking for edge-transitivity does not provide an increase in performance; symmetries can still be important to have access to other mechanisms as it will be shown.

Some of the previously mentioned Cayley graphs are edge-transitive. The Gaussian graphs are edge-transitive by action of the automorphism $\phi : x \mapsto xi$ and the Eisenstein–Jacobi graphs by action of the automorphism $\phi : x \mapsto x\omega$. Cycles and complete graphs are trivially symmetric. Powers (Cartesian or Kronecker) of graphs inherit the symmetry of the base graph.

1.6.3 Degree Diameter Problem

The *degree-diameter*, or *d-k-problem*, consists in finding a graph G for a given degree Δ and diameter k with the maximum number of vertices $N(\Delta, k)$. An upper bound in $N(\Delta, k)$ is the Moore bound, of value [HS60]

$$M(\Delta, k) = \frac{\Delta(\Delta - 1)^k - 2}{\Delta - 2}.$$

[2]In some cases there are several minimal path between two endpoints x, y. To solve it, simply increase the count for e by $\frac{\#\ x,y\text{-minimal paths traversing } e}{\#\ x,y\text{-minimal paths}}$.

Graphs reaching this bound are called *Moore graphs*. Optimizing the degree-diameter problem provides the largest possible network with optimal performance under uniform traffic. However, practical constraints such as regularity of the topology, *fine-grain scalability*[3], convenient layouts and cable length, number of computing nodes per router (or *concentration level*), routing mechanisms and performance under alternative traffic patterns make that other topologies with a lower amount of network nodes become more attractive.

The Moore bound sets a limit on the degree-diameter problem. A thorough survey of the problem and Moore graphs can be found in [MŠ13]. For diameter $k = 1$ the complete graphs $K_{\Delta+1}$ attain the bound. Their simplicity and existence for any size make them very interesting; however they are subject to technological constraints given the large degree necessary to reach a high number of nodes. For diameter $k = 2$ there are only 2 or 3 Moore graphs [HS60]: the Petersen graph ($\Delta = 3$, $N = 10$), the Hoffman–Singleton graph ($\Delta = 7$, $N = 50$) and an hypothetical graph with $\Delta = 57$ and $N = 3250$ whose existence is still an open problem. This sporadic existence of Moore graphs complicates scalability, being very difficult to decide which topology to use for a given network size. The problem can be relaxed by considering only the asymptotic behaviour. This relaxed problem consists in finding, for every diameter k, an infinite family of graphs with about Δ^k vertices. Such graphs exist for $k = 1$ (complete graphs), $k = 2$ [Bro66, BBC13], $k = 3$ and $k = 5$ [Del85]. They are conjectured to exist for any diameter, but even the best general bounds are exponential in k. The work in [BBC13] seems to be the first to propose one of these families as interconnection networks, but fails to address many practical problems. More recently, Besta and Hoefler [BH14] have studied some known good families of graphs as topologies for interconnection networks, considering partially aspects as cost, layout, energy and oversubscription.

For Cayley graphs over Abelian groups of diameter 2 there is an upper bound of $\frac{1}{2}\Delta^2 + \Delta + 1$ vertices; the current best construction inside this family is the given in [MŠŠ12], which achieves $\frac{3}{8}(\Delta^2 - 4)$ vertices, about $\frac{3}{4}$ of the bound.

The degree-diameter problem can be useful to state if a topology has good *scalability*—maintaining good performance for large amounts of compute nodes. However it only gives a partial idea of the performance, so other measures are necessary. In networking literature there is a commonly used parameter called the *bisection bandwidth* [Tho79]. Take a graph G, then consider time measured in cycles and communication units called phits. Every phit has a vertex as target destination and the network tries to move the phits from their origin to their destination. Each cycle every vertex generates l phits and every edges allows the transmission of 1 phit. Now, assume that the origin and destination of phits are selected uniformly at random and partition $V(G)$ into two sets A and B with $|A| = |B|$ (or close), trying to minimize the number of edges from A into B, $|\delta(A, B)|$. Then, in the set A there is a generation of $l|A|$ phits each cycle. Thus, $\frac{l|A|\cdot|B|}{|V(G)|-1}$ phits will have as destination a vertex in set B, so they must cross $|\delta(A, B)|$ edges. Thus $\frac{l|A|\cdot|B|}{|V(G)|-1} \leq |\delta(A, B)|$, which is $l \leq \frac{|V(G)|-1}{|A|\cdot|B|}|\delta(A, B)| \approx \frac{|V(G)|}{4}|\delta(A, B)|$. The amount $|\delta(A, B)|$ is called the bisection bandwidth of a network, which, as it has been shown, limits the throughput of a network.

In spectral theory there is a similar concept called *edge expansion*. Let $\delta(S) =$

[3]Being able to construct topologies for many different sizes. For example, the binary hypercube requires 2^n routers, and hence, it is not fine-grain scalable.

$\delta(S, V(G) \setminus S)$. The edge expansion of G is

$$h(G) = \min_{0<|S|\leq\frac{|V(G)|}{2}} \frac{|\delta(S)|}{|S|}.$$

Except for constants and the naivety of the selection of the set, the expression is the same as the bisection bandwidth. This means that for uniform traffic good expanding topologies are desired for the networks. The name of expansion makes reference to that for any small set of vertices, the set of their neighbours is larger; other form to view this, is that a set of vertices expands when adding its neighbours. Thus, in a good expander graph, random walks quickly converge to a random vertex following an uniform distribution. These graphs have applications in many areas; for example to reduce the need for randomness in probabilistic algorithms and to find good error-correcting codes [Nie05]. Their relation to the d-k-problem can be seen by means of this rapid convergence of random walks. Note that in a Moore graph of diameter k, most vertices are at distance k from the origin. Then, a random walk of length k ends in a vertex at distance diameter with probability $(\frac{\Delta-1}{\Delta})^{k-1} \approx 1$. Thus, for large degree, random walks of length k finish with the same probability for the majority of vertices. There are a few vertices that have probability very different. It can be calculated that walks of length $2k+1$ are enough for all the vertices have almost the same probability. Therefore, random walks in graphs close to the Moore bound rapidly arrive to random vertices with uniform probability, so they must be good expander graphs.

The edge expansion is related to the second greatest eigenvalue λ of the adjacency matrix of G (the first one equals the degree) by the Cheeger inequalities:

$$\frac{1}{2}(\Delta - \lambda) \leq h(G) \leq \sqrt{2\Delta(\Delta - \lambda)}.$$

Thus, maximizing the edge expansion is similar to minimize λ. A known bound is $\lambda \geq 2\sqrt{\Delta - 1} - o(1)$, that is, for every $\varepsilon > 0$ there is only a finite number of exceptions to $\lambda \geq 2\sqrt{\Delta - 1} - \varepsilon$. A *Ramanujan graph* is a graph with $\lambda \leq 2\sqrt{\Delta - 1}$ [DSV03], *i.e.,* a graph that is optimal in the spectral expansion sense. In Chapter 4 some new Ramanujan graphs are presented. In that chapter, an infinite family of graphs of diameter 3 is built for which the first tens can be computed to be Ramanujan graphs. Unfortunately, we do not have yet proved that the infinitely many graphs of the family are Ramanujan.

It was shown in [CMV+10] a few examples of topologies in which the bisection bandwidth does not reflect performance accurately; since for any cut several minimal paths traverse the cut twice. For the case of edge-transitive graphs there is a simple accurate bound for the maximum throughput. As, under uniform traffic at rate l, l phits are injected into each node each cycle, there is a total of $l|V(G)|\bar{k}$ links being used each cycle. Any link can only transfer 2 phits—one in each way—each cycle, which implies that $l|V(G)|\bar{k} \leq 2|E(G)| = \Delta|V(G)|$. Thus, it is obtained that network throughput is bounded by

$$l \leq \Delta \bar{k}^{-1}. \tag{1.1}$$

The d-k-problem emerged from a graph-theoretical point of view, so it is not exactly the practical problem to be optimized. In the degree-diameter problem, one maximize the number of routers N given a degree Δ and diameter k. However, in practice, it can be preferable to instead maximize the number of compute nodes M given the router radix $R = \Delta_0 + \Delta$ and diameter k, where Δ_0 is the number of compute nodes attached to

every router. Nevertheless, in networks close to the Moore bound, these problems become equivalent. In first place, a network close to the Moore bound will be locally similar to a Δ-ary tree. Thus, such networks are close to be edge-balanced and by the above argument for edge-balanced networks, its load will be approximately the value of Equation (1.1), $\Delta\bar{k}^{-1}$. Furthermore, these graphs have high density and hence the number of vertices at distance k is much greater than the number of vertices at distance $k-1$; this implies that the average distance $\bar{k}$ is close to the diameter k. Therefore, it can be assumed that there are $\Delta_0 \approx \Delta k^{-1}$ compute nodes attached to every router. Then, the number of compute nodes M can be related to the radix R of the routers as

$$\frac{M}{R^{k+1}} = \frac{N\Delta_0}{(\Delta+\Delta_0)^{k+1}} \approx \frac{N\Delta k^{-1}}{(\Delta+\Delta k^{-1})^{k+1}} = \frac{N\Delta k^{-1}}{(\frac{k+1}{k}\Delta)^{k+1}} = \frac{N}{\Delta^k}\cdot\frac{1}{k(\frac{k+1}{k})^{k+1}}$$
$$= \frac{N}{\Delta^k}\cdot\frac{1}{k(1+k^{-1})^{k+1}}.$$

Thus, the two problems are related by a constant that depends only on the diameter k.

1.6.4 Routing

Packets are injected into queues of routers, which are the vertices of the topology, and they have another vertex as destination. Somehow it must be established a path for the packet to go to its destination. Some time ago it was common to establish the whole path at the time of the injection of the packet. However, that prevented from using those links for other communications during the transmission. Because of that, now messages are divided into small packets—which typically can fit into any buffer—and only allocate local resources. Specifically, in *virtual cut-through* when a packet moves, it allocates in the next buffer enough space for the whole packet. With this mechanism the decision of which path to take can be made every cycle. Here *cycle* means the smallest unit of time in which routers operate; the context will avoid any possible confusion with the cycle graph C_n.

Regardless whether the whole path is decided at the beginning of every cycle, the router must be able, given a destination vertex, to decide by which of its incident edges the packet will move. For this, we consider that every vertex has a *label* and then there is an algorithm that, given source and destination labels, determines at least the first edge of the route. For example, in the cycle C_n, two possible labellings are $\{0,1,\ldots,n-1\}$ and $\{-\lceil\frac{n-1}{2}\rceil, 1-\lceil\frac{n-1}{2}\rceil,\ldots,-1,0,1,\ldots,\lfloor\frac{n-1}{2}\rfloor-1,\lfloor\frac{n-1}{2}\rfloor\}$.

Usually the routes are made trying them to be as shortest as possible, although sometimes it is necessary to make a longer one to avoid problems like faulty components or congestion.

Classically, routing has been done with tables. In this approach every router contains a table indicating the next edge to use for each destination. For large graphs the size of the table can incur in high costs (which translates into chip area, energy consumption or time to fill the tables among others) and other approach can be preferred.

An algorithmic routing is possible for some topologies, which allows to avoid tables. An illustrative example is the mesh. Let G be a n-dimensional mesh of side a, for example $n=2$ and $a=8$. Given source $\mathbf{s}=(1,2)$ and destination $\mathbf{d}=(4,3)$ compute $\mathbf{r}=\mathbf{d}-\mathbf{s}=(3,1)$. Then if $\mathbf{r}=\mathbf{0}$ the destination is in the current router, otherwise—as is the case of the example—let i be such that $\mathbf{r}_i \neq 0$—take $i=0$ in the example. Then if the current router is $\mathbf{x}$, the edge $\{\mathbf{x}, \mathbf{x}+sign(r_i)\mathbf{e}_i\}$ is an edge in one of the minimum routes—in the example, as $\mathbf{r}_1=3$, there is a minimal path from $\mathbf{s}$ to $\mathbf{d}$ beginning with the

edge $\{\mathbf{s}, \mathbf{s}+\mathbf{e}_1\}$. Moreover, the vector $\mathbf{r}$ is called a *routing record* and it can be computed once and updated in every hop of the packet. It also allows for *adaptive routing*, since there are generally several values for i with $\mathbf{r}_i \neq 0$; or to select them in a specific order to obtain other properties.

Using minimal routes is optimal for uniform traffic and good enough for many others traffic patterns. However, there are always some traffic patterns for which using minimal routes results in very poor performance. As using minimal routes is optimal for uniform traffic, there is an elegant solution: *traffic randomization*. The scheme by Valiant [Val82] doubles and randomizes the traffic; for a packet with origin in x and destination in z a random router y is selected, then the packet is first routed from x to y and later from y to z. As the resulting traffic is uniform, the subroutes can be made minimal. Therefore, this solution guarantees a worst case of half the throughput than in uniform traffic.

Deadlock is a major problem in networking. If the edge selected by the routing algorithm have all its buffers filled, then the packet must wait. This can happen simultaneously for all the edges in a cycle, causing that none of them can ever move. The most immediate solution to this problem is the one taken in Ethernet, which consists in dropping (and resending) deadlocked packets. However, for latency-sensitive applications like the ones used in HPC this is not an option, and deadlocks must be avoided or resolved without packet loss.

To avoid deadlock, one option is to have deadlock-free routing algorithms. Consider the case of a n-dimensional mesh, as in the above example. The Dimension Order Routing (DOR) is the routing in which the least i such that $\mathbf{r}_i \neq 0$ is selected every cycle. This means that the edges in direction i can only wait to edges in a direction j such that $i \leq j$. As a path is deadlock free, the direction n is deadlock-free, and by induction the whole topology. This approach implies a restriction on routing and hence it can harm performance [GN92].

Another option to avoid deadlock is to use Virtual Channels (VCs). In every edge put several buffers and make the routing algorithm select between them. Now, to get a deadlock, there must be a cycle (a cycle graph, not the unit of time) in the graph of buffers, which can be avoided with enough VCs. A cycle can be made deadlock-free by using 2VCs. For general topologies it can be solved with a number of VCs equal to the diameter, a mechanism by Günther [Gün81], which consists in using the VC i for the i-th hop. This approach to solve the deadlock problem incurs in a greater cost depending on the number of VCs, both for the memory associated to it and the logic to control it.

In the case of the topology being a cycle there is also the option of using Bubble Routing [CBGV97]. By forbidding the packets in injection queues to move to transit queues when there is space for only one packet, it is guaranteed that there is at least space for one packet among all the transit queues of the cycle. This space moves continuously, which implies that all packets reach their destination. This mechanism can be extended to tori-like topologies (and indeed for any graph of the ones studied in Chapter 2). Thus, one way to make a torus deadlock-free is to adopt the bubble mechanism together with the DOR routing.

Furthermore, although virtual channels have a clear motivation by preventing deadlock they have also impact on performance. Specifically, they mitigate the *Head of Line Blocking* (HoLB) problem. The HoLB is the situation in which the first packet of a buffer cannot advance because it requires a link being used for another packet. While, there can be subsequent packets that will go for currently free links, but these are blocked by the first packet in the buffer. This situation is illustrated in Figure 1.1, which shows two

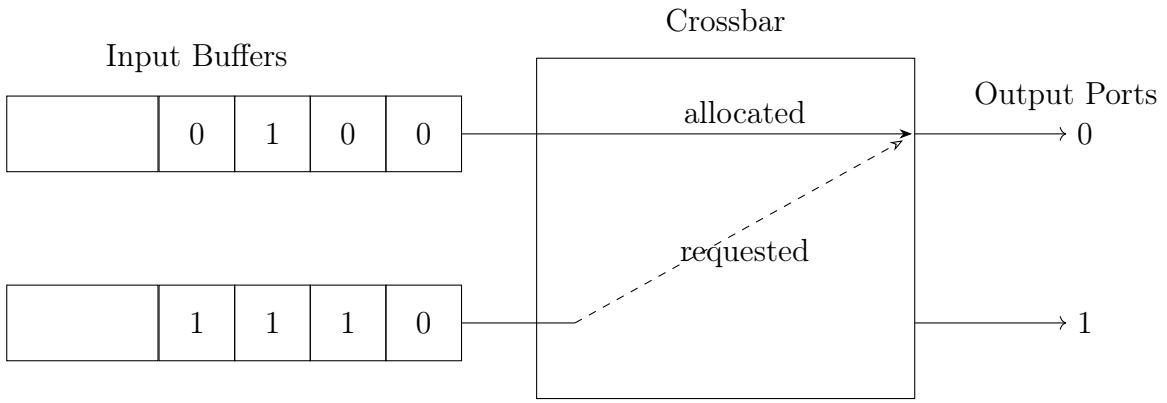


Figure 1.1: Head of Line Blocking among two buffers.

entry buffers on the left and two exit ports on the right. In the buffers there are packets in which it is written their required exit port. The head of the bottom buffer cannot advance because the port is allocated to the other buffer. However, there are packets in the bottom buffer that could make use of port 1, but they are blocked by the queue's head. The use of several channels, even if used in a random way, reduces the probability of this situation, as it increases the number of head packets. Some policies on virtual channels can reduce the HoLB even further. For example, if there are so many virtual channels as possible destination ports, reserve every virtual channel for packets that require to exit by a specific port, then there is not HoLB in routers. However, even with that policy, some similar blocking situations can occur at a network level.

Chapter 2

Lattice Graphs

Tori are not well suited to support global and remote communications. Their relatively long paths among nodes, especially their diameter and average distance, incur high latencies and limited throughput. Thus, reducing topological distances in the network should be pursued. In order to achieve network distance reductions changes must be done to the topology. These topological changes depend on the router degree. If the router degree must be kept within moderate values, that is between 5 and 20, it would be interesting to preserve the good topological properties of tori such as grid locality, easy partitioning and simple routing. Hence, practicable topological changes should not be radical. A typical technique employed to this end has been twisting the wrap-around links of tori [BBK+68, Seq81, BHBA91, Mar81]. Interestingly, this twisting also allows for edge-symmetric networks of sizes for which their corresponding tori are asymmetric [CMV+10, CMB13]. Twisting 2D tori is nearly as old as the history of supercomputers. The Illiac IV developed in 1971 already employed a twisted network. Many works dealing with twisted 2D tori have been published since then. However, when scaling dimensions, the problem of finding a good twisting scheme becomes harder. Very few solutions are known for 3D, with the one presented in [CMV+10] being a practicable example. Studying the effect of twists in higher dimensions remains, to our knowledge, an unexplored domain. A target of this chapter is to improve current topologies for moderate degree interconnection networks. By twisting links of the tori, distance properties are improved and graph symmetry can be enforced. Both topological parameters have impact on performance, as demonstrated in Sections 5.3 and 5.5. If the router degree can be increased, a radically different solution for reducing network diameter can be used in high-degree hierarchical networks, to which Chapter 3 is devoted.

It has been recognized for a long time that Cayley graphs are well suited to interconnection networks. Actually, the widely used rings and tori are Cayley graphs. Nowadays, rings are common in on-chip networks [PBB+10] and, as stated in Chapter 1, many tori are in the top of high-end supercomputing. In [Fio95], Fiol introduced multidimensional circulant graphs as a new algebraic representation for Cayley graphs over Abelian groups. In this chapter, *lattice graphs* are introduced as multidimensional circulant graphs with orthonormal adjacencies, that is, multidimensional meshes plus additional wrap-around links that complete their regular adjacency. Therefore, this chapter is devoted to the study of low and high dimensional twisted tori topologies by means of lattice graphs. Special emphasis on the study of network upgrading and sub-network decompositions is done. Later, special attention will be devoted to symmetric 3D networks, which are completely characterized in Appendix A.

This chapter is organized as follows. Section 2.1 defines lattice graphs and introduces the concepts of graph lift and projection. Section 2.2 describes symmetric lattice graphs, with special emphasis on those based on the cubic crystal lattices. Section 2.3 studies routing in lattice graphs. For the bidimensional case, a matrix reduction can be easily applied, which gives all the information to calculate diameter, average distance and performing routing. For more dimensions a hierarchical algorithm is presented, specially though for symmetric topologies. In Section 2.4 some proposals for physical realization of these topologies are given. First, a proposal for more common technology like the used in Cray's tori. And second, a proposal inspired in the Blue Gene technology, which uses *link chips* to define dynamically the peripheral links. Section 2.5 ends the chapter with a few conclusions.

2.1 Definition of Lattice Graphs

In this section *lattice graphs* are introduced, which will be used to model interconnection topologies of any finite dimension. The lattice graph is not a new concept; in fact, it has different uses. In its most common use, which is also the one considered in this thesis, is a graph built over an n-dimensional grid that induces a regular tiling of the space. In [Fio95], multidimensional circulant graphs were defined as lattice graphs but for any set of adjacencies (not only the orthonormal adjacencies leading to the grids considered in this work), which *a priori* can seem to be a wider family of graphs. However, it can be proved that any multidimensional circulant can be seen as a lattice graph. Hence, the study presented in this section is devoted, in fact, to the family of Cayley graphs over finite Abelian groups; fact that will be proved in Theorem 2.1.3 after stating a few definitions.

Lattice graphs are defined over the integer lattice $\mathbb{Z}^n$. Hence, their nodes are labelled by means of n-dimensional (column) integral vectors. A lattice graph can be intuitively seen as a multidimensional finite grid with additional wrap-around links between opposite faces that complete its regular adjacency.

To define the finite set of nodes of these graphs and their wrap-around links, a modulo function using a square integer matrix will be used. Hence, congruences of vectors modulo matrices are introduced in the next definition.

Definition 2.1.1. *[Fio87] Let $M \in \mathbb{Z}^{n\times n}$ be a non-singular square matrix of dimension n. Two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{Z}^n$ are congruent modulo M if and only if we have $\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \in \mathbb{Z}^n$ such that*

$$\mathbf{v} - \mathbf{w} = u_1\mathbf{m}_1 + u_2\mathbf{m}_2 + \cdots + u_n\mathbf{m}_n = M\mathbf{u},$$

where $\mathbf{m}_j$ denotes the j-th column of M. We will denote this congruence as $\mathbf{v} \equiv \mathbf{w} \pmod{M}$ and the congruence class of $\mathbf{v}$ by $(\mathbf{v} \pmod{M})$.

The set of nodes of a lattice graph will be the elements of the quotient group

$$\mathbb{Z}^n/M\mathbb{Z}^n = \{\mathbf{v} \pmod{M} \mid \mathbf{v} \in \mathbb{Z}^n\}$$

generated by the equivalence relation induced by M. As was proved in [Fio87], $\mathbb{Z}^n/M\mathbb{Z}^n$ has $|\det(M)|$ elements. Now, the formal definition of a lattice graph can be posed.

Definition 2.1.2. *Given a square non-singular integral matrix $M \in \mathbb{Z}^{n\times n}$, the lattice graph generated by M is defined as $\mathcal{G}(M)$, where:*

i) The vertex set is $\mathbb{Z}^n/M\mathbb{Z}^n = \{\mathbf{v} \pmod M \mid \mathbf{v} \in \mathbb{Z}^n\}$.

ii) Two nodes $\mathbf{v}$ and $\mathbf{w}$ are adjacent if and only if $\mathbf{v} - \mathbf{w} \equiv \pm\mathbf{e}_i \pmod M$ for some $i = 1, \ldots, n$.

From here onwards, all matrices will be considered to be non-singular, unless the contrary is stated. Note that, since $\mathbb{Z}^n/M\mathbb{Z}^n$ has $|\det(M)|$ elements, this will be the number of nodes of $\mathcal{G}(M)$. Moreover, since any vertex $\mathbf{v}$ is adjacent to $\mathbf{v} \pm \mathbf{e}_i \pmod M$, the lattice graph $\mathcal{G}(M)$ is, in general[1], regular of degree $2n$, that is, any node has $2n$ different neighbours. Next, we show that the family of lattice graphs coincide with the family of Cayley graphs over Abelian groups.

Theorem 2.1.3. *For any connected Cayley graph G over a finite Abelian group there is $M \in \mathbb{Z}^{n\times n}$ non-singular such that $G \cong \mathcal{G}(M)$.*

Proof. Let Γ be any Abelian finite group, $\{g_1, \ldots, g_n\}$ a subset of Γ and $G_k = Cay(\Gamma; \{\pm g_1, \ldots, \pm g_k\})$. It is proved by induction in k that for any k there is a matrix M_k, a positive integer c and an isomorphism f from $c \times \mathcal{G}(M_k)$ into G_k satisfying $f(0, \mathbf{e}_i) = g_i$ for $i \in \{1, \ldots, k\}$. For the base case $k = 1$ the matrix $M_1 = \big(ord(g_1)\big)$ satisfies the conditions. Otherwise, by induction hypothesis, let M_{k-1} be a matrix such that $c' \times \mathcal{G}(M_{k-1}) \cong Cay(\Gamma; \{g_1, \ldots, g_{k-1}\})$ with an isomorphism $f(0, \mathbf{e}_i) = g_i$ for $i \in \{1, \ldots, k-1\}$. Then, let a be the minimum positive integer such that $ag_k = x_1g_1 + x_2g_2 + \cdots + x_{k-1}g_{k-1}$ for integers x_i (which exists because Γ is finite). Then $M = \begin{pmatrix} M_{k-1} & \mathbf{x} \\ 0 & a \end{pmatrix}$ satisfies $\mathcal{G}(M) \cong c \times G_k$ for some c divisor of c', with an isomorphism $f(0, \mathbf{e}_i) = g_i$ for $i \in \{1, \ldots, k\}$.

As G_n is connected by hypothesis it follows that $G_n \cong \mathcal{G}(M_n)$. □

Example 2.1.4. *The graph $C_{17}(1, 3, 7)$ has as set of nodes the group $\mathbb{Z}_{17}$, and every node n is adjacent to the other six nodes $n \pm 1, \pm 3, \pm 7 \pmod{17}$. As this graph is a Cayley graph over $\mathbb{Z}_{17}$ is isomorphic to some lattice graph. Specifically, this graph is isomorphic to the lattice graph generated by the matrix*

$$\begin{pmatrix} 17 & 3 & 7 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Isomorphisms of lattice graphs is related to equivalences of integral matrices.

Definition 2.1.5. *M_1 is right equivalent to M_2, which is denoted by $M_1 \cong M_2$, if and only if there exists a unitary matrix $P \in \mathbb{Z}^{n\times n}$ such that $M_1 = M_2P$.*

As was proved in [Fio95], if $M_1 \cong M_2$ then the graphs $\mathcal{G}(M_1)$ and $\mathcal{G}(M_2)$ are isomorphic. Moreover, if P is an unitary matrix then

$$Cay(\mathbb{Z}^n/M\mathbb{Z}^n; \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}) \cong Cay(\mathbb{Z}^n/PM\mathbb{Z}^n; \{P\mathbf{a}_1, \ldots, P\mathbf{a}_n\});$$

the isomorphism being $f(\mathbf{x}) = A\mathbf{x}$. It follows that in a lattice graph swapping two rows or changing the sign of one row also results in an isomorphic graph.

Hence, the list of elementary matrix operations that preserve graph isomorphy is:

[1] Unless $\mathbf{e}_i \equiv \pm\mathbf{e}_j \pmod M$ or $2\mathbf{e}_i \equiv \mathbf{0} \pmod M$ for some $i, j \in \{1, \ldots, n\}$.

- add/subtract a column to another,
- swap two columns,
- swap two rows,
- change the sign of a column,
- change the sign of a row.

2.1.1 Projections and Lifts of Lattice Graphs

In this subsection, the concepts of *projection* and *lift* of a lattice graph will be stated. Projecting a lattice graph allows the study of the different lattice graphs of smaller dimensions that are embedded on it, while lifting a lattice graph will be used for increasing its dimension.

Now, performing Gaussian elimination by columns in a matrix is a right-equivalent operation. Therefore, after one step of Gaussian elimination in the generating matrix of a lattice graph gives isomorphic graphs. The resulting matrix would be

$$M \cong \begin{pmatrix} B & \mathbf{c} \\ \mathbf{0}^t & a \end{pmatrix},$$

where $B \in \mathbb{Z}^{n-1 \times n-1}$ is a matrix of smaller dimension, $\mathbf{c} \in \mathbb{Z}^{n-1}$ is a column vector and a is a positive integer. As a consequence, we obtain that $|\det(M)| = |\det(B)|a$, that is, the number of nodes of $\mathcal{G}(M)$ can be expressed in terms of $\mathcal{G}(B)$ and the integer a. Moreover, the lattice graph $\mathcal{G}(B)$ is isomorphic to the subgraph of $\mathcal{G}(M)$ generated by $\{\pm\mathbf{e}_1, \pm\mathbf{e}_2, \ldots, \pm\mathbf{e}_{n-1}\}$, which allows us to state the following definition.

Definition 2.1.6. *Let $M \in \mathbb{Z}^{n \times n}$ be non-singular and $\mathcal{G}(M)$ be lattice graph it generates. Let us consider $M \cong \begin{pmatrix} B & \mathbf{c} \\ \mathbf{0}^t & a \end{pmatrix}$ such that a is a positive integer. Then, we will say that a is the side of $\mathcal{G}(M)$ and $\mathcal{G}(B)$ its projection over $\mathbf{e}_n$. Moreover, we will call $\mathcal{G}(M)$ a lift of $\mathcal{G}(B)$.*

In particular, any lattice graph can be considered to be generated by its unique Hermite matrix, which may be convenient as Examples 2.1.8 and 2.1.9 attempt to show. Before stating the examples, the Hermite normal form of a matrix is recalled.

Definition 2.1.7. *A matrix H is said to be in Hermite normal form if it is upper triangular, has positive diagonal and each $H_{i,j}$ with $j > i$ lies in a complete set of residues modulo $H_{i,i}$.*

Definitions 2.1.6 and 2.1.7 allow to consider a helpful graphical visualization of any lattice graph, that will also be used for routing in Subsection 2.3.2. First, lattice graphs and their subgraphs can be seen as n-dimensional spaces whose dimensions are sized by the elements in the principal diagonal of M. Each column vector in M represents a graph dimension, signaling the point in the space at which a new copy of the tile induced by M is located; this is important as column vectors dictate the pattern of the wrap-around connections of each dimension.

Moreover, from the cardinal equality $|\mathcal{G}(M)| = |\mathcal{G}(B)|a$, the lattice graph $\mathcal{G}(M)$ can be seen as composed of a disjoint copies of its projection $\mathcal{G}(B)$. One or several parallel cycles

connect these disjoint copies completing the adjacency pattern. The length of these cycles can be computed as $ord(\mathbf{e}_n)$, which is the order of the element $\mathbf{e}_n$ in the group $\mathbb{Z}^n/M\mathbb{Z}^n$. According to [Fio87], the order of any element $\mathbf{x}$ can be computed as

$$ord(\mathbf{x}) = \frac{\det(M)}{\gcd(\det(M), \gcd(\det(M)M^{-1}\mathbf{x}))}.$$

Note that the second gcd (greatest common divisor) in the fraction corresponds to the gcd of the elements of a vector. The number of vertices of each cycle lying in each copy of $\mathcal{G}(B)$ can be calculated as the length of the cycle over the side of the graph, that is $\frac{ord(\mathbf{e}_n)}{a}$.

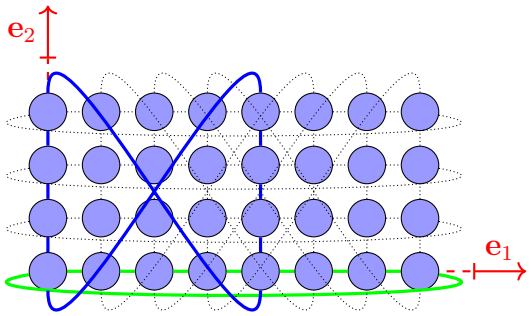


Figure 2.1: Two perpendicular cycles of length 8 in the $RTT(4)$.

Example 2.1.8. *The rectangular twisted torus[CMV+10] is a lattice graph of size $2a \times a$ and twist a; it is denoted as $RTT(a)$. A graphical representation of $RTT(4)$ can be seen in Figure 2.1. This graph is generated by the matrix $H = \begin{pmatrix} 2a & a \\ 0 & a \end{pmatrix}$ and its side is a. Using H, the graph can be seen as a mesh of $2a \times a$ ($h_{1,1} \times h_{2,2}$). In the previous representation, wrap-around links in $\mathbf{e}_1$ (first) dimension conserve their horizontality since $h_{2,1} = 0$; wrap-around links in $\mathbf{e}_2$ (second) dimension do not conserve their verticality but suffer a twist of a columns since $h_{1,2} = a$. According to Definition 2.1.6, the projection over $\mathbf{e_2}$ of $RTT(a)$ is a cycle of $2a$ nodes. As the side of $RTT(a)$ is a, it will have a disjoint cycles of $2a$ nodes. As $ord(\mathbf{e}_2)$ (the element representing a jump in $\mathbf{e}_2$ dimension) is $2a$, the graph will have a parallel cycles of length $2a$ in that dimension. Each of these a cycles contains two vertices of each projection.*

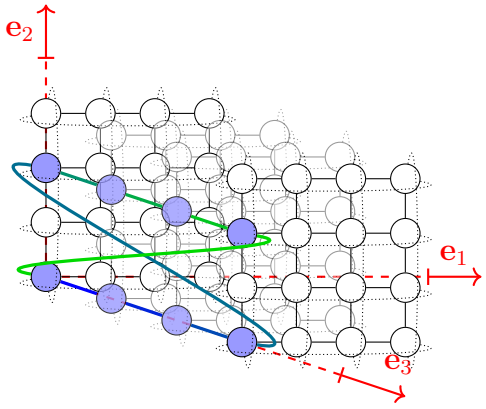


Figure 2.2: The cycle $\langle \mathbf{e}_3 \rangle$ joining the disjoint copies of the projection.

Example 2.1.9. *Now, let* $M = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 2 \\ 0 & 0 & 4 \end{pmatrix}$ *and consider the lattice graph* $\mathcal{G}(M)$*. Note that* M *is in Hermite form.* $\mathcal{G}(M)$ *can be seen as a* $4 \times 4 \times 4$ *cubic grid whose side is also 4. Three sets of wrap-around links, each one connecting opposite faces, have to be added to the grid-based cube. Wrap-around links in* $\mathbf{e}_1$ *always remain horizontal by construction, as imposed by the* $n-1$ *zeros in the first column vector of any Hermite matrix. Wrap-around links in the* $\mathbf{e}_2$ *dimension remain vertical in this graph because* $m_{1,2} = 0$ *but, in general, they can undergo only a twist over the* $\mathbf{e}_1$ *dimension of* $m_{1,2}$ *units. Finally, wrap-around links in the* $\mathbf{e}_3$ *dimension can undergo twists over both* $\mathbf{e}_1$ *and* $\mathbf{e}_2$ *dimensions. In the graph of this example, no twist is applied in* $\mathbf{e}_3$ *over* $\mathbf{e}_1$ *because* $m_{1,3} = 0$ *and a twist of 2 units is applied over the* $\mathbf{e}_2$ *dimension as* $m_{2,3} = 2$*. As can be seen in Figure 2.2, the projection of* $\mathcal{G}(M)$ *is* $\mathcal{G}(\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix})$*, a 2D torus* $T(4,4)$*. Thus, the graph is composed of 4 disjoint copies of its projection, each of them connected by a cycle of length 8, as represented in the figure. Note that for every vertex in the graph there will be a similar cycle with the same pattern as the one represented in the figure. The cycle intersects in two vertices with each copy of the projection. For the sake of the clarity, only one cycle between copies of* $\mathcal{G}(\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix})$ *has been represented.*

Note that the projection can be over any $\mathbf{e}_i$, simply by swapping rows i and n (which gives an isomorphic graph) and then, project over $\mathbf{e}_n$. Moreover, as we will see later, symmetries will make irrelevant over which dimension we project, so we will consider $\mathbf{e}_n$ by default. The resulting projection can again be projected over another vector, which results in a projection over a plane of the lattice graph. Clearly, projecting over a pair of vectors $\{\mathbf{e}_i, \mathbf{e}_j\}$ can be done in any order, since projecting first over $\mathbf{e}_i$ and then over $\mathbf{e}_j$ results in the same graph as projecting first over $\mathbf{e}_j$ and then over $\mathbf{e}_i$. Following the same idea, we can project over several dimensions iteratively. Therefore, the result of projecting iteratively over the vectors in the set $\{\mathbf{e}_{i_1}, \ldots, \mathbf{e}_{i_r}\}$ will be called the *projection* of $\mathcal{G}(M)$ *over the set*. In this case we will call it an r-dimensional projection which turns into a lattice graph generated by a $(n-r) \times (n-r)$ matrix.

Now, let us consider a new way of lifting lattice graphs. In this new operation, given two lattice graphs we will look for another one which has them as projections but minimizing the resulting degree.

Definition 2.1.10. *The lattice graph* $\mathcal{G}(M)$ *is a common lift of* $\mathcal{G}(M_1)$ *and* $\mathcal{G}(M_2)$ *if both can be obtained as projections of* $\mathcal{G}(M)$*.*

Remark 2.1.11. *There are several ways of obtaining different common lifts of two given lattice graphs. A straightforward one is to consider the lattice graph* $\mathcal{G}(M_1 \oplus M_2)$ *generated by the direct sum of the matrices. As we state next, this option leads to the Cartesian product of the two given lattice graphs.*

Lemma 2.1.12. $\mathcal{G}(M_1 \oplus M_2)$ *is a common lift of* $\mathcal{G}(M_1)$ *and* $\mathcal{G}(M_2)$ *and* $\mathcal{G}(M_1 \oplus M_2) \cong \mathcal{G}(M_1) \square \mathcal{G}(M_2)$*.*

In addition, there exist other common lifts that obtain $\mathcal{G}(M_1)$ and $\mathcal{G}(M_2)$ as projections but generating a lattice graph of smaller dimension. Note that this would be beneficial for cost aspects, such as minimizing the degree of the network routers, and to provide a good relation between the size of the graph and its projections.

Theorem 2.1.13. *Given two lattice graphs $\mathcal{G}(M_1)$ and $\mathcal{G}(M_2)$, the lattice graph $\mathcal{G}(M_1 \boxplus M_2)$ is defined as follows: Let $M_1 \cong H_1$ and $M_2 \cong H_2$ with H_1 and H_2 in Hermite normal form. Let C be the submatrix with the first common columns of H_1 and H_2. Then $H_1 = \begin{pmatrix} C & R_A \\ 0 & A \end{pmatrix}$ and $H_2 = \begin{pmatrix} C & R_B \\ 0 & B \end{pmatrix}$, where A and B are square matrices. Then*

$$M_1 \boxplus M_2 = \begin{pmatrix} C & R_A & R_B \\ 0 & A & 0 \\ 0 & 0 & B \end{pmatrix}.$$

It is obtained that:

i) $\mathcal{G}(M_1 \boxplus M_2)$ *is a common lift of* $\mathcal{G}(M_1)$ *and* $\mathcal{G}(M_2)$

ii) $\max\{dim(\mathcal{G}(M_1)), dim(\mathcal{G}(M_2))\} \leq dim(\mathcal{G}(M_1 \boxplus M_2)) \leq dim(\mathcal{G}(M_1 \oplus M_2))$

Proof. The first item is obtained by construction. For the second one, consider $\max\{dim(\mathcal{G}(M_1)), dim(\mathcal{G}(M_2))\} \leq \dim(\mathcal{G}(M_1 \boxplus M_2)) = dim(\mathcal{G}(M_1)) + dim(\mathcal{G}(M_2)) - dim(\mathcal{G}(C)) \leq dim(\mathcal{G}(M_1)) + dim(\mathcal{G}(M_2)) = dim(\mathcal{G}(M_1 \oplus M_2))$ □

Note that when the matrices M_1 and M_2 have no common columns, both $\mathcal{G}(M_1 \boxplus M_2)$ and $\mathcal{G}(M_1 \oplus M_2)$ coincide. Moreover, by construction, the operation $\mathcal{G}(M_1 \boxplus M_2)$ provides a lift that minimizes its dimension. Although in Subsection 2.2.4 several examples will be considered, the next one tries to clarify this definition.

Example 2.1.14. *Let M_1 and M_2 the integral matrices defined by $M_1 = \begin{pmatrix} 2a & 0 \\ 0 & 2a \end{pmatrix}$ and $M_2 = \begin{pmatrix} 2a & a \\ 0 & a \end{pmatrix}$. Clearly, $\mathcal{G}(M_1)$ is the 2D-torus of side $2a$ and $\mathcal{G}(M_2)$ the RTT. Then if we consider $M_1 \boxplus M_2 = \begin{pmatrix} 2a & 0 & a \\ 0 & 2a & 0 \\ 0 & 0 & a \end{pmatrix}$, the resulting is a lattice graph of degree 6 having both graphs as its projections.*

2.2 Symmetric Lattice Graphs

Symmetry is a desirable property for any network as it impacts on performance and routing efficiency. Many interconnection networks have been based on vertex-symmetric graphs, but less attention has been devoted to edge-symmetric networks. Square and cubic tori have been the networks of choice for many designs as they are symmetric (vertex and edge symmetric). For this reason, symmetric lattice graphs will be considered in this section.

Symmetry is a desirable characteristic for any network as it has a big impact on performance and router design complexity. In terms of performance, tori are clearly superior to meshes that do not use wraparound edges which simplifies their planar design at the price of losing vertex-symmetry. Many interconnection networks have been based on vertex-symmetric graphs. In a vertex-symmetric graph, any node can “observe" the same environment. This is the case of current parallel computers from Cray and IBM, among others, that are built around torus networks.

Less attention has been devoted to edge-symmetric networks, that is, those in which any link has the same surrounding environment. Square torus has been the network of

choice for many designs as it is symmetric (vertex and edge symmetric). However, for practical reasons such as packaging, modularity, cost and scalability, the number of nodes per dimension might be different. These topologies are denoted as mixed-radix networks in [DT03]. Mixed-radix tori have the drawback of being non-edge-symmetric which leads to an imbalanced utilization of network links and buffers. For different commonly used traffic patterns, the load on the longer dimension is higher than on the shorter one, and hence, links in the longer dimension become network bottlenecks, [CMV+10].

It is important to state the following result as projections of symmetric graphs will be considered later.

Theorem 2.2.1. *The projections of a linearly symmetric lattice graph are all isomorphic.*

Proof. Let $proj_i(\mathcal{G}(M))$ denote the projection of $\mathcal{G}(M)$ over $\mathbf{e}_i$ and $\mathcal{B}_n$ the n-dimensional orthonormal basis. Clearly, $proj_i(\mathcal{G}(M))$ is isomorphic to the subgraph of $\mathcal{G}(M)$ generated by $\mathcal{B}_n \setminus \{\mathbf{e}_i\}$. Since $\mathcal{G}(M)$ is symmetric there is an automorphism $\phi \in Aut(\mathcal{G}(M))$ such that $\phi(\mathbf{e}_i) = \pm\mathbf{e}_j$. As $\mathbf{e}_i$ is the only generator not in $proj_i(\mathcal{G}(M))$, $\mathbf{e}_j$ is the only generator not in $\phi(proj_i(\mathcal{G}(M)))$. Hence, as ϕ is an automorphism, it follows that $proj_i(\mathcal{G}(M)) \cong proj_j(\mathcal{G}(M))$. □

Remember that all Cayley graphs are vertex-transitive, and therefore, a lattice graph is symmetric if and only if it is edge-transitive. In Appendix A there is a characterization of the matrices which generate symmetric lattice graphs for dimensions 2 and 3. For the sake of clarity this result is also summarized here:

Theorem 2.2.2. *Let $M \in \mathbb{Z}^{2\times 2}$. Then, the lattice graph $\mathcal{G}(M)$ is edge-transitive if and only if it is isomorphic to $\mathcal{G}(M')$ for M' being one of the following matrices for some $a, b \in \mathbb{Z}$.*

i) $\begin{pmatrix} a & b \\ b & a \end{pmatrix}$,

ii) $\begin{pmatrix} a & -b \\ b & a \end{pmatrix}$,

iii) $\begin{pmatrix} a & -b \\ a & b \end{pmatrix}$,

iv) $\begin{pmatrix} a & 2 \\ 0 & 2 \end{pmatrix}$,

v) $\begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$, $\begin{pmatrix} 3 & 3 \\ 1 & -1 \end{pmatrix}$ *or* $\begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}$.

Theorem 2.2.3. *Let $M \in \mathbb{Z}^{3\times 3}$. Then, the lattice graph $\mathcal{G}(M)$ is linearly symmetric if and only if it is isomorphic to $\mathcal{G}(M')$, for some $a, b, c \in \mathbb{Z}$, where:*

$$M' \in \left\{ \begin{pmatrix} a & c & b \\ b & a & c \\ c & b & a \end{pmatrix}, \begin{pmatrix} a & b & c \\ a & c & -b-c \\ a & -b-c & b \end{pmatrix} \right\}.$$

These graphs are very interesting. The lattice graph generated by the matrix $\begin{pmatrix} a & -b \\ b & a \end{pmatrix}$ is isomorphic to the Gaussian graph generated by $a + bi$. The family of Gaussian graphs was introduced in [MBS+08] as a model for interconnection network topologies. This family englobes the RTT as the case $a = b$. The lattice graphs generated by the matrices $\begin{pmatrix} a & b \\ b & a \end{pmatrix}$ or $\begin{pmatrix} a & -b \\ a & b \end{pmatrix}$ are isomorphic to the Kronecker product of cycles as stated in the following theorem:

Theorem 2.2.4. *Let $a, b \in \mathbb{Z}$. Then, the Kronecker product of the two cycles $C_a \times C_b$ is isomorphic to:*

- $\mathcal{G}(M)$, *where* $M = \begin{pmatrix} \frac{a+b}{2} & \frac{a-b}{2} \\ \frac{a-b}{2} & \frac{a+b}{2} \end{pmatrix}$, *if a and b are odd integers.*
- *Two disjoint copies of* $\mathcal{G}(M)$, *with* $M = \begin{pmatrix} \frac{a}{2} & \frac{-b}{2} \\ \frac{a}{2} & \frac{b}{2} \end{pmatrix}$, *if a and b are even integers.*
- $\mathcal{G}(M)$, *with* $M = \begin{pmatrix} \frac{a}{2} & -b \\ \frac{a}{2} & b \end{pmatrix}$, *if a is an even integer and b is an odd integer.*

It is clear that the previous characterization gives us a broad family of symmetric graphs. For the three-dimensional case, note that the side of the matrix $\begin{pmatrix} a & c & b \\ b & a & c \\ c & b & a \end{pmatrix}$ is $\gcd(a, b, c)$. Thus, maximizing the side implies without loss of generality that $b, c \in \{0, \pm a\}$. This is exactly the case of cubic crystal lattices [Jan73], which are:

- **Primitive Cubic Lattice:** $\begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix}$.
- **Face-centered Cubic Lattice:** $\begin{pmatrix} a & a & 0 \\ a & 0 & a \\ 0 & a & a \end{pmatrix}$.
- **Body-centered Cubic Lattice:** $\begin{pmatrix} -a & a & a \\ a & -a & a \\ a & a & -a \end{pmatrix}$.

For the rest of the chapter there will be a focus on cubic crystal lattice graphs for three major reasons. First, the projections of these lattice graphs are also symmetric. Moreover, the fact that their side is maximum will provide an efficient routing algorithm. Finally, the selection of this family of 3D symmetric lattice graphs exemplify how the previously introduced graph operations can be applied to construct a wide variety of new topologies for interconnection networks.

In the following the lattice graphs defined by the cubic crystal lattices will be considered, together with their isomorphisms with previously studied network topologies and a comparison among them in terms of their distance properties.

Once detailed the special case of the cubic crystal graphs, their upgrading process can be considered. As it has been asserted before, symmetry helps when an application runs

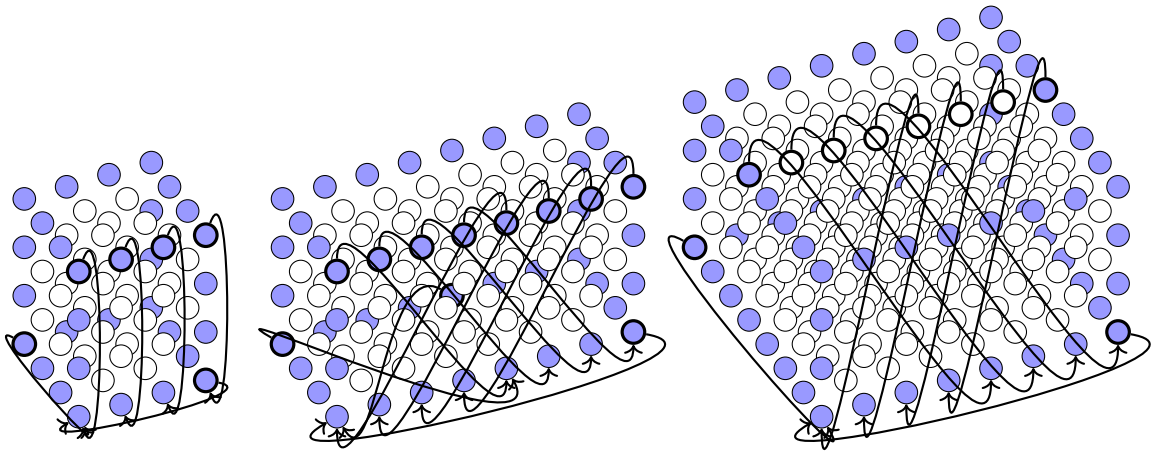


Figure 2.3: The three Cubic Crystal Graphs: PC, FCC and BCC.

on the whole network. However, in big systems the user typically only has a partition of the complete machine assigned. Therefore, looking for symmetry in higher dimensions cannot be prioritized. Nevertheless, reducing the distance properties of the whole network would be still beneficial since applications and system software sometimes run over the entire network. Consequently, what will be looked for in higher dimensional networks is to embed the previous crystal cubic lattice graphs. Therefore, two more subsections are included in which two different methods for upgrading cubic crystal lattice graphs are explored. The first one is to consider the lifting of crystal graphs, which results in 4D topologies. Whenever possible, the lift is done in such a way that the resulting eight-degree topology preserves symmetry. Furthermore, it will be introduced a tree that represents the process of network upgrading, preserving symmetry.

2.2.1 Cubic Crystal Lattice Graphs

The **Primitive Cubic Lattice Graph** $PC(a)$ is defined as the lattice graph generated by the matrix associated with the primitive cubic lattice, that is:

$$\begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix}.$$

Clearly, the number of nodes of the graph is a^3, which is the determinant of the diagonal matrix. Clearly, $PC(a)$ is isomorphic to the 3D torus of side a, or equivalently, the a-ary 3-cube.

Lemma 2.2.5. *The projection of $PC(a)$ is the 2D torus graph of side a or $\mathcal{G}(\begin{pmatrix} a & 0 \\ 0 & a \end{pmatrix})$.*

The **Face-centered Cubic lattice graph** $FCC(a)$ of side a can be defined as the lattice graph generated by the matrix associated with the face-centered cubic crystal lattice, that is:

$$\begin{pmatrix} a & a & 0 \\ a & 0 & a \\ 0 & a & a \end{pmatrix} \cong \begin{pmatrix} 2a & a & a \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix}. \tag{2.1}$$

The number of nodes of the graph is $|\det(M)| = 2|a|^3$.

Lemma 2.2.6. *The projection of $FCC(a)$ is the rectangular twisted torus graph of side a, $RTT(a)$.*

Proof. After performing Gaussian elimination, as in the previous expression (2.1), it is obtained the Hermite form of the matrix. Then it is immediate to see that its projection is generated by $\begin{pmatrix} 2a & a \\ 0 & a \end{pmatrix}$. As it has been seen before and it was proved in [CMB13], this graph is isomorphic to the rectangular twisted torus $RTT(a)$ of side a or the *Gaussian graph* generated by $a + ai$ [MBS+08]. □

A $FCC(a)$ is isomorphic to the *prismatic doubly twisted torus* of side a ($PDTT(a)$), introduced in [CMV+10], as the next proposition proves.

Proposition 2.2.7. *$FCC(a)$ is isomorphic to the prismatic doubly twisted torus of side a, $PDTT(a)$.*

Proof. The $PDTT(a)$ was defined in [CMV+10] as a graph in which the connectivity of each plane is a $RTT(a)$, hence the isomorphism is immediate once it has been proved that all the projections of $FCC(a)$ are isomorphic to $RTT(a)$. Note that this fact can be inferred from Lemma 2.2.6 and Theorem 2.2.1. □

The **Body-centered Cubic lattice graph** $BCC(a)$ of side a can be defined as the lattice graph generated by the matrix:

$$\begin{pmatrix} -a & a & a \\ a & -a & a \\ a & a & -a \end{pmatrix} \cong \begin{pmatrix} 2a & 0 & a \\ 0 & 2a & a \\ 0 & 0 & a \end{pmatrix}. \tag{2.2}$$

The number of nodes of the graph is $4a^3$. As far as we know, this graph has not previously been considered for interconnection networks. However, as it will be seen later, the graph not only meets the symmetry requirements but also has a better nodes/diameter ratio than PC and FCC, as it will be explained later. Moreover, it embeds 2D symmetric tori as is proved in:

Lemma 2.2.8. *The projection of $BCC(a)$ is the 2D torus graph $T(2a, 2a)$*

Proof. Performing Gaussian elimination as in expression (2.2) shows that the projection of $BCC(a)$ is the lattice graph generated by the matrix $\begin{pmatrix} 2a & 0 \\ 0 & 2a \end{pmatrix}$, which is the 2D torus of side $2a$. □

A graphical representation of the three topologies introduced in this subsection is presented in Figure 2.3.

2.2.2 Cubic Crystal Lattice Graph Comparison

Among the three different 3D symmetric topologies based on cubic crystal lattices, two of them—the 3D torus or PC and the PDTT or FCC—were previously known, and the last one, that is the BCC, is a new proposal introduced in this chapter. In the remainder of this subsection, our aim is to consider their distance properties and to perform a first comparison in terms of diameter, average distance and projections.

First of all, it is important to highlight that a cubic crystal lattice graph exists for any number of nodes that is a power of two. This is significant because it allows to gracefully upgrade a network in three steps while conserving symmetry. If t is a positive integer, then:

- There exists a primitive cubic lattice graph with 2^{3t} nodes.
- There exists a face-centered cubic lattice graph with 2^{3t+1} nodes.
- There exists a body-centered cubic lattice graph with 2^{3t+2} nodes.

Although this fact provides practical versatility, it complicates the comparison among networks. The following expressions for average distance of the three crystals have been calculated under the assumption that $8\bar{k}(|\det(M)|-1)$ is a polynomial and computationally checked for a number of nodes up to $40,000$.

$PC(a)$ has average distance

$$\bar{k} = \begin{cases} \frac{3a^4}{4(a^3-1)} & \text{if } 2|a \text{ and} \\ \frac{3a^4-3a^2}{4(a^3-1)} & \text{if } 2\not|a. \end{cases}$$

$FCC(a)$ has average distance

$$\bar{k} = \begin{cases} \frac{7a^4-2a^2}{4(2a^3-1)} & \text{if } 2|a \text{ and} \\ \frac{7a^4-2a^2-1}{4(2a^3-1)} & \text{if } 2\not|a. \end{cases}$$

$BCC(a)$ has average distance

$$\bar{k} = \begin{cases} \frac{35a^4-8a^2}{8(4a^3-1)} & \text{if } 2|a \text{ and} \\ \frac{35a^4-14a^2+30}{8(4a^3-1)} & \text{if } 2\not|a. \end{cases}$$

In Table 2.1 the distance properties for the three graphs are summarized. For an easier comparison, note that average distance values are given as approximations. Mixed-radix torus graphs that have the same number of nodes of the FCC and BCC crystals have been also added in the table for comparison. Clearly, crystals have better distance properties than their corresponding torus networks. Moreover, BCC is more *dense* than the other two cubic crystals in the sense that for the same diameter, it attains a greater number of nodes. Finally, as we have seen in previous subsections, while FCC has the twisted torus as its projection, both PC and BCC are lifts of a 2D symmetric torus graph.

Having considered distance-related parameters for comparing crystals, let us also take into account other topological parameters to complete the study. As said in Section 1.6, the bisection bandwidth is commonly used to evaluate a topology. However the work in [CMV+10] showed that it is not a tight bound for network throughput in twisted topologies. Indeed, the same happens with any non-torus lattice graph.

Hence, for symmetric lattice graphs is better to use the bound based on the average distance given in Equation (1.1). For lattice graphs, $\Delta = 2n$ where n is the number of dimensions. Hence, in $FCC(a)$, maximum throughput will be bounded by $\frac{48}{7a}$ and in $BCC(a)$ by $\frac{192}{35a}$. Nevertheless, the previous count cannot be applied to edge-asymmetric networks such as mixed-radix tori. In that case, it can be seen that throughput is inversely

Topology	Nodes	Diameter	Average Distance
$PC(a)$	a^3	$3\left\lfloor \frac{a}{2} \right\rfloor$	$\approx \frac{3}{4}a = 0.75a$
$T(2a, a, a)$	$2a^3$	$a + 2\left\lfloor \frac{a}{2} \right\rfloor$	$\approx a$
$FCC(a)$	$2a^3$	$\left\lfloor \frac{3}{2}a \right\rfloor$	$\approx \frac{7}{8}a = 0.875a$
$T(2a, 2a, a)$	$4a^3$	$\left\lfloor \frac{5}{2}a \right\rfloor$	$\approx \frac{5}{4}a = 1.25a$
$BCC(a)$	$4a^3$	$\left\lfloor \frac{3}{2}a \right\rfloor$	$\approx \frac{35}{32}a = 1.09375a$

Table 2.1: Distance properties of cubic crystal lattice graphs.

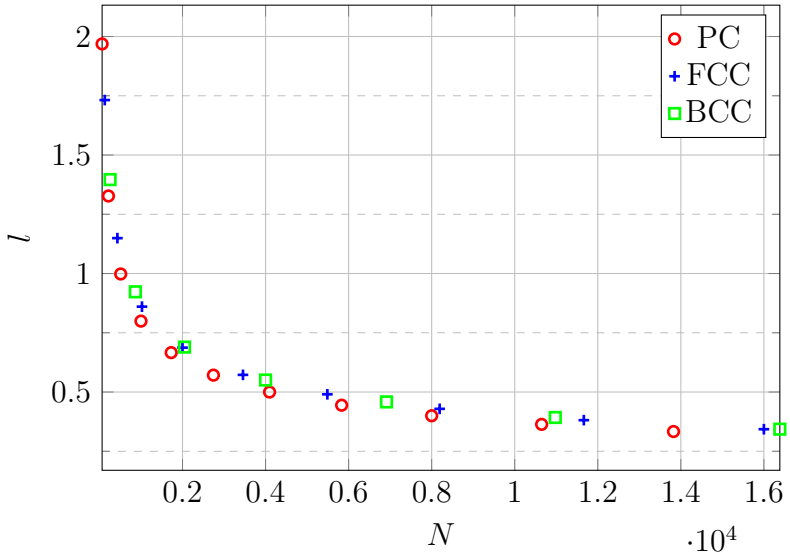


Figure 2.4: Maximum injected phits/cycle/node to each even network size $N = |\det(M)|$.

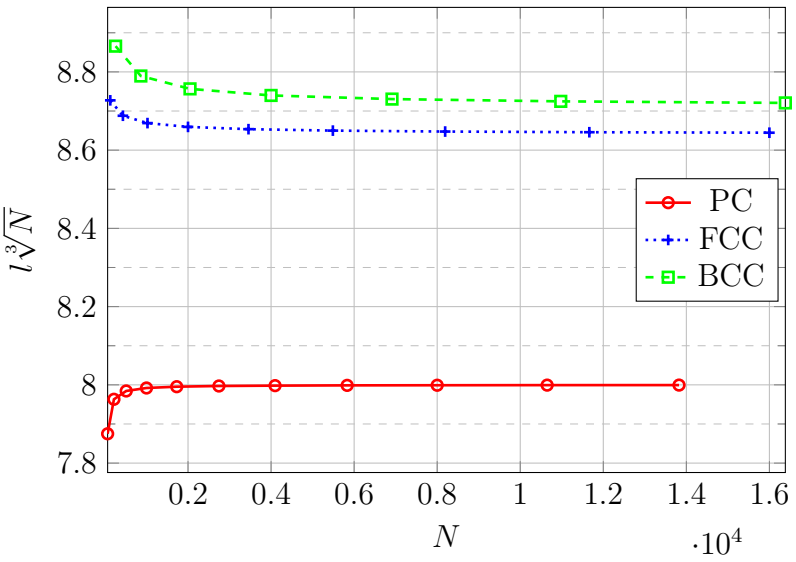


Figure 2.5: $l\sqrt[3]{N}$ to each even network size $N = |\det(M)|$. Quotients are preserved.

proportional to the maximum average distance per dimension, namely $\frac{\Delta}{n\bar{k}_{max}}$, as inferred from [CMV+10]. Network throughput for both $T(2a, a, a)$ and $T(2a, 2a, a)$ is bounded by $\frac{12}{3a} = \frac{4}{a}$ as $\bar{k}_{max} \approx \frac{a}{2}$, given that their longest dimensions are $2a$-node rings. This leads to an improvement in maximum throughput under uniform traffic of 71% when comparing $FCC(a)$ to $T(2a, a, a)$ and 37% for $BCC(a)$ versus $T(2a, 2a, a)$.

Being symmetric has more positive impact when the number of nodes is $2a^3$. In $T(2a, a, a)$, when the links in the longest dimension are fully utilized, links in the other two shortest dimensions are used at 50%. This is because, on average, the length of the paths in the longest dimension doubles the length of the shortest ones. When the number of nodes is $4a^3$, $T(2a, 2a, a)$ uses its resources better as only links in one dimension operate at half rate.

Since network throughput under uniform traffic is bounded by an expression on the side of the lattice graph, it is possible to make a graphical comparison of crystal graphs. In first place, Figure 2.4 shows for different network sizes $N = |\det(M)|$ this theoretical load l injected under uniform traffic. After a normalization, Figure 2.5 shows the amount $l\sqrt[3]{N}$ instead of l; value chosen to make each curve converge to some constant. Note that for each graph size, the quotient between these amounts is the same as the quotient of the loads themselves. Thus, this shows that BCCs are about 9% more efficient under uniform traffic than PCs.

2.2.3 Symmetric Lifts of Cubic Crystal Graphs

First, there is a straightforward way of lifting a $PC(a)$ to 4D, which is the Cartesian product of the PC by one cycle of length a, thus obtaining the generator matrix

$$\begin{pmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & a \end{pmatrix}.$$

The 4D torus generated by the previous matrix is a symmetric lift of $PC(a)$. However, the lifting technique can be used to embed the symmetric 3D torus in a different lattice graph. The *body centered hypercube lattice graph* will be denoted as $4D\text{-}BCC$, that is, the lattice graph generated by the matrix

$$\begin{pmatrix} 2a & 0 & 0 & a \\ 0 & 2a & 0 & a \\ 0 & 0 & 2a & a \\ 0 & 0 & 0 & a \end{pmatrix}.$$

Proposition 2.2.9. *$4D\text{-}BCC(a)$ is a symmetric lattice graph of side a and projection $PC(2a)$.*

Proof. Let ϕ be defined by $\phi(\mathbf{e}_i) = \mathbf{e}_{i+1 \pmod n}$. The matrix associated to the function ϕ is $P = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$. As $Q = M^{-1}PM = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 2 & 1 \end{pmatrix}$ is an integer matrix, it is concluded that ϕ is an automorphism of $4D\text{-}BCC$ (by Theorems A.3.1 and A.3.2). In

the group generated by ϕ there are enough automorphisms to provide the edge-symmetry. It should be noted that the projection is straightforward as the matrix is triangular superior. □

Now, to obtain a lift of the FCC, there are two ways of doing so which make the lifted graph symmetric. The first one will be denoted as $4D\text{-}FCC$ (*4-dimensional face-centered cubic lattice graph*), that is, the lattice graph generated by the matrix

$$\begin{pmatrix} 2a & a & a & a \\ 0 & a & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & a \end{pmatrix}.$$

Proposition 2.2.10. *$4D\text{-}FCC(a)$ is a symmetric lattice graph of side a whose projection is a $FCC(a)$.*

Proof. Exactly like the proof of Proposition 2.2.9; the matrix $Q = M^{-1}PM$ is different but still with integer entries. □

The second way to lift a FCC is introduced below.

Proposition 2.2.11. *The lattice graph generated by the matrix* $\begin{pmatrix} a & -a & -a & -a \\ a & a & -a & a \\ a & a & a & -a \\ a & -a & a & a \end{pmatrix}$ *is a symmetric lifting of the $FCC(2a)$.*

Proof. First, the following two matrices are right-equivalent:

$$\begin{pmatrix} a & -a & -a & -a \\ a & a & -a & a \\ a & a & a & -a \\ a & -a & a & a \end{pmatrix} \cong \begin{pmatrix} 2a & -2a & 0 & -a \\ 0 & 2a & -2a & a \\ 2a & 0 & 2a & -a \\ 0 & 0 & 0 & a \end{pmatrix}$$

Hence, the corresponding lattice graphs are isomorphic. Note that the $(4,4)$-minor corresponds with the generating matrix of $FCC(2a)$. Finally, for symmetry, the procedure described in the proof of Proposition 2.2.9 is repeated. □

This second lifting relates the graphs obtained to the family of Lipschitz graphs and quaternion algebras, introduced in [MBG09], for obtaining perfect codes over 4D spaces. This graph will be denoted as $Lip(a)$. Specifically $Lip(a) \cong Cay(\frac{\mathbb{H}[\mathbb{Z}]}{(a+ai+aj+ak)\mathbb{H}[\mathbb{Z}]}; \{\pm 1, \pm i, \pm j, \pm k\})$, where $\mathbb{H}[\mathbb{Z}]$ are the integer quaternions and $1, i, j, k$ the quaternion units.

Finally, there are several ways of lifting the BCC, although none of them preserves symmetry as proved in the next theorem.

Theorem 2.2.12. *Any lift of BCC yields a non-edge-symmetric graph.*

Proof. Let $M = \begin{pmatrix} 2a & 0 & a \\ 0 & 2a & a \\ 0 & 0 & a \end{pmatrix}$, $BCC(a) \cong \mathcal{G}(M)$. Assume that there exists a symmetric lift $\mathcal{G}(L)$ of $BCC(a)$, where

$$L = \begin{pmatrix} 2a & 0 & a & x \\ 0 & 2a & a & y \\ 0 & 0 & a & z \\ 0 & 0 & 0 & t \end{pmatrix}$$

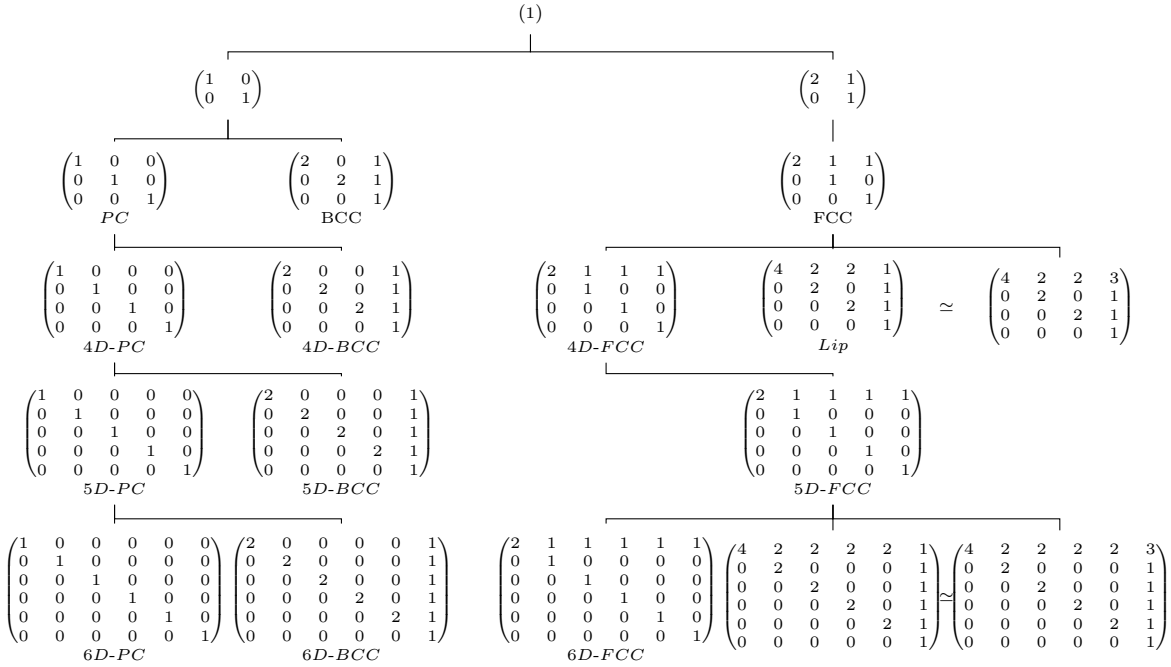


Figure 2.6: Tree showing lifts and projections of cubic crystal graphs up to dimension 6.

is in Hermite form, *i.e.*, $0 \leq x, y < 2a$ and $0 \leq z < a$. For symmetry, the gcd of every row must be the same (map $\mathbf{e}_i$ into $\mathbf{e}_n$ and Gauss-reduce), hence t divides all the other entries of L and without loss of generality it can be assumed that $t = 1$. By Theorems A.3.1 and A.3.2 it is known that automorphisms are matrices P satisfying the condition that $L^{-1}PL$ is an integer matrix where P is unitary and has only ± 1 entries. Both, the sets of these matrices that would give edge-transitivity, and the possible lifts, are finite. Hence it is possible to run a computation that gives the negative result. □

As it has been concluded before, there is no decisive interest in obtaining a symmetric graph in 4D such that its 3D partitions remain themselves symmetric. Therefore, it is of interest to explore which of the lattice graphs whose projection is a BCC would be the most interesting.

Figure 2.6 summarizes how the previous constructions can be generalized to any number of dimensions. The procedure is represented in a tree. In this tree, nodes are the matrices of the lattice graphs. Note that, for an easier visualization, matrices have been normalized by multiplying by $\frac{1}{a}$. Hence, each child is a lift of its parent. Moreover, lift are restricted to those whose side is greater or equal to the half of the side of its projection, otherwise many more graphs would appear.

The root of the tree is the matrix associated with a cycle. The lifts of the cycle conserving symmetry, and fulfilling the restrictions mentioned above, are the torus and the RTT introduced in Section 2.1. Then, as it has been seen in Section 2.2, the cubic crystal lattice graphs are lifts of these two. The two branches show that only two families are obtained. The left branch consists of the infinite family of symmetric tori or n-dimensional PCs; and each $nD\text{-}PC$ has a $nD\text{-}BCC$ sibling that is a leaf, without any further symmetric lift. The right branch is the family of the n-dimensional FCCs; the $nD\text{-}FCC$ always has the $(n+1)D\text{-}FCC$ as a symmetric lift. Moreover, there are some dimensions (4 and 6 in the figure) in which a different lift exists. Interestingly, two non-right-equivalent matrices

generate isomorphic graphs (denoted with $\simeq$). The two branches in the tree are really different and, as it is shown next, they can be used to obtain new hybrid lattice graphs.

2.2.4 Hybrid Graphs: Common Lift of Crystal Graphs

In this subsection a different approach for embedding crystal graphs is considered, that is, to create common lifts that do not necessarily combine symmetric graphs. As shown in the next example, to handle graphs using the $\boxplus$ operator that belong to the same branch of the tree in Figure 2.6 has some advantages.

Example 2.2.13. *The first one is the hybrid graph obtained as a common lift of the $PC(2a)$ and $BCC(a)$. The calculation described in the Theorem 2.1.13 leads to the matrix*

$$\begin{pmatrix} 2a & 0 & 0 \\ 0 & 2a & 0 \\ 0 & 0 & 2a \end{pmatrix} \boxplus \begin{pmatrix} 2a & 0 & a \\ 0 & 2a & a \\ 0 & 0 & a \end{pmatrix} = \begin{pmatrix} 2a & 0 & 0 & a \\ 0 & 2a & 0 & a \\ 0 & 0 & 2a & 0 \\ 0 & 0 & 0 & a \end{pmatrix},$$

which corresponds to a 4D lattice graph. On the other hand, when making the common lift of $PC(2a)$ and $FCC(a)$ it is obtained the matrix

$$\begin{pmatrix} 2a & 0 & 0 \\ 0 & 2a & 0 \\ 0 & 0 & 2a \end{pmatrix} \boxplus \begin{pmatrix} 2a & a & a \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} = \begin{pmatrix} 2a & 0 & 0 & a & a \\ 0 & 2a & 0 & 0 & 0 \\ 0 & 0 & 2a & 0 & 0 \\ 0 & 0 & 0 & a & 0 \\ 0 & 0 & 0 & 0 & a \end{pmatrix},$$

which generates a 5D lattice graph. In this case, the common lift has one extra dimension since the graphs considered belong to different branches of the tree (Figure 2.6). The same happens with the mix $FCC(a)$ and $BCC(a)$, as shown next:

$$\begin{pmatrix} 2a & a & a \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix} \boxplus \begin{pmatrix} 2a & 0 & a \\ 0 & 2a & a \\ 0 & 0 & a \end{pmatrix} = \begin{pmatrix} 2a & a & a & 0 & a \\ 0 & a & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 \\ 0 & 0 & 0 & 2a & a \\ 0 & 0 & 0 & 0 & a \end{pmatrix}.$$

Finally, in Table 2.2 it is presented a selection of lattice graphs composed following the guidelines presented in this section. The table also includes their main topological characteristics. Depending on the focus some of them outperform the others. For example, when looking for a 4-dimensional topology that embeds tori networks, $4D\text{-}BCC(a)$ and $PC(2a)\boxplus BCC(a)$ must be considered. Both topologies equal their number of nodes, so if it is desired to minimize distance properties, $4D\text{-}BCC(a)$ should be a good candidate. On the other hand, if there is interest on 5 dimensions and a great number of different embedded topologies, $PC(2a)\boxplus FCC(a)$ would be a good choice having a good nodes/distance ratio. Therefore, what these examples show is that there is a wide range of possibilities that provides the previous $\boxplus$-operation.

2.3 Routing in Lattice Graphs

Most interconnection networks use routing tables but their size can compromise system scalability. In this section routing algorithms for lattice graphs are presented. In this way,

Topology	Dimension	Nodes	Projection	Diameter	Average Dist.
$T(2a, 2a) \boxplus RTT(a)$	3	$4a^3$	vary	$2a$	$\approx 1.14877a$
$4D\text{-}FCC(a)$	4	$2a^4$	$FCC(a)$	$2a$	$\approx 1.10396a$
$4D\text{-}BCC(a)$	4	$8a^4$	$T(2a, 2a, 2a)$	$2a$	$\approx 1.5379a$
$Lip(a)$	4	$16a^4$	$FCC(2a)$	$3a$	$\approx 1.815a$
$PC(2a) \boxplus BCC(a)$	4	$8a^4$	vary	$2.5a$	$\approx 1.59715a$
$PC(2a) \boxplus FCC(a)$	5	$8a^5$	vary	$3.5a$	$\approx 1.87856a$
$BCC(a) \boxplus FCC(a)$	5	$4a^5$	vary	$2.5a$	$\approx 1.52522a$

Table 2.2: Distance properties of several lattice graphs.

algorithmic routing can be used to avoid the need of tables. If tables are going to be used, the algorithms presented can be employed to fill them.

Routing in circulant graphs was first related to the Closest Vector Problem (CVP) in [CHM+99] for the l_1-norm. Later, this fact was used to optimize a routing algorithm for circulant graphs of degree four in [GGI+05]. Following the same ideas, similar complexity for the CVP can be inferred for routing in lattice graphs. As proved in [DV12] and [DV13], CVP can be solved with asymptotical complexity $2^{O(n)}$. However, algorithms for particular graphs can have lower complexity.

In order to solve the routing problem over lattice graphs there is the need to state which labelling set will be applied. A labelling set is the set that contains the labels for the vertices of the graph. There are many choices for the labelling set. In the 2D case, several approaches to the routing problem have been made in [FB10, Rob96, CMB13]. In those articles, several labellings such as the one given by the fundamental parallelepiped of the lattice, the set of integers modulo N or the set of minimum norm residues have been considered. Anyway, for labelling a lattice graph of dimension n, a subset of $\mathbb{Z}^n$ will be needed.

Definition 2.3.1. *Given a lattice graph $\mathcal{G}(M)$ of dimension n a labelling set of the graph is a set $\mathcal{L} \subset \mathbb{Z}^n$ such that $|\mathcal{L}| = |\det(M)|$ and for every pair $\mathbf{l}_1, \mathbf{l}_2 \in \mathcal{L}$ if $\mathbf{l}_1 \neq \mathbf{l}_2$ then $\mathbf{l}_1 \not\equiv \mathbf{l}_2 \pmod{M}$.*

If $\mathbf{v}_s, \mathbf{v}_d \in \mathcal{L}$, where $\mathbf{v}_s$ labels the source node and $\mathbf{v}_d$ labels the destination node, any vector $\mathbf{r} \in \mathbb{Z}^n$ will be called a **routing record** when

$$\mathbf{v}_d - \mathbf{v}_s \equiv \mathbf{r} \pmod{M}$$

with $\mathbf{v}_d - \mathbf{v}_s \in \mathbb{Z}^n$ such that

$$\mathbf{v}_d - \mathbf{v}_s \in \mathcal{L} - \mathcal{L} = \{\mathbf{x} - \mathbf{y} \mid \mathbf{x}, \mathbf{y} \in \mathcal{L}\}.$$

Each component of a routing record indicates the number of hops in the corresponding dimension and its sign, the direction of the hops. The length of a path associated with a routing record is given by its l_1-*norm*:

$$|\mathbf{r}| = \sum_i |r_i|$$

As minimal routing requires shortest paths, minimum norm routing records should be obtained. Hence, the **routing problem** over $\mathcal{G}(M)$ can be stated as follows:

$$\begin{aligned} \textbf{input}: &\quad \mathbf{v} := \mathbf{v}_d - \mathbf{v}_s \in \mathcal{L} - \mathcal{L} \\ \textbf{output}: &\quad \underset{\mathbf{r} \equiv \mathbf{v} \pmod{M}}{\arg\min} \; (|\mathbf{r}|) \end{aligned}$$

where arg min states for the element in the set $\{\mathbf{r} \in \mathbb{Z}^n \mid \mathbf{r} \equiv \mathbf{v} \pmod{M}\}$ minimizing $|\mathbf{r}|$.

The integral points inside the fundamental parallel give a very useful labelling. For Gaussian integers it was already considered by Huber in [Hub94].

Theorem 2.3.2. *Let M be an integral matrix. The sets*

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{Z}^n \mid 0 \leq (\mathrm{adj}(M)\mathbf{x})_i < |\det(M)|, \textit{ for } 1 \leq i \leq n\}$$

and

$$\mathcal{P}_0 = \{\mathbf{x} \in \mathbb{Z}^n \mid -|\det(M)| \leq 2(\mathrm{adj}(M)\mathbf{x})_i < |\det(M)|, \textit{for } 1 \leq i \leq n\}$$

are both sets of representatives of $\mathbb{Z}^n/M\mathbb{Z}^n$, where $\mathrm{adj}(M) = \det(M)M^{-1}$ is the adjoint matrix of M.

Proof. Geometrically, the set $\mathcal{P}$ is the parallelepiped with vertices in the sum of a subset of columns of M. In order to see that for any $\mathbf{x} \in \mathbb{Z}^n$ there is a unique $\mathbf{x}' \in \mathcal{P}$ such that $\mathbf{x} \equiv \mathbf{x}' \pmod{M}$, apply Euclidean division on $\mathrm{adj}(M)\mathbf{x}$. Thus, it is obtained that for some unique integer vectors $\mathbf{y}$ and $\mathbf{k}$, $\mathrm{adj}(M)\mathbf{x} = \mathbf{y} + \mathbf{k}\det(M)$ with $0 \leq \mathbf{y}_i < |\det(M)|$. Then $\mathbf{x}' = \mathbf{x} - M\mathbf{k} = \mathrm{adj}(M)^{-1}\mathbf{y}$ satisfies both $\mathbf{x}' \in \mathcal{P}$ and $\mathbf{x} \equiv \mathbf{x}' \pmod{M}$.

The second set, $\mathcal{P}_0$, is then obtained by shifting this parallelepiped to center it at the origin. □

Another useful set of representatives is the one of minimum distances. This is,

$$\mathcal{M} = \{ \underset{\mathbf{r} \equiv \mathbf{x} \pmod{M}}{\arg\min} \; (|\mathbf{r}|) \mid \mathbf{x} \in \mathbb{Z}^n\}. \tag{2.3}$$

The interest of this set is clear from the fact that routing can be thought as function from $\mathcal{L} - \mathcal{L}$ into $\mathcal{M}$. This set was already used for Gaussian integers in [FG04].

From a design perspective, it is convenient to label the graph nodes according to their positive coordinates. Hence, it is interesting to consider the labelling given by the Hermite normal form of the generating matrix. Therefore, let H be the Hermite normal form of M and define the set of representatives

$$\mathcal{H} = \{\mathbf{x} \in \mathbb{Z}^n \mid 0 \leq x_i < H_{i,i}\}. \tag{2.4}$$

In the following subsections it will be seen that for the 2D case, lattice reduction solves the problem of routing and additionally gives expressions for the diameter and average distance. Later a hierarchical routing algorithm is presented, which although it has been thought for the crystal graphs, it also works for general lattice graphs.

2.3.1 Distance Properties and Routing of 2D Lattice Graphs

This subsection summarizes the routing and distance properties of 2D lattice graphs. A more extensive discussion can be found in [Cam10] and [CMB13].

With a variation of the concept of *reduced lattice basis* from [KS96] it was obtained that:

Theorem 2.3.3. *For any matrix $M \in \mathbb{Z}^{2\times 2}$ there exists another $M' = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ such that $\mathcal{G}(M) \cong \mathcal{G}(M')$ and*

$$\begin{aligned} |c|, |b| &\leq d \leq a, \\ 2b + c &\leq a, \\ 2c + b &\leq d, \\ 0 &\leq b + c. \end{aligned}$$

The matrix M' is called positive-reduced.

This positive-reduced matrix can be exploited to get expressions for the diameter and average distance.

Theorem 2.3.4. *If $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is a positive-reduced matrix and δ is defined as*

$$\delta = \left\lfloor \frac{a - b - c + d}{2} \right\rfloor.$$

Then, the diameter k of $\mathcal{G}(M)$ is

$$k = \begin{cases} \delta - 1 & \text{if } b = -c,\ N \equiv 1 \pmod 2 \text{ and } a \equiv d \pmod 2; \\ \delta & \text{otherwise.} \end{cases}$$

Example 2.3.5. *The previous theorem provides a closed expression for the diameter of any 2D lattice graph. In addition, it generalizes some other results that can be found in the literature. For Gaussian networks, [Mar07], the diameter of G_{a+bi} is a when $a^2 + b^2$ is even and $a - 1$ otherwise. This can be easily obtained from our result by realizing that the matrix $\begin{pmatrix} a & -b \\ b & a \end{pmatrix}$ is, by default, a positive-reduced one.*

In [TP94], the diameter of the Kronecker product of two cycles of odd lengths a and b, with $a \geq b$, was described. As shown in Theorem 2.2.4, the resulting graph can be seen as the 2D lattice graph with matrix $\begin{pmatrix} \frac{a+b}{2} & \frac{a-b}{2} \\ \frac{a-b}{2} & \frac{a+b}{2} \end{pmatrix} = \begin{pmatrix} p & q \\ q & p \end{pmatrix}$ with $0 \leq q \leq p$, which needs, at most, one division to make it positive-reduced. Clearly, when $3q \leq p$ the matrix is positive-reduced, thus obtaining a diameter of $k = p - q = b$ (or $k = b - 1$ if $q = 0$). Otherwise, the matrix $\begin{pmatrix} p + s(p-q) & p - q \\ s(p-q) - q & p - q \end{pmatrix}$ is the positive-reduced of M for some integer s, obtaining $k = \left\lfloor \frac{p+q}{2} \right\rfloor = \left\lfloor \frac{a}{2} \right\rfloor$. Consequently, the diameter is

$$k = \begin{cases} \max\{b, \left\lfloor \frac{a}{2} \right\rfloor\} & \text{if } a > b \\ b - 1 & \text{if } a = b \end{cases}.$$

For the Kronecker product of cycles of lengths with even parities (or even and odd, respectively), we have not found previous expression in the literature but they can be easily obtained by considering their matrix $\begin{pmatrix} a & -b \\ a & b \end{pmatrix}$ with $a \geq b$; consequently, $k = a$.

The labelling with minimum distances given by Equation (2.3) can be seen in Figure 2.7. All the regions in the figure can be expressed as differences of triangular numbers, and hence, the average distance can be computed:

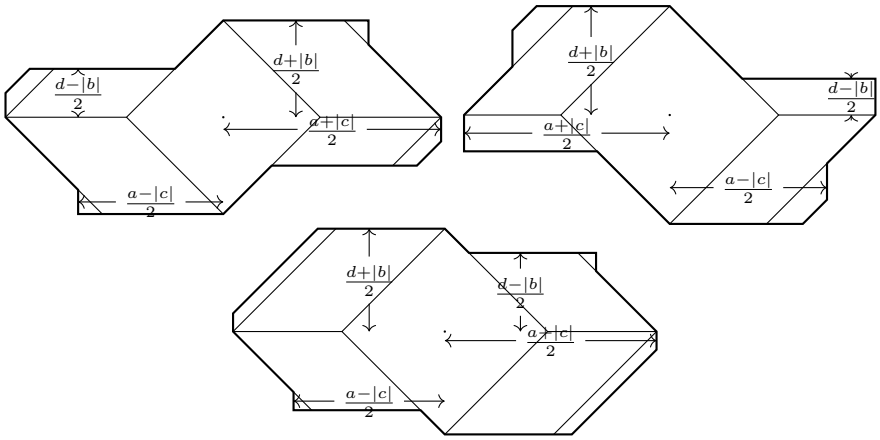


Figure 2.7: Representations with minimum norm, respectively for $b < 0$, $c < 0$ and $0 < b, c$.

Theorem 2.3.6. *Let* $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ *be a positive-reduced matrix. Then, the average distance* $\bar{k}$ *of* $\mathcal{G}(M)$ *is given by*

$$\begin{aligned} 12(|\det(M)| - 1)\bar{k} = &- 6bcd - 6abc + 3a^2d + 3ad^2 \\ &- 3ab^2 + 6b^2c + 6bc^2 - 3c^2d \\ &+ 4b^3 - 4b \quad \textit{if } b > 0 \\ &+ 4c^3 - 4c \quad \textit{if } c > 0 \\ &- 3d \quad \textit{if } a \not\equiv c \pmod 2 \\ &- 3a \quad \textit{if } b \not\equiv d \pmod 2 \\ &+ 6b + 6c \quad \begin{cases} \textit{if } a \not\equiv c \pmod 2 \\ \wedge\ b \not\equiv d \pmod 2. \end{cases} \end{aligned}$$

The minimum labelling depicted in Figure 2.7 can be employed in a geometrical routing. Let $\mathcal{M}$ be the minimum distance labelling of the lattice graph $\mathcal{G}(M)$. It is known that there is a small set S depending only on M with $|S| \leq 11$ such that for any $\mathbf{x}, \mathbf{y} \in \mathcal{M}$ there is $\mathbf{r} \in \mathcal{M}$ and $\mathbf{s} \in S$ such that $\mathbf{y} - \mathbf{x} = \mathbf{r} + M\mathbf{s}$. Thus, routing in 2D lattice graphs can be approached by Algorithm 1. The set S depends on the tilling with copies of $\mathcal{M}$. Its cardinal is usually 7 or 9 but it can get up to 11 because of non-convexity of $\mathcal{M}$. When M is positive-reduced the elements of S have the absolute value of its entries bounded by 2.

Algorithm 1: Routing record in 2D lattice graphs.

Input: Generator matrix $M \in \mathbb{Z}^{2\times 2}$
Precomputed set S with $|S| \leq 11$ that depends only on M
$\mathbf{v}, \mathbf{w} \in \mathbb{Z}^2$ with minimum l_1-norm
Output: $\mathbf{r} \in \mathbb{Z}^2$ with minimum l_1-norm and $\mathbf{r} \equiv \mathbf{w} - \mathbf{v} \pmod M$
R:=$\{\mathbf{w} - \mathbf{v} + M\mathbf{s} \mid \mathbf{s} \in S\}$;
$\mathbf{r}$:=$\arg\min\ l_1\text{-norm}(\mathbf{x})$ for $\mathbf{x} \in R$;
return $\mathbf{r}$;

Example 2.3.7. *This example shows how the routing Algorithm 1 performs in a particular*

2D lattice graph $\mathcal{G}(M)$ *given by* $M = \begin{pmatrix} 2 & -9 \\ 3 & 10 \end{pmatrix}$. *Figure 2.8 shows the representation of this 2D lattice graph in minimum norm representation.*

Note that the positive-reduced of M *is the matrix* $M' = \begin{pmatrix} 15 & 2 \\ -1 & 3 \end{pmatrix}$. *Using the matrix* M' *as generator the minimal set* S *is* $S = \{\mathbf{0}, \pm\mathbf{e}_1, \pm\mathbf{e}_2, \pm(\mathbf{e}_1 - \mathbf{e}_2)\}$ *Now, in order to route from a node* $\mathbf{v}_o = \begin{pmatrix} -6 \\ 2 \end{pmatrix}$ *to a node* $\mathbf{v}_d = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$ *of* $\mathcal{G}(M)$, *it must be found a representative* $\mathbf{r}$ *with minimum norm of* $\mathbf{v}_d - \mathbf{v}_o = \begin{pmatrix} 4 \\ -1 \end{pmatrix}$. *Therefore, it is computed*

$$\begin{aligned} R &= \{\mathbf{v}_d - \mathbf{v}_o + M'\mathbf{s} \mid \mathbf{s} \in S\} \\ &= \left\{ \begin{pmatrix} 6 \\ 2 \end{pmatrix}, \begin{pmatrix} 19 \\ -2 \end{pmatrix}, \begin{pmatrix} -9 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ -1 \end{pmatrix}, \begin{pmatrix} 17 \\ -5 \end{pmatrix}, \begin{pmatrix} -11 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ -4 \end{pmatrix} \right\}. \end{aligned}$$

Finally, the one with minimum l_1*-norm is* $\mathbf{r} = \begin{pmatrix} 4 \\ -1 \end{pmatrix}$, *which corresponds to the routing record. In Figure 2.9 a drawing of the graph and the tessellation using* $\mathcal{M}$ *as tiles. Note that the 7 tiles given by the set* S *are enough to cover the set in grey color, which correspond to the set of differences of the minimum norm labellings.*

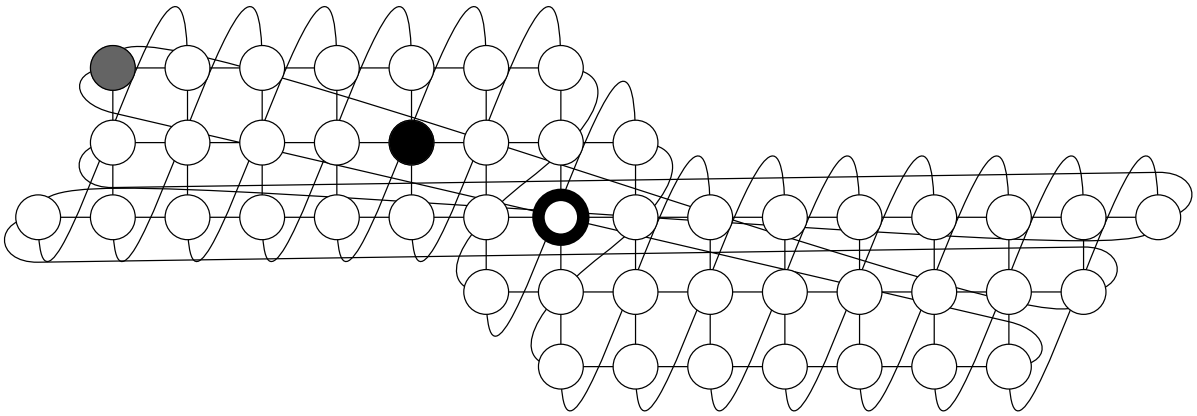


Figure 2.8: Routing example of a packet in a 2D lattice graph with minimum norm labelling.

Although Algorithm 1 already has constant complexity, for some cases there are more elegant algorithms with a lower constant. For example, the routing in a 2D torus $T(a_1, a_2)$ can be done by two comparisons or explicitly by $\mathbf{r} = (\mathrm{rem}(x + \frac{a_1}{2}, a_1) - \frac{a_1}{2}, \mathrm{rem}(y + \frac{a_2}{2}, a_2) - \frac{a_2}{2})^t$, and the routing in $RTT(a)$ can be done by Algorithm 2.

Algorithm 2: Routing in $RTT(a)$.

Input: $x, y := \mathbf{v}_d - \mathbf{v}_s$
Output: $\mathbf{r}$ routing record
$p := \mathrm{rem}(x + y + a, 2a)$;
$q := \mathrm{rem}(y - x + a, 2a)$;
$x' := (p - q)/2$;
$y' := (p + q - 2a)/2$;
$\mathbf{r} := (x', y')^t$;

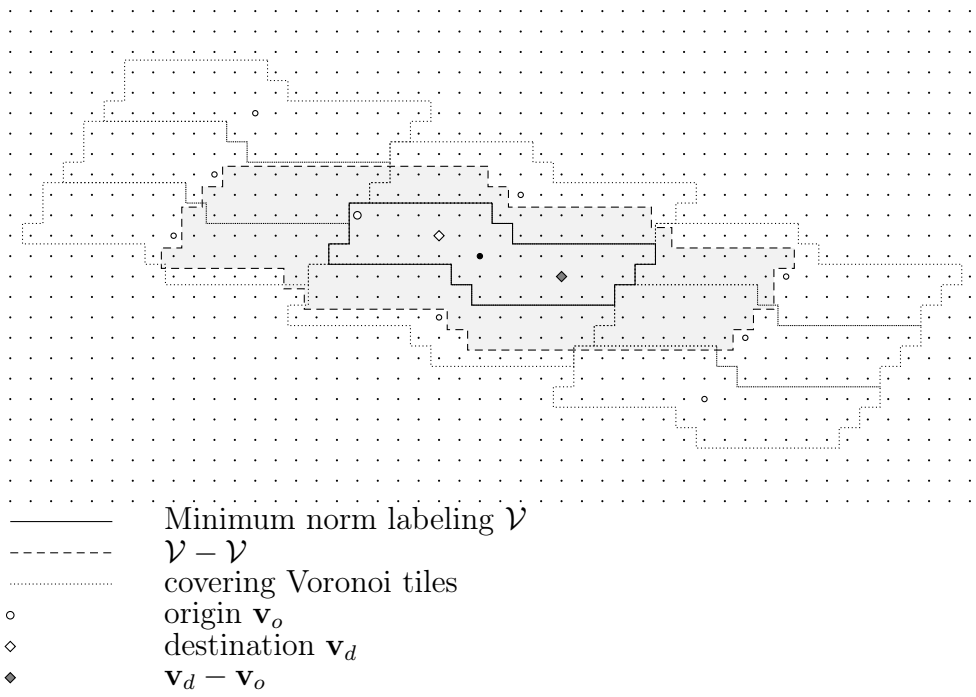


Figure 2.9: Routing in a 2D lattice graph with minimum norm labelling.

2.3.2 A Hierarchical Routing for Lattice Graphs

Now, a routing algorithm is proposed based on the hierarchy induced by the projecting operation. The idea is that routing in a lifted graph can be done by routing in its projection and in the cycle that joins the disjoint projections. First, the node labelling to be adopted is stated and then the general hierarchical routing is presented. Finally, complexity and implementation aspects are considered.

Remember that the lattice graph $\mathcal{G}(M)$ with $M \cong \begin{pmatrix} B & \mathbf{c} \\ 0 & a \end{pmatrix}$ has a disjoint copies of its projection $\mathcal{G}(B)$ embedded, which are connected by $\frac{|\det(M)|}{ord(\mathbf{e}_n)}$ parallel cycles. The cycles have length $ord(\mathbf{e}_n)$. The number of vertices belonging to a cycle that lies in the same copy of $\mathcal{G}(B)$ is $\frac{ord(\mathbf{e}_n)}{a}$. Hence, the elements of the routing record can be considered separately in the following way:

Proposition 2.3.8. *Let $M \cong \begin{pmatrix} B & \mathbf{c} \\ 0 & a \end{pmatrix}$. Then, a labelling set $\mathcal{L}_M$ of the lattice graph $\mathcal{G}(M)$ can be obtained from a labelling set $\mathcal{L}_B$ of its projection $\mathcal{G}(B)$ as*

$$\mathcal{L}_M = \left\{ \begin{pmatrix} \mathbf{x} \\ y \end{pmatrix} \mid \mathbf{x} \in \mathcal{L}_B, 0 \leq y < a \right\}.$$

If this is done recursively, then the labelling obtained is the one defined by Equation (2.4), denoted by $\mathcal{H}$. Now, it is possible to give the following main result:

Theorem 2.3.9. *If $[\mathcal{G}(B)]_y$ is the projection $\mathcal{G}(B)$ of $\mathcal{G}(M)$ that contains $y\mathbf{e}_n$, $\mathcal{C}$ denotes the cycle generated by $\mathbf{e}_n$ and, given a vertex $\mathbf{v} \in \mathbb{Z}^n$, $\mathbf{v} + \mathcal{C}$ denotes the translation of the cycle to this vertex. Algorithm 3 gives minimum routing records in any lattice graph.*

Algorithm 3: Hierarchical routing in lattice graphs.

Input: $\mathbf{v}_s$ source, $\mathbf{v}_d$ destination
Output: $\mathbf{r}$ minimum routing record from $\mathbf{v}_s$ to $\mathbf{v}_d$
Let y be the last component of $\mathbf{v}_d$;
$\mathbf{v}_s + \mathcal{C}$ is the cycle translated to $\mathbf{v}_s$;
foreach *vertex* $\mathbf{c}_i$ *of the cycle in the copy* $[\mathcal{G}(B)]_y$ **do**
  $r_i^{\mathcal{C}}$: Route in the cycle from $\mathbf{v}_s$ to vertex $\mathbf{c}_i$;
  $\mathbf{r}_i^{\mathcal{G}(B)}$: Route in $[\mathcal{G}(B)]_y$ from $\mathbf{c}_i$ to $\mathbf{v}_d$;
end
Return the routing record that minimizes the weight of $\begin{pmatrix} \mathbf{r}_i^{\mathcal{G}(B)} \\ r_i^{\mathcal{C}} \end{pmatrix}$;

Proof. Since the algorithm composes routing records from two subgraphs, then the result is indeed a routing record.

In order to see that the minimum one is found, let $\mathbf{r}^{\min}$ be one of the routing records with minimum norm. Since $\mathbf{v}_s + \mathbf{r}_n^{\min}$ is in the cycle mentioned in the algorithm, then there is an index i such that $\mathbf{r}_n^{\min}$ is the minimum route in the cycle from $\mathbf{v}_s$ to $\mathbf{c}_i$. As $\mathbf{r}^{\min}$ is minimal, the minimal routing from $\mathbf{c}_i$ to $\mathbf{v}_d$ does not use the n dimension. Thus, routing in $[\mathcal{G}(B)]_y$ gives the minimum. By composing both, the algorithm finds the minimum routing $\mathbf{r}^{\min}$ and returns it or another one with same norm. □

Remark 2.3.10. *In the last step of Algorithm 3 there can sometimes be several routing records with the same weight. In this case it is advisable to choose one of them at random, thus balancing the use of the paths.*

Remark 2.3.11. *Let* $\mathcal{G}(M)$ *be a lattice graph with* $M \cong \begin{pmatrix} B & \mathbf{c} \\ 0 & a \end{pmatrix}$. *Clearly, the complexity of Algorithm 3 is* $O(C\frac{ord(\mathbf{e_n})}{a})$, *where* C *denotes the complexity of routing in* $[\mathcal{G}(B)]_y$. *If routing is done with the same algorithm by means of recursive calls, the final complexity would be* $O(\prod_{i=1}^{n} \frac{ord(\mathbf{e_i}, M_i)}{a_i})$, *where* M_i *are the successive projections,* a_i *denotes the side of* $\mathcal{G}(M_i)$ *and* $ord(\mathbf{e_i}, M_i)$ *the order of* $\mathbf{e_i}$ *in* $\mathcal{G}(M_i)$. *In the worst case, this complexity would attain* $O(\det(M)^n)$. *However, in some families the order of* $\mathbf{e_i}$ *is upper bounded, thus obtaining good complexities for this recursive version of the routing algorithm, as it will be seen in the following section.*

Routing Discussion of Crystal Lattice Graphs

The above ideas can be used to calculate the routing complexity for the crystal lattice graphs and its lifts. As it has be seen, $\frac{ord(\mathbf{e}_n)}{a}$ determines the number of intersections of the cycle with the destination projection, which dictates the number of nested routing calls. First, $\frac{ord(\mathbf{e}_n)}{a} = 1$ in nD-PC. Second, $\frac{ord(\mathbf{e}_n)}{a} = 2$ in nD-BCC and nD-FCC. Clearly, these are constant values, which imply just one or two calls to the routing of dimension $n-1$ in Algorithm 3. Therefore, if the algorithm is used in a recursive form, it follows that:

- The routing in nD-PC can be done immediately with n comparisons in parallel.
- The hierarchical routing in nD-BCC requires 2 calls to the routing algorithm for $(n-1)D$-PC.

- The hierarchical routing in $nD\text{-}FCC$ requires 2 calls to the $(n-1)D\text{-}FCC$, which accumulates into 2^{n-2} calls to $2D\text{-}FCC$ or RTT. These last routing calls will be performed by Algorithm 2.

As it was seen in Subsection 2.1.1, hybrid graphs are obtained as common lifts of different lattice graphs. Therefore, given a hybrid graph $\mathcal{G}(M)$ there would be several possible lattice graphs that could be considered as its projection. Since the heaviest computation part in Algorithm 3 corresponds to the routing calls in the projection, that projection should be carefully chosen. For example, let $\mathcal{G}(M)$ be given by

$$M = \begin{pmatrix} 2a & 0 & 0 & a \\ 0 & 2a & 0 & a \\ 0 & 0 & 2a & 0 \\ 0 & 0 & 0 & a \end{pmatrix}.$$

This graph, as previously seen, is obtained as the common lift of $PC(2a)$ and $BCC(a)$. Clearly, taking $BCC(a)$ as the projection, would complicate the routing function. Hence, $PC(2a)$ should be chosen as projection, in which dependencies among dimensions do not exist and routing will be less laborious.

Now, Algorithm 3 is specialized for cubic crystal graphs; stating precisely the operations to work with the labelling $\mathcal{H}$, defined in Equation (2.4). Since routing in PC is widely known, only the particular cases of FCC and BCC are considered.

As it has been previously seen, $FCC(a)$ is defined as the lattice graph generated by

$$\begin{pmatrix} 2a & a & a \\ 0 & a & 0 \\ 0 & 0 & a \end{pmatrix}$$

is isomorphic to the PDTT presented in [CMV+10], where a generic graph routing was used. As can be observed, its projection is the graph with matrix $\begin{pmatrix} 2a & a \\ 0 & a \end{pmatrix}$, denoted as $RTT(a)$ in [CMV+10]. It is easy to verify that the order of $\mathbf{e}_n$ is $2a$, which implies that the cardinal of the intersection between $v_s + \mathcal{C}$ and $[\mathcal{G}(B)]_y$ is 2, that is, two calls to $route_B$ are needed. Using this mechanism Algorithm 4 is obtained for $FCC(a)$. Note that the product by a Boolean is defined as $a \cdot true = a$ and $a \cdot false = 0$. An algorithm for routing in the projected 2D graph can be seen in Algorithm 2 and it has been introduced in [CVM+13].

Remark 2.3.12. *When a is a power of 2, the starting arithmetic operations are easier to calculate as $rem(y, a)$, $rem(z, a)$ and $rem(\hat{x}, 2a)$.*

Example 2.3.13. *As an example consider the lattice graph $FCC(4)$. The labelling used for this graph is*

$$\mathcal{L} = \{(x, y, z)^t \mid 0 \leq x < 8,\ 0 \leq y, z < 4\}.$$

If there is need to route from $\mathbf{v}_s = (1, 3, 3)^t$ to $\mathbf{v}_d = (6, 0, 1)^t$, first $\mathbf{v} = \mathbf{v}_d - \mathbf{v}_s = (5, -3, -2)^t$ is computed, which is in the set of differences:

$$\mathbf{v} \in \mathcal{L} - \mathcal{L} = \{(x, y, z)^t \mid -8 < x < 8,\ -4 < y, z < 4\}.$$

*According to Algorithm 4, since that $y = -3 < 0$ and $z = -2 < 0$ these values have to be modified as $y' = -3 + 4 = 1$ and $z' = -2 + 4 = 2$. Moreover, since $(-3 < 0)$**xor**$(-2 <$*

Algorithm 4: Routing in $FCC(a)$.

Input: $(x,y,z)^t := \mathbf{v}_d - \mathbf{v}_s \in \mathcal{L} - \mathcal{L}$
Output: $\mathbf{r}$ minimum routing record from $\mathbf{v}_s$ to $\mathbf{v}_d$
$y' := y + a(y < 0)$;
$z' := z + a(z < 0)$;
$\hat{x} := x + a\big((y < 0)\mathbf{xor}(z < 0)\big)$;
$x' := \hat{x} + 2a(\hat{x} < 0) - 2a(\hat{x} \geq 2a)$;
It is hold that $(x', y', z')^t \in \mathcal{L}$;
$\mathbf{r}_1^{\mathcal{G}(B)} := route_{\left(\begin{smallmatrix} 2a & a \\ 0 & a \end{smallmatrix}\right)}(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} x' \\ y' \end{pmatrix})$;
$\mathbf{r}_2^{\mathcal{G}(B)} := route_{\left(\begin{smallmatrix} 2a & a \\ 0 & a \end{smallmatrix}\right)}(\begin{pmatrix} a \\ 0 \end{pmatrix}, \begin{pmatrix} x' \\ y' \end{pmatrix})$;
$\mathbf{r} := \arg\min(|\mathbf{k}| \mid \mathbf{k} \in \{\begin{pmatrix} \mathbf{r}_1^{\mathcal{G}(B)} \\ z' \end{pmatrix}, \begin{pmatrix} \mathbf{r}_2^{\mathcal{G}(B)} \\ z' - a \end{pmatrix}\})$;

Algorithm 5: Routing in $BCC(a)$.

Input: $(x,y,z)^t := \mathbf{v}_d - \mathbf{v}_s \in \mathcal{L} - \mathcal{L}$
Output: $\mathbf{r}$ minimum routing record from $\mathbf{v}_s$ to $\mathbf{v}_d$
$z' := z + a(z < 0)$;
$\hat{x} := x + a(z < 0)$;
$\hat{y} := x + a(z < 0)$;
$x' := \hat{x} + 2a(\hat{x} < 0) - 2a(\hat{x} \geq 2a)$;
$y' := \hat{x} + 2a(\hat{y} < 0) - 2a(\hat{y} \geq 2a)$;
It is hold that $(x', y', z')^t \in \mathcal{L}$;
$\mathbf{r}_1^{\mathcal{G}(B)} := route_{\left(\begin{smallmatrix} 2a & 0 \\ 0 & 2a \end{smallmatrix}\right)}(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} x' \\ y' \end{pmatrix})$;
$\mathbf{r}_2^{\mathcal{G}(B)} := route_{\left(\begin{smallmatrix} 2a & 0 \\ 0 & 2a \end{smallmatrix}\right)}(\begin{pmatrix} a \\ a \end{pmatrix}, \begin{pmatrix} x' \\ y' \end{pmatrix})$;
$\mathbf{r} := \arg\min(|\mathbf{k}| \mid \mathbf{k} \in \{\begin{pmatrix} \mathbf{r}_1^{\mathcal{G}(B)} \\ z' \end{pmatrix}, \begin{pmatrix} \mathbf{r}_2^{\mathcal{G}(B)} \\ z' - a \end{pmatrix}\})$;

$0) \equiv false$ *it is obtained that* $\hat{x} = x = 5$. *Finally, as* $0 \leq 5 < 8$ *this implies* $x' = 5$ *and* $\mathbf{v} \equiv (5,1,2)^t \in \mathcal{L}$.

Now, in $RTT(a)$ *a minimum route from* $(0,0)^t$ *to* $(5,1)^t$ *is* $(1,-3)^t$ *and a minimum route from* $(4,0)^t$ *to* $(5,1)^t$ *is* $(1,1)^t$. *Consequently,* $\mathbf{r}_1 = (1,-3,2)^t$ *and* $\mathbf{r}_2 = (1,1,-2)^t$. *Finally, after comparing the two norms* $|\mathbf{r}_1| = 6$ *and* $|\mathbf{r}_2| = 4$, *we find that the minimum routing record to reach* $\mathbf{v}_d$ *from* $\mathbf{v}_s$ *is given by* $\mathbf{r} = \mathbf{r}_2$.

Similarly, for the network $BCC(a)$, Algorithm 5 is obtained. Again, the order is $ord(\mathbf{e}_n) = 2a$, which implies 2 calls to the routing of a 2D torus $T(2a, 2a)$.

2.4 Layout

This section considers how the physical implementation of these networks clearly can be done in a room. The computing units will be arranged in *node boards* and several node

boards will lay in a *rack*. Additionally, the half of the node boards in a rack will receive the name of *midplane*.

It is not difficult to conceive a package hierarchy and a 3D physical organization to deploy systems based on lattice graphs. For illustrating this organization, let us first consider the approaches followed by manufacturers. Cray uses a straightforward structure. For example, an actual configuration [Bla09], was a $T(25, 32, 16)$ packaged on a 200 rack system arranged as an 8×25 rectangle. The system can be seen as:

- The full system composed of $25 \times 8 \times 1$ racks.
- Each rack composed of $1 \times 4 \times 16$ nodes (routers).

That is, the third dimension is completely inside the racks and the first dimension is formed entirely joining racks. However the second dimension is partially inside the rack and requires connecting rack columns by rings. Taking into account forthcoming improvements in integration and packaging technologies, it could be expected that a 4D torus would have two dimensions internal to the racks and the other 2 external to the racks. This idea generalizes to lattice graphs. If $\mathcal{G}(M)$ is a 4D lattice graph, its 2D projections would be built inside racks, which would be a torus or a twisted torus. Then it becomes a question of completing the lattice by adjusting the offsets of the cables connecting the racks. Moreover, folding techniques for 3D networks presented in [CMV+10] can also be of application in our case and easily generalized to higher dimensions.

IBM presents a more elaborated organization in the Blue Gene family [CBC+05]. Although the complete network is a torus, each midplane (half of a rack) has additional edge hardware that enables the midplane to disconnect from the remainder of the network and to become itself a small torus. By arranging several midplanes, this additional hardware enables a multitude of different tori shapes to be connected. With slight modifications to such hardware, it would be possible to allow each group to be a symmetric crystal (or another lattice if desired) instead of a mixed-radix torus. This hardware changes its configuration only between different application runs. Then, the potentially added functionality would not have any negative impact on the system.

In this chapter different topologies have been proposed of different degrees as alternatives to current HPC toroidal networks. Since such a system is used by multiple users through partitions, the starting point in this study was to build machines having as natural partitions the cubic crystals graphs developed in Section 2.2.

In this section both the physical layout and partitioning problems of the proposed networks are considered. There are mainly two companies in the market that stand out the development of interconnection networks whose topologies correspond to torus networks: Cray and IBM. Hence, both the physical layout and partitioning problems are considered according to what these two companies do. Therefore, in the first part of this section it is studied how are layout and partitioning problems solved in the Blue Gene family and general guidelines are given for embedding crystal graphs instead of plain tori. In the second part of this section the physical layout of two lattice graphs of dimensions 4 and 5 will be developed, mimicking what it is used in the Cray family for the XT5 supercomputer.

2.4.1 Layout and Partitioning: Cray Technology

Tori-based Cray systems are classified in which they call *network topology classes*; there are four of these classes:

- Class 0: 1 row, 1 to 3 racks
- Class 1: 1 row, 4 to 16 racks
- Class 2: 2 equal rows, 14 to 48 racks
- Class 3: 48 to 320 racks
  - 3 equal rows: mesh in some dimensions
  - An even number of rows greater or equal to 4: torus in all dimensions

Class 3 is the more interesting to us—when it is a whole torus.

An actual configuration was a 200 rack system in 8 rows, totalling a torus of $25 \times 32 \times 16$ nodes.

- Racks of $1 \times 4 \times 16$ nodes.
- System of $25 \times 8 \times 1$ racks.

This is, the third dimension is totally inside the racks and the first dimension is done entirely joining racks. However the second dimension is partially inside the rack. Following that policy it could be expected that 4D torus would have two dimensions internal to the racks and the other 2 dimensions external to the racks.

With this idea is very easy to make a network based on any other lattice graph. As example, a layout of $PC(8) \boxplus BCC(4)$ is shown in Figure 2.10. We have proposed racks which contain a torus of $8 \times 8 \times 1 \times 1$ nodes and a full system of $1 \times 1 \times 8 \times 4$ racks. In the left part of the figure, the system is shown as a connection of 8×4 racks making a 2D torus. The only twist is made inside of some of the $\mathbf{e}_4$ cables. The cables indicated with a mark ($\dashrightarrow$) connect a node $\begin{pmatrix} a \\ b \end{pmatrix}$ in a rack with the node $\begin{pmatrix} a+4 \pmod 8 \\ b+4 \pmod 8 \end{pmatrix}$ of the other rack. Hence, this is shown in right part of Figure 2.10, where solid lines indicate the links belonging to the cycle $\langle \mathbf{e}_4 \rangle$ and with dashed lines a parallel cycle.

The partitions given to users in the machines of Cray are generally meshes. When a partition has a full dimension then it becomes a cycle. Hence the projections of the full machine are the toric partitions that they offer. Hence, using $PC(2a) \boxplus BCC(a)$ or other of the mentioned in Subsection 2.1.1 would be very useful to provide the users with a variety of symmetric graphs.

2.4.2 Layout and Partitioning: IBM Technology

This subsection begins explaining how is the layout of IBM's implementations. Later it shows how it can be extended for lattice graphs other than tori.

The topologies of the Blue Gene families (BGs) have been all tori and the typical configurations are:

- For BG/L or BG/P a system of $72 \times 32 \times 32$ nodes with midplanes of $8 \times 8 \times 8$.
- For BG/Q, the Sequoia supercomputer has $16 \times 16 \times 16 \times 12 \times 2$ node boards with midplanes of $4 \times 4 \times 4 \times 4 \times 2$ [BKM+10].

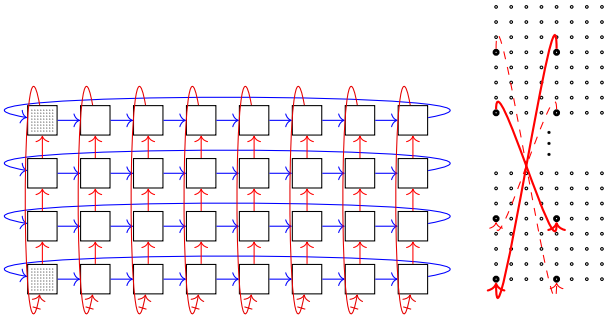


Figure 2.10: Cray-like physical layout of $PC(8) \boxplus BCC(4)$.

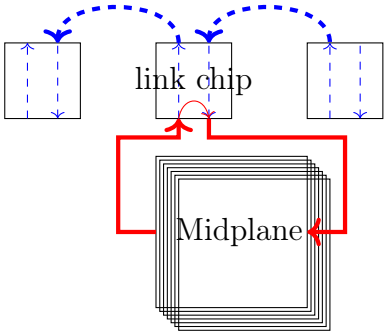


Figure 2.11: Connecting a midplane to itself and to others.

Usually, the partition assigned to a user will be a multiple of a midplane. Smaller partitions will not be considered in this study. The following will deal with the connections between midplanes, hence the last dimension, of side 2, of the BG/Q, will be ignored, and its topology treated as a four dimensional one. Each midplane has connections to be itself a torus of the same degree than the complete topology. This is done using additional links and specific hardware, which will be called *link chip*, that selects between the used links. There will be a link chip for each dimension in each midplane. In a typical configuration, the BG/Q has cycles of 4 midplanes, in which each midplane has connection to itself and to the next midplane, as represented in Figure 2.11. Therefore, the configuration considered in the figure will allow the midplane to be itself a torus. Also, a different configuration or *mode* of the link chip would allow to connect adjacent midplanes to form a larger torus.

In the BG/L the cycles are of 8 midplanes, as shown in Figure 2.12. To allow several cycles formed by less than 8 midplanes, there exist extra links or *split-redirection cables* which are represented by dashed lines in the figure, [CBC+05].

The link chips of BG/Q have 4 ports (out to midplane, in from midplane, in from previous midplane, out to next midplane) while in the BG/L they have 6 ports (the same

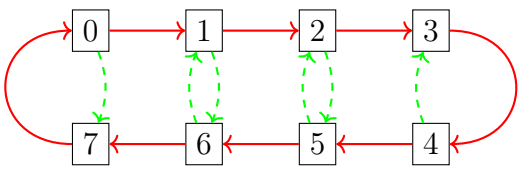


Figure 2.12: Split-redirection cables in BG/L.

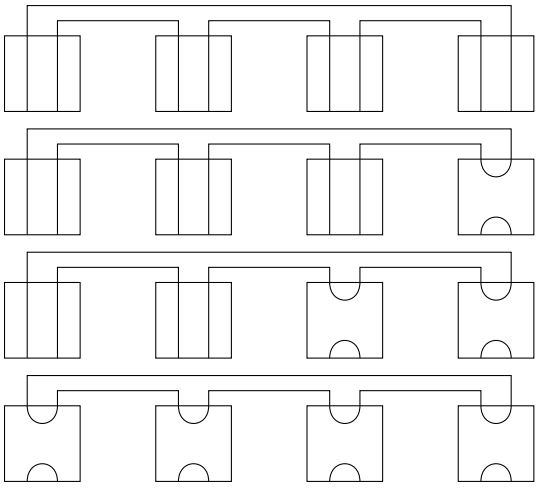


Figure 2.13: The 4 partitions available in the BG/Q in every dimension.

as the BG/Q plus two new ports: in split cable, out split cable). Each link chip has several modes and in each mode a different pair of ports is connected. The change of mode is made each time a partition is allocated to an user. The potential number of modes for $2n$ ports, that is, the number of possible pairings is $\prod_{k=1}^{n}(2k-1)$. Therefore, for 4 and 6 ports the number of potential modes are 3 and 15, respectively. However BG/Q only uses 2 modes and BG/L only uses 9 modes, which are the only modes which provide useful connectivity.

For each midplane and dimension there is a link chip[2]. The choice of a mode for every link chip determines the partitioning of the full machine into smaller tori.

A look for the cycles that can be made in BG/Q shows that only 4 configurations can be made (see Figure 2.13), albeit the integer 4 has 5 partitions—ways to write it as a sum of positive integers[3]: $4 = 3+1 = 2+1+1 = 1+1+1+1 = 2+2$. The partition $2+2$ (which would correspond to 2 tori, each torus of 2 midplanes) is impossible to make since the first 2-midplane torus uses all the inter-midplane links. In the BG/L the split-redirection cables mitigate this problem (which with cycles of 8, which has 22 partitions, is more important) and only remains a problem with partitions of odd length [LK12].

In the following it is considered how to obtain symmetric partitions in the Blue Gene family. The partitions assigned to a user are a multiple of a midplane. Thus, it will be shown how to make this partitions to have the topology of any cubic crystal graphs considered in previous sections. For the sake of simplicity it begins with dimension $n = 2$. Since tori can be done as in the Blue Gene family, it is studied how to make partitions based on $\mathcal{G}(\begin{pmatrix} 2a & a \\ 0 & a \end{pmatrix})$. The link chips associated to the dimension $\mathbf{e}_1$ will be as in the Blue Gene. On the other hand, the ones corresponding to dimension $\mathbf{e}_2$ will have two additional ports. In Figure 2.14 it can be seen how to select modes of some link chips to make a partition based on $\mathcal{G}(\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix})$, were the cables with twist do not impose more restrictions in the mode of links outside the partition than which is imposed by the normal cables. In

[2]Actually in the BG/Q the link chip is inside of every node board, but all the node boards of the same midplane are always in the same mode [Mil12].

[3]These are the partitions with the meaning of number theory. Nevertheless, they coincide with the potential physical partitions in one dimension.

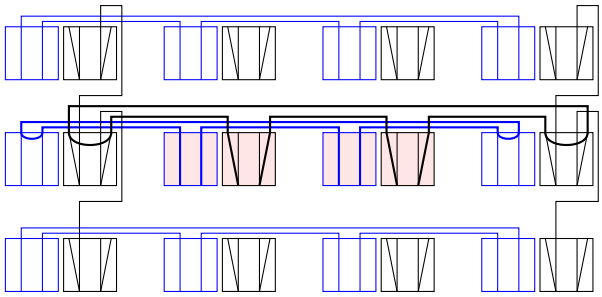


Figure 2.14: Building a RTT of two midplanes.

the figure the two link chips of the midplanes are represented: the left one associated to the first dimension and the right one associated to the second dimension. The partition is colored and the cables used are in bold. Additionally it is possible to build a $\mathcal{G}(\begin{pmatrix} 4 & 2 \\ 0 & 2 \end{pmatrix})$, as in Figure 2.15, forcing link chips outside the partition in similar way to how is done for a typical torus in the BGs. As a consequence, a user who wants to run an application in a partition formed by 2^k midplanes could always obtain a symmetric one, thus getting a torus for k even and a RTT when k odd.

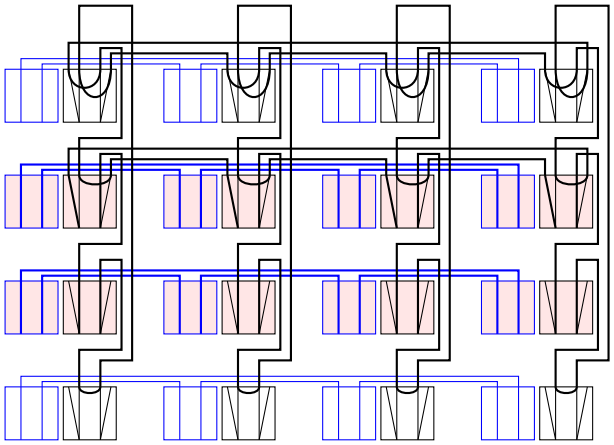


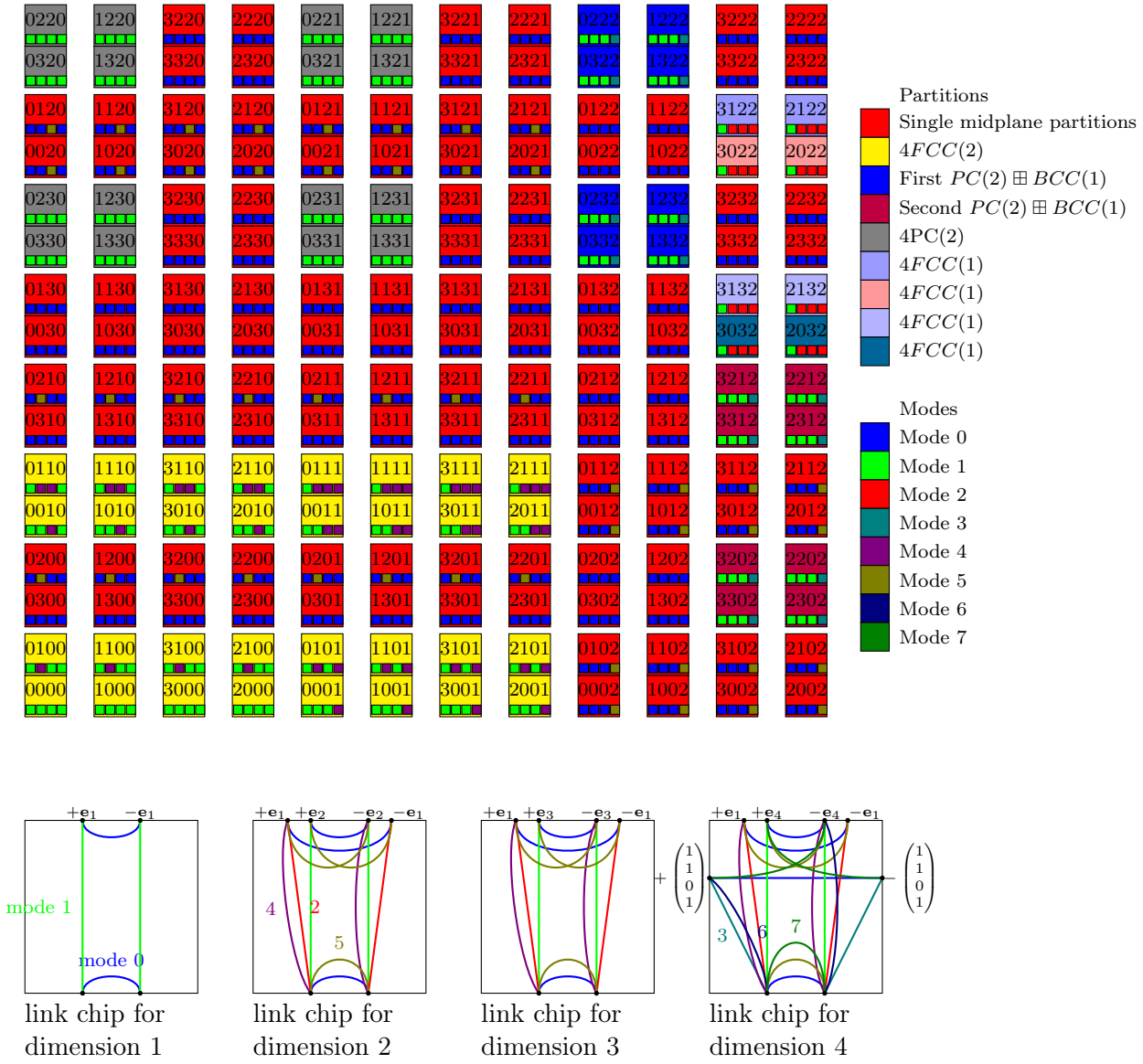
Figure 2.15: Building a RTT of eight midplanes.

For more dimensions the same idea can be realized. The link chips of the dimension $\mathbf{e}_1$ are always done like BG, that is, 4 ports for length 4 and 6 ports for length 8. The other link chips will have additional ports to make the twists. If it is desired to allow t additional topologies to the tori it is required to add at most $2t$ ports to each link chip from $\mathbf{e}_2$ to $\mathbf{e}_n$. In particular, when $n = 3$ one can decide to allow the topologies PC, FCC and BCC which are symmetric and in a whole they allow to users to obtain partitions with any power of two as size.

However, for current machines using $n = 4$ (like BG/Q in terms of midplanes) it is not possible to give all powers of 2 giving only lattice graphs with all the possible symmetries. Nevertheless, there are still a large selection of graphs from which to build the partitions. For example the following partitions are possible:

i) A $4PC$ for 2^{4k} nodes,

ii) A $4FCC$ for 2^{4k+1} nodes,



link chip for dimension 1 | link chip for dimension 2 | link chip for dimension 3 | link chip for dimension 4

Figure 2.16: Physical layout and partitioning example.

iii) A $\mathcal{G}\left(\begin{pmatrix} 2a & 0 & a & a \\ 0 & 2a & a & a \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & a \end{pmatrix}\right)$ for 2^{4k+2} nodes, although it is not symmetric and only has the crystal BCC as its projection,

iv) A $4BCC$ for 2^{4k+3} nodes; or maybe $T(2a, 2a, 2a) \boxplus BCC(a)$, which can be useful if the users have 3D applications.

Figure 2.16 shows a physical layout of a complete system similar to the Sequoia supercomputer which allows as partitions the topologies $4PC$, $4FCC$ and $T(2a, 2a, 2a) \boxplus BCC(a)$. This figure is done in a similar way to the one found in [CBC+05]. In the bottom of the figure there are the four classes of link chips, each corresponding to a dimension. The ports are shown as black dots, with a label that indicates to which midplane it is connected. The ones labelled with $\mathbf{e}_i$, where i is its dimension, were already found in the BG/Q, the

others have been added to allow the new topologies. Each line color corresponds to a mode of the link chip (following the legend in the right side), with lines connecting pairs of ports. In the central part of the figure it is shown the physical layout of the system. It is distributed in the same way than the Sequoia, with 12 rows of 8 racks each one. The racks are divided in 2 midplanes, which are represented with a square with 4 numbers corresponding to its logical position. The layout has been folded, as it would be done in a realistic scenario to keep the length of the cables bounded. For example, the midplane numbered as 1230 correspond to the vector $\begin{pmatrix}1\\2\\3\\0\end{pmatrix}$ over the system $\begin{pmatrix}4&0&0&0\\0&4&0&0\\0&0&4&0\\0&0&0&3\end{pmatrix}$ (in terms of midplanes). Furthermore, the figure shows a partitioning of the system. The color of a midplane indicates the partition to which it belongs (following the colors in the legend) with the special case of the isolated midplanes, which are shown in red. The partitioning is defined by the selection of the modes of the link chips. In each midplane 4 small colored squares are drawn to indicate the mode of the respective link chips.

Finally it should be noted that additional resources have been added over the resources of Sequoia to embed additionally a $4FCC$ and a $T(2a, 2a, 2a) \boxplus BCC(a)$. Thus, the present configuration allows the previous partitions and these new ones. This has been done by changing 3 of the 4 classes of link chips and doubling the number of cables between midplanes. Clearly, the fewer graphs which are allowed the fewer resources should be added.

2.5 Conclusions

This chapter has been focused on the study and proposal of new multidimensional twisted torus interconnection networks. Due to their complex spatial characteristics, their analysis is far from being straightforward. Nevertheless, it had been taken advantage of an algebraic tool based on integral square matrices presented in [Fio95]. Such matrices define the graph and its topological characteristics. Adequate algebraic manipulations of the matrices enable a better understanding of different network properties. For example, when using the Hermite normal form, matrices reveal the subgraphs naturally embedded in the network.

It has been seen that lattice graphs englobe many previously considered regular network topologies. As it has been proved, the lattice graph family includes tori and twisted tori, circulant graphs, chordal rings, Gaussian graphs, Kronecker products of cycles, Midimews and many other graphs.

Using this tool, several networks have been proposed and analyzed in this chapter. Two graph lifting methods have been introduced that allow for higher dimensional networks that embed lattice subnetworks. Complementarily, the use of graph projections facilitates the conception of routing algorithms for these networks. Based on this graph operation, minimal routing schemes have been proposed for all the topologies. Then, the focus is on 3D symmetric networks as alternatives to mixed-radix tori that are not edge-symmetric. Taking the matrices that define cubic crystallographic lattices, it was possible to evaluate and compare their associated interconnection networks. If symmetry is desired, the best path when upgrading 3D systems clearly seems to be $PC(a) \rightarrow FCC(a) \rightarrow BCC(a) \rightarrow PC(2a)$, that is, duplicating the machine size on each step and maintaining most of the original connections. Although the focus has been on typical network configurations derived from powers of two, the results remain valid for any other network size. For bidimensional

lattice graphs closed expressions for their diameter and average distance has been provided, with the addition of an optimal routing algorithm.

The chapter addresses some practical issues. Physical packaging and system organization in racks have been taken into account, concluding that, for deploying networks based on lattice graphs, very few changes over typical tori would be necessary. In addition to the algebraic analysis carried out through the chapter, an empirical evaluation of different interesting topologies has been considered. The evaluation in Section 5.5 will certify that hyper-dimensional twisted tori clearly outperform their standard (not twisted) counterparts for sizes of current machines. Noticeable gains are exhibited by twisted lattice topologies for both configurations under consideration.

Chapter 3

Hamming and Dragonfly Networks

Part of the current HPC and most datacenter networks rely on large-radix routers. Hamming graphs (Cartesian products of complete graphs) and dragonflies (two-level direct networks with nodes organized in groups) are some direct topologies proposed for such networks. The original definition of the dragonfly topology is very loose, with several degrees of freedom such as the inter- and intra-group topology, the specific global connectivity and the number of parallel links between groups (or trunking level).

This chapter provides a comprehensive analysis of the topological properties of the dragonfly network, providing balancing conditions for network dimensioning, as well as introducing and classifying several alternatives for the global connectivity and trunking level. From a topological study of the network, it is noted that a Hamming graph can be seen as a canonical dragonfly topology with a large level of trunking. Based on this observation and by carefully selecting the global connectivity, the Dimension Order Routing (DOR) mechanism safely used to avoid deadlock in Hamming graphs is adapted to dragonfly networks with trunking. The resulting routing algorithms approximate the performance of minimal, non-minimal and adaptive routings typically used in dragonflies, but without requiring virtual channels to avoid packet deadlock, thus allowing for lower-cost router implementations. This is obtained by selecting properly the link to route between groups, based on a graph coloring of the network routers. Evaluations presented in Section 5.7 show that the proposed mechanisms are competitive to traditional solutions when using the same number of virtual channels, and enable for simpler implementations with lower cost. Finally, multilevel dragonflies are discussed, considering how the proposed mechanisms could be adapted to them.

3.1 Introduction

Technology trends suggest that the use of high-radix routers [KDTG05] is the most cost-efficient alternative for the interconnection networks typically used in datacenters and High-Performance Computers (HPC). Frequent direct topologies proposed for HPC and datacenters are those based on meshes, tori, dragonflies [KDSA08] and Hamming graphs (also known as flattened butterflies [KDA07]). Among these, dragonflies and Hamming graphs are suitable for their use with high-radix routers, and they will be studied in detail in this chapter.

This chapter characterizes and compares Hamming and dragonfly topologies, studying their scalability, their respective degrees of freedom and providing a systematic characterization of each graph including balancing conditions that lead to a uniform use of network

resources under uniform traffic. The relationship between the Hamming graph and the dragonfly topology is studied, showing that the former can be seen as a dragonfly topology with an extremely high level of *global trunking*—several links among each pair of groups. Based on this relationship, the dimension-ordered deadlock-free routing (DOR) mechanism used in Hamming graphs, which does not rely on virtual channels (VCs), is adapted to dragonflies. Minimal and non-minimal routing mechanisms of this type are introduced for dragonflies with global trunking $t \geq 2$ and $t \geq 4$ respectively. These mechanisms rely on routing restrictions and therefore they decouple the number and use of virtual channels from deadlock avoidance. The posterior evaluation in Chapter 5 shows that the proposed mechanisms are competitive with state-of-the-art alternatives, without imposing minimal VC requirements on the router design.

On the other hand, high-radix is the norm for current HPC discrete routers, forthcoming designs such as Intel's Knights Landing and future Xeon chips will implement on-chip routers [Haz14]. In such designs, the router competes with on-chip cores, memories and I/O for the chip resources, including the pin bandwidth. This will necessarily lead to lower-radix routers. Scaling to large networks based on low-radix switches can be accomplished using multi-level dragonflies. Such designs will be studied in the last part of the chapter, compared to previously proposed routing mechanisms.

The rest of the chapter is organized as follows. Section 3.2 presents related work in the area. Sections 3.3 and 3.4 introduce and characterize the Hamming and dragonfly topologies. Section 3.5 focuses on dragonflies with trunking in the global level. Section 3.6 introduces two novel deadlock-free routing mechanisms for dragonflies with trunking, based on coloring the underlying graphs, which are evaluated in Section 5.7. To finish the contributions, Section 3.7 makes some remarks about the scalability and routing of multi-level dragonfly networks, discussing how to adapt the previous proposals for such cases. Finally, Section 3.8 concludes the chapter.

3.2 Related Work

The Hamming graph [Mul82] has been extensively studied. Other names for this graph, or for topologies based on it, are *rook's graph*, *K-cube* [LKF03], *generalized hypercube* [BA84], *flattened butterfly* [KDA07] and *HyperX topology* [ABD+09]. This graph has been also considered in [ASK13] as one of the base topologies for an intra-switch network[1]. The dragonfly network was first introduced by Kim *et al.* in [KDSA08]. Different routing mechanisms for dragonflies that better adapt to the traffic pattern or reduce the implementation cost have been proposed in other works [JKD09, GVB+12b, GVB+13c]. Industrial implementations have been the IBM PERCS [AAC+10] and Cray Cascade [FBR+12].

Network dimensioning typically seeks to balance the utilization of the network resources to maximize performance. Resource usage being balanced or not depends on the topology, traffic pattern and routing employed. Under uniform traffic and minimal routing, Square Hamming graphs and dragonflies with twice as many local ports as global ports per router are balanced [KDSA08], as will be detailed later. By contrast, an unbalanced design such as a rectangular Hamming graph would provide reduced performance caused by the bottleneck in the scarcest resources. However, even a balanced network can easily saturate under adverse traffic using minimal routing. This occurs when all the traffic concentrates

[1] [ASK13] also considers *local* and *global* topologies as it is done in this work, but they refer to the intra-switch topology and the traditional topology between switches, instead of per-group and intra-group.

on some few links, which leads to severe congestion. Valiant routing [Val82] selects a random intermediate router; traffic is first sent minimally to the intermediate router and then minimally to the final destination. This randomizes the network load, balancing the use of links, but doubles the utilization of the resources, halving its maximum throughput. Alternatively, task placement randomization [BGJK11] avoids hotspots by randomizing communications. Given the disparity of performance depending on the traffic pattern and routing, Hamming and dragonfly networks typically require adaptive routing mechanisms which rely on minimal routing for uniform traffic and revert to Valiant routing for adverse traffic patterns. Several of such adaptive routing mechanisms have been proposed in the literature [KDA07, KDSA08, JKD09, GVB$^+$12b, GVB$^+$13c].

Networks built on Hamming graphs are deadlock-free under DOR. Valiant routing can be made deadlock-free when DOR is employed for each half of the path using different VCs, requiring two of them. For dragonflies, most of the previous proposals adapt the distance-based mechanism by Günther [Gün81], employing as many VCs per router port as the longest path allowed in the network. When local and global links are always traversed in the same sequence, their VCs can be considered independently, leading to 2/1 VCs (local/global) required for minimal routing and 4/2 for Valiant routing [Val82, PRG$^+$14]. In [KDSA08] Kim *et al.* reduce this number to 3/2 by misrouting traffic to an intermediate group instead of an specific intermediate router, but in this way the traffic is not completely uniform and pathological performance problems can arise [GVB$^+$12b]. In OFAR [GVB$^+$12b] a simple deadlock-free escape network is embedded in the dragonfly and packets have the option to move to the escape network to avoid deadlock. Hence, in each port only 1 or 2 VCs are necessary (depending if it belongs to the escape subnetwork). However, this mechanism does not guarantee bounded paths per se, and requires a congestion management mechanism to avoid saturation in the escape subnetwork, [GVB$^+$13a]. Restricted Local Misrouting (RLM [GVB$^+$13c]) allows for local misrouting within any group of a canonical dragonfly without increasing the number of required VCs. This is implemented by forbidding certain combinations of two local hops which would generate cycles, in a similar way to how the routing mechanisms for dragonflies with trunking introduced in Section 3.6 select the global links that guarantee deadlock avoidance. Opportunistic Local Misrouting (OLM [GVB$^+$13c]) allows for cyclic dependencies to appear when applying local misrouting, but it guarantees that an alternative safe escape path always exists at any hop in the network.

The use of multiple virtual channels besides providing deadlock-freedom helps to reduce Head of Line Blocking (HoLB). However, they entail a significant cost. Not only they increase the area and power requirements for the router, but also make some router allocator stages more complex, leading to lower router frequencies and reduced throughput [PD01]. For this reason, multiple works propose alternatives to avoid or reduce the number of VCs in network routers, such as [WCP13, GVB$^+$12b]. HoLB is typically mitigated in these cases employing internal speedup, such as in [AAC$^+$10, FBR$^+$12].

3.3 Hamming Graphs

This section defines Hamming graphs, their properties, some alternative isomorphic definitions and the main routing mechanisms proposed for networks based on it.

The *Hamming distance* between two vectors is the number of components in which the vectors differ. Given a space S over which the Hamming distance is defined, the *Hamming graph* is defined as the graph with S as vertex set in which two vertices are connected if

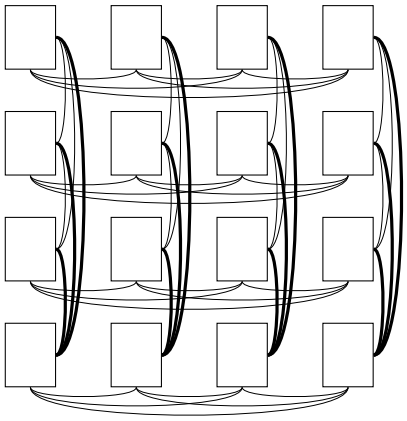


Figure 3.1: Hamming graph $K_4 \square K_4$ with vertices arranged in rows and columns.

and only if their Hamming distance is 1. For the Hamming distance the only relevant characteristics of the space are the number of components (dimensions) and the possible values of each component, this is, it can be assumed that the space is $\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_n}$ for some integers m_i. Figure 3.1 shows a representation of the Hamming graph over $\mathbb{Z}_4 \times \mathbb{Z}_4$.

The Hamming graph is isomorphic to the Cartesian product of complete graphs $K_{m_1} \square \cdots \square K_{m_n}$. As in the complete graph all vertices are connected, in the Hamming graph every vertex is connected with any other which differs in exactly one component. The Hamming graph is also isomorphic to the Cayley graph over the Abelian group $(\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_n}, +)$, with generator set $\bigcup_{i=1}^{n} \{x\mathbf{e}_i \mid \mathbf{x} \in \mathbb{Z}_{m_i} \setminus \{\mathbf{0}\}\}$. Hence it is a lattice graph, although with a generator matrix of order $\prod_{i=1}^{n} \lfloor \frac{m_i}{2} \rfloor$.

This chapter focuses in the bidimensional case, *i.e.* the Hamming graph over the space $\mathbb{Z}_a \times \mathbb{Z}_b$, for any pair of integers a, b. This Hamming graph is a diameter $k = 2$, Δ-regular graph, for $\Delta = a + b - 2$, comprising ab vertices. In the square case, this corresponds to $\frac{1}{4}\Delta^2 + \Delta + 1$ vertices. As said in Section 1.6, for Cayley graphs over Abelian groups of diameter 2 there is an upper bound of $\frac{1}{2}\Delta^2 + \Delta + 1$ vertices; the current best construction inside this family is the given in [MŠŠ12], which achieves $\frac{3}{8}(\Delta^2 - 4)$ vertices, about $\frac{3}{4}$ of the bound. Square Hamming graphs have about 1/2 of this bound, so while they are not the best, they have a good position among Cayley graphs over Abelian groups, while existing for any even degree. Each of these vertices represents one router in the network, to which Δ_0 compute nodes are attached (also known as *concentration level*). Thus, each router requires $R = \Delta + \Delta_0 = \Delta_0 + a + b - 2$ ports.

The Hamming topology admits up to $\Delta_0 = \min\{a, b\}$ compute nodes per router injecting at full rate under uniform traffic. Hence, larger concentration values are oversubscribed, leading to potential congestion; while lower values imply an underutilization of the network. This can be proved as follows. First, assume without loss of generality $a < b$ and consider the traffic from the region $\{(x, y) \mid 0 \le x < \frac{a}{2}, 0 \le y < b\}$ into $\{(x, y) \mid \frac{a}{2} \le x < a, 0 \le y < b\}$, with a even for simplicity. Each region contains $\frac{ab}{2}$ routers, each router attached to Δ_0 compute nodes. As the regions have the same size, the probability of having a destination in the other one is $\frac{1}{2}$. Thus $\frac{ab}{4}\Delta_0$ packets must traverse the links joining the regions each cycle. The number of these links is $b \cdot \frac{a}{2} \cdot \frac{a}{2}$; thus, to avoid saturation $\frac{ab}{4}\Delta_0 \le \frac{ba^2}{4}$ is required, which simplifies to $\Delta_0 \le a$. Then, in a balanced Hamming network with $a = b = \Delta_0$, there are a^3 compute nodes for routers of radix $R = 3a - 2$. Then, for a given radix R the

network connects up to $\left(\frac{R+2}{3}\right)^3$ compute nodes.

Like all Cayley graphs, the Hamming graph is vertex-transitive [AK89]. This can be seen with the automorphism $f(v) = v + v_2 - v_1$ for some vertices v_1, v_2, for which $f(v_1) = v_2$. The edges from (x, y) to (x', y) can be naturally denoted as a-edges and the edges from (x, y) to (x, y') as b-edges, corresponding to the two different dimensions in the Hamming graph. Under uniform traffic, a minimal network path will have one a-link with a probability $\frac{(a-1)b}{ab-1}$ and one b-link with probability $\frac{(b-1)a}{ab-1}$, which are both almost 1. Thus, in order to balance the use of the network links, the required condition is to have the same number of links per dimension ($a = b$), which corresponds to a square Hamming graph. Indeed, the Hamming graph is edge-transitive if and only if it is square. The sufficient condition is simple, if $a = b$ there exists an automorphism which maps each vertex (x, y) into (y, x). For the necessary condition, assume without loss of generality that $a < b$; then every a-link is included in some K_a subgraph but not in any K_b subgraph, thus a-links cannot be mapped into b-links. An unbalanced (not edge-transitive) implementation has less links in the shorter dimension, which becomes a performance bottleneck because of their higher utilization.

Networks based on the Hamming graph are deadlock-free under a DOR policy. This imposes restrictions on the paths that packets can follow, but not on the number of VCs employed by routers. Alternatively, distance-based deadlock avoidance mechanisms could be used without routing restrictions if the routers employ at least two VCs: one for the first hop and the other for the second. Finally, it is interesting to note that perfect error-correcting Hamming codes directly translate into solutions for the resource placement problem in Hamming networks (in a analogous way to [BB96]). However, this can be useful for Hamming networks of high dimension but it is not very relevant for the bidimensional case, since there are not perfect codes of length 2.

3.4 Dragonfly Topologies

This section presents the dragonfly topology analyzing its multiple degrees of freedom. Next, it discusses how some dragonfly topologies are subgraphs of a bidimensional Hamming graph. Finally, it introduces a formal definition of the canonical dragonfly topology with several alternatives for its global link arrangement.

The *dragonfly network* was proposed by Kim *et al.* [KDSA08] as a two-level hierarchical direct network. A dragonfly topology has b groups $(0, \ldots, b-1)$ each group being composed of a routers $(0, \ldots, a-1)$. Routers within a group (first level) are connected by short, cheap, electrical *local* links. Different groups (second level) are connected by long, expensive, optical *global* links. The definition of the dragonfly in [KDSA08] is, purposely, very loose, focusing on technological and economical aspects, rather than providing a closed definition of the underlying graph. Thus, from a formal point of view, multiple different topologies can be considered as variants of the dragonfly.

Apart from the parameters a and b, there are three degrees of freedom in the definition of a dragonfly topology:

i) *local topology*: the connectivity pattern of the routers within a group,

ii) *global topology*: the connectivity pattern between the different groups, and

iii) *global link arrangement*: the specific router on each group to which each global link connects.

The diameter k of the dragonfly topology depends on the diameters of the global topology k_g and local topology k_l as follows: $k \leq k_g + (k_g + 1)k_l = k_g + k_g k_l + k_l$. That is, a limit of k_g global links, $k_g + 1$ visited groups, with at most k_l local links in each of the visited groups. In order to minimize the diameter, the complete graph can be employed as both local and global topologies, leading to $k = 3$. Furthermore, the complete graphs reach the Moore bound and thus are very good candidates considering scalability. This choice of topologies has been the one of previous proposals [KDSA08, AAC+10, JKD09, GVB+12b, GVB+13c] and hence it will be called *canonical dragonfly* to the dragonfly network using complete graphs K_a and K_b in both local and global topologies. The canonical dragonfly, for $k = 3$, asymptotically reaches $4/27$ of the vertices of the Moore bound. The known families of graphs exceeding this value are theoretical works to reach the bound; mainly the family introduced in [Del85], which has severe practical inconveniences, such as restricting $\Delta - 1$ to odd powers of 2.

Alternative implementations to the canonical dragonfly also exist, such as in Cray Cascade [FBR+12], where a complete graph is used for the global topology and a rectangular 2D Hamming graph is used for the local one. Section 3.7 will discuss how this topology can be considered as a 3-level dragonfly. Topologies can employ parallel links between routers (*trunking*) what will be considered later in Section 3.5 and, unless otherwise noted, it is not employed in the dragonfly.

The degree of the topology Δ can be divided into the two levels. The degree associated to the first level is denoted by Δ_1, this is, the number of local links connected to each router. Analogously, Δ_2 represents the number of global links connected to each router. Thus, the topology has degree $\Delta = \Delta_1 + \Delta_2$ and the routers have a total number of ports or radix $R = \Delta_0 + \Delta_1 + \Delta_2$, treating compute nodes as the level 0. In any canonical dragonfly $b = a\Delta_2 + 1$ and $a = \Delta_1 + 1$. To achieve a balanced use of resources under uniform traffic, this is, to have similar load in local and global links, the condition $2\Delta_2 \approx \Delta_1$ needs to hold; the balancing condition proposed in [KDSA08] is $a = 2\Delta_2$, whereas up to $\Delta_0 = \Delta_2$ compute nodes can be connected to each router without saturating the network under uniform traffic. In a canonical dragonfly this imposes a relation between the group size a and the number of groups b. Given a group size a, the network is balanced only for the corresponding number of groups b; with less groups there would be too few global links which would become a bottleneck, and with more groups the local links would be the bottleneck. The second case should be typically forbidden by design by setting a maximum system size, but the first one is common in not fully populated systems which can be upgraded by installing more groups. In such case, balanced networks with a low number of groups b can be built using trunking; the corresponding balancing conditions will be discussed in detail in Section 3.5.1. In a balanced canonical dragonfly without trunking, the routers have radix $R \approx 2a$, there are about $a^3/2$ routers and about $a^4/4$ compute nodes. Then, for a given radix R the network comprises up to $\approx R^4/2^6$ compute nodes.

Proposed routing mechanisms in the canonical dragonfly are hierarchical, routing first to the destination group and then to the destination node. The *minimal routing* introduced in [KDSA08] first locates the global link between the source and destination groups; the path consists of one local link l to the router with the required global link, then the global link g itself and finally a local link l to the destination; this is denoted as a *lgl* route. Using such hierarchical routing (instead of a flat routing) avoids paths with only two global links *gg*, which can be shorter in terms of hops but typically have higher latency because of the longer physical length of global links. Most deadlock-free routing mechanisms for dragonflies rely on an ordered use of virtual channels. Each hop of a path employs a

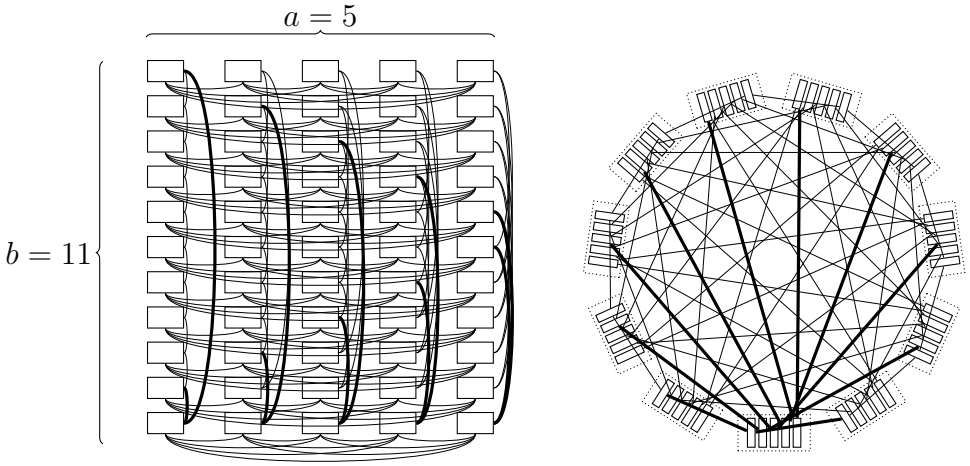


Figure 3.2: Two layouts of the same dragonfly topology which is a subgraph of $K_5 \square K_{11}$, with $\Delta_2 = 2$, with nodes organized in rows and columns (left, each row corresponds to a different group) or groups (right). Global links leaving group 0 are in bold.

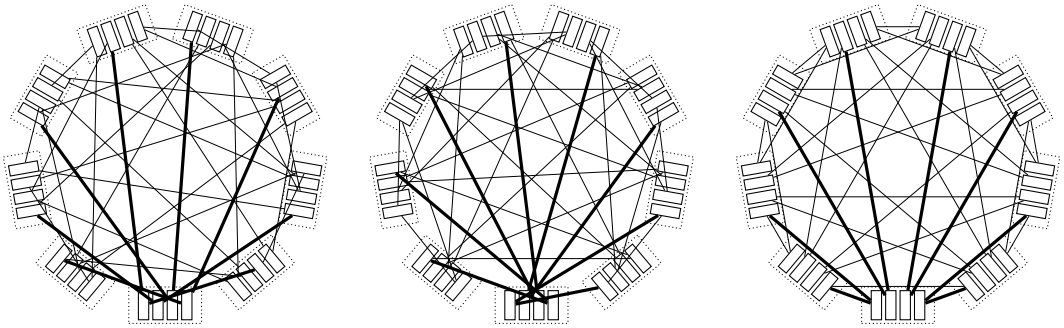
different VC in an increasing order, thus avoiding cyclic dependencies. Since minimal paths are always of type *lgl*, or a subset of them in the same order (but never *gll* or *llg*), using minimal routing, local ports employ two different VCs and global ports do not need to employ VCs.

In some cases, a canonical dragonfly topology is a subgraph of the rectangular Hamming graph $K_a \square K_b$. Figure 3.2 presents an example with $a = 5$, $b = 11$ and $\Delta_2 = 2$. The local topology of each dragonfly group corresponds to each of the complete graphs K_a, whereas the global topology links need to connect vertices in the same position of each group in order to belong to the original Hamming graph. Thus, a independent graphs, $G_0, G_1, \ldots, G_{a-1}$, define the global link connectivity. In order to build a canonical dragonfly, each of these G_i graphs needs to have b vertices $V(G_i) = \{0, 1, \ldots, b-1\}$ and degree Δ_2, which exists if and only if $b \geq \Delta_2 + 1$ and $b\Delta_2$ is even. The global topology composed as the union of all G_i's needs to be a complete graph K_b so the result is a canonical dragonfly; the *union* of graphs over the same set of vertices is the graph containing all the edges of the factors, this is, $E(\bigcup_i G_i) = \{e \mid e \in E(G_i) \text{ for some } i\}$. Thus the problem is to decompose K_b into a graphs, $G_0, \ldots, G_{a-1}$, of degree Δ_2. Systematic decompositions can be found easily for b odd and Δ_2 even. For $\Delta_2 = 2$, as in Figure 3.2, K_b can be decomposed into $\frac{b-1}{2}$ cycles for b odd, [Hil84]. For $\Delta_2 > 2$ even, several of such cycles can be merged into each of the G_i. Although only for certain parameters, as it will be further discussed in Subsection 3.4.1, these subgraphs of Hamming graphs are the only vertex-transitive canonical dragonfly arrangements which we have encountered.

3.4.1 Global Link Arrangement and Network Symmetries

Given a canonical dragonfly, there exist $b^{2^{O(a\Delta_2)}}$ possible arrangements for the global links. This subsection discusses link arrangements in general and a few specific cases: *consecutive*, *palmtree*, and *circulant-based* in which the topology is a subgraph of the Hamming graph, as introduced above. Finally, it presents a brief discussion on the selection of an arrangement. Arrangements with trunking will be presented in Section 3.5.2.

In general, any arrangement can be implemented as follows:



(a) Random (b) Consecutive (c) Palmtree

Figure 3.3: Three arrangements for $a = 4, b = 9, \Delta_2 = 2$ with nodes organized in groups.

i) For each group g, partition the set of groups other than g, into a subsets (sets of groups) of cardinality Δ_2. Then assign one subset to each router of g.

ii) For every pair of groups A, B, find in A the router assigned to group B and in B the router assigned to group A. Then, add a global link between the routers found.

A *random* arrangement makes the choices in the first step at random. An example is presented in Figure 3.3a. Any network configuration admits being implemented in this way, although sometimes there is a simpler ad hoc implementation.

Consecutive Arrangement

The *consecutive* allocation of global links consists on connecting the routers in each group in consecutive order, with the groups in the network also in consecutive order, starting always from group 0 and skipping those links with source and destination being in the same group. Specifically, the vertex i in group j is connected for every integer $k = 0, \ldots, \Delta_2 - 1$ with the vertex $\lfloor \frac{j-1}{\Delta_2} \rfloor$ of the group $g = i\Delta_2 + k$ if $g < j$ and with the vertex $\lfloor \frac{j}{\Delta_2} \rfloor$ of the group $g + 1$ otherwise. Although not explicitly indicated, this consecutive arrangement can be inferred from the figures in [KDSA08]. Figure 3.3b shows an example for $a = 4$.

Palmtree Arrangement

The *palmtree*[2] arrangement presents the same global connectivity pattern in each group of the system. In this arrangement, vertex i in group j is connected to vertices $a - 1 - i$ in groups $j - i\Delta_2 - 1, j - i\Delta_2 - 2, \ldots, j - i\Delta_2 - \Delta_2 \pmod b$. Although not explicitly indicated, the palmtree arrangement can be inferred from the figures in [GVB+12b]. A palmtree for $a = 4$ is included in Figure 3.3c.

The palmtree arrangement presents notable symmetries. The clearest one is the rotational symmetry given by the automorphism defined by sending the vertex x in group y, (x, y), to $(x, y + 1 \pmod b)$. This rotation shows that all groups are equivalent modulo automorphism. Another symmetry is given by $f(x, y) = (a - 1 - x, -y \pmod b)$, which is a reflection in each group. Therefore, there are at most $a/2$ classes of vertices modulo the equivalence relation induced by automorphisms. Interestingly, for any pair of vertices of

[2]The name is inspired by the similarity of the links leaving each group with the Palm Islands in Dubai, which are shaped as a palmtree.

the same class of these $a/2$ classes, there is a path between them using only global links. Reciprocally, each global link connects vertices of the same class.

Circulant-based Arrangement

This arrangement is a particular case in which the dragonfly network is a subgraph of the Hamming graph, as introduced above in this section. Here, the set of global links is the union of circulant graphs and Δ_2 is restricted to be even for simplicity. In this arrangement, vertex i in group j is connected to vertices i in groups $j \pm (i\Delta_2/2+1), j \pm (i\Delta_2/2+2), \ldots, j \pm (i\Delta_2/2 + \Delta_2/2) \pmod b$. In the example of Figure 3.2 with $a = 5, b = 11, \Delta_2 = 2$, each graph G_i (corresponding to column i) contains the edges from x to $x \pm i \pmod b$ and thus it is a circulant graph.

Interestingly, this arrangement has the property that for $\Delta_2 = 2$ if b is a prime number, the resultant topology is vertex-transitive. To see that, consider the following automorphisms: an automorphism f that maps the vertex (i, j) into $(i, j + 1 \pmod b)$ and an automorphism g which maps the vertex (i, j) into $(\min\{2i + 1, b - 3 - 2i\}, 2j \pmod b)$. The automorphism f cycles the groups, and hence, there are at most a classes of vertices. Then, if b is odd, g is an automorphism, and if b is a prime number then g acts transitively into the vertices of the group 0. Thus, for b prime there is only one class modulo isomorphism and the graph is vertex-transitive or node-symmetric.

Discussion on the Global Link Arrangement Selection

While the global link arrangement is important to fully characterize a topology, the simulations in Section 5.6 will show that the impact of the selected arrangement on network performance under uniform traffic is, in general, negligible[3]. However, specific arrangements have different topological properties, such as symmetries and the possibility of defining multiple classes of vertices in the network, what can be exploited to simplify routing. The *palmtree* and any subgraph of the Hamming graph allow for a natural vertex coloring with $\frac{a}{2}$ (for a even) and a colors respectively, in a way such that every global link has the same color in its two endpoints. This property will be used by the routing mechanisms in Section 3.6.

Additionally, as studied in [GVB+12b], for certain traffic patterns, pathological saturation of local links occurs when using the Valiant variant from [KDSA08], which does not employ local misrouting in the intermediate group. This occurs when all the nodes in group i send traffic to nodes in group $i + \Delta_2 \pmod b$. The saturation arises in the intermediate group, in which almost all of the traffic received from the Δ_2 global links from a router leaves the group using the same neighbour router. The single link between these routers becomes a performance bottleneck. With the *consecutive* or *palmtree* arrangements, all traffic received by router i needs to leave by router $i + 1$, leading to a throughput limit of $1/\Delta_2$ *phits/node/cycle* (a *phit* is the amount of data transferred on a link on a single cycle). In the *circulant-based* arrangement only $\Delta_2/2$ of such links would compete for the same local link, leading to a throughput limit of $2/\Delta_2$ *phits/node/cycle*. A *random* arrangement would typically eliminate this problem, at the cost of regularity in the network. In any case, such pathological performance issues can be solved using the original implementation of Valiant routing [Val82] (as discussed in [PRG+14] and implemented in [FBR+12]) or

[3]Different traffic patterns such as global permutations could be impacted by the global link arrangement.

allowing for local misrouting in the intermediate group (as in the OLM routing [GVB+13c], which is employed as a reference in Section 5.7).

3.5 Dragonfly topologies with Global Trunking

This section considers trunking in dragonfly topologies and discusses how the Hamming graph responds to the definition of a dragonfly topology with trunking. Based on this observation, Hamming graphs and canonical dragonfly topologies are considered as the two extreme possibilities of trunking and the spectrum between them is studied considering the corresponding balancing conditions.

The *trunking level* in a topology refers to the number of parallel links that are employed to increase the aggregated bandwidth, increasing also the number of router ports used. In a dragonfly topology, *local trunking* refers to parallel links between pairs of routers within a group. Such parallel links between pairs of routers are typically known as a LAG (Link Aggregation Group). This LAG could be also implemented in a Hamming topology to increase bandwidth between routers in the same row or column. The *global trunking* level t is the number of global links between every pair of groups. In this case, there are multiple alternative implementations. Trunk links can join a single pair of routers (LAG), one router in a group and multiple routers in the other (often called as Multi-Chassis LAG, MC-LAG), or different pairs of routers.

Unless otherwise noted, trunking will always refer to global trunking between different pairs of routers, that increases both bandwidth and reliability. As discussed in [FBR+12], trunking is required to retain optimal global bandwidth in systems with less groups than the maximum allowed. A dragonfly with trunking is specified by the number of routers per group a, the groups b, the global links per router Δ_2, the global link arrangement and the global trunking $t > 1$ ($t = 1$ for a canonical dragonfly without trunking as defined in Section 3.4). Dragonflies with global trunking obey the relation:

$$a\Delta_2 = t(b-1). \tag{3.1}$$

The vertices of a Hamming graph $K_a \square K_b$ can be partitioned into b groups by defining the group y as the set $\{(x,y) \mid x \in \mathbb{Z}_a\}$. Clearly these groups are subgraphs isomorphic to K_a and hence the Hamming graph satisfies the definition of the canonical dragonfly topology (complete graphs for local and global topologies) with trunking $t = a$. Between groups y_1 and y_2 there is the set of a global links $\big\{\{(x,y_1),(x,y_2)\} \mid x \in \mathbb{Z}_a\big\}$.

Therefore from a topological point of view, the Hamming graph $K_{a'} \square K_{b'}$ is a trunked dragonfly with parameters $a = a'$, $b = b'$, $t = a'$ and $\Delta_2 = b - 1$; using a specific global link arrangement which connects all routers in the same position of each group. An example of the Hamming graph represented as a trunked dragonfly can be seen in Figure 3.4.

3.5.1 Balancing Conditions for the Trunked Dragonfly

The requirements for a balanced trunked dragonfly are studied in detail in this subsection, considering uniform traffic and minimal routing. As discussed before, non-uniform traffic can be made uniform by randomizing it (like Valiant routing) or by other means such as randomizing task placement, so it is not considered in this analysis. The detailed balancing conditions are derived from calculating the average distance on each type of links and relating it to the number of links of each type to be used, considering network trunking.

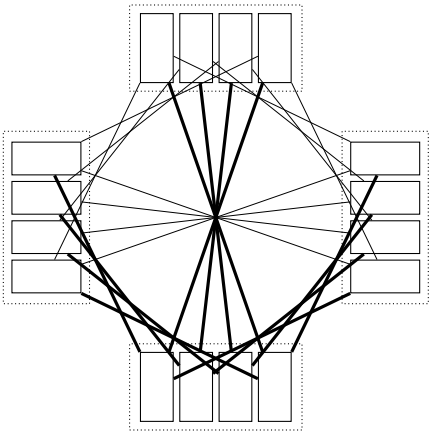


Figure 3.4: Hamming graph $K_4 \square K_4$ with nodes organized in groups.

Let $\bar{k}$ be the average distance, this is, the quotient between the length and the number of all possible minimal paths. This distance can be divided into $\bar{k} = \bar{k}_1 + \bar{k}_2$, with $\bar{k}_1$ being the average number of hops in local links and $\bar{k}_2$ in global links. A similar relation can be established with the total number of edges $|E| = |E_1| + |E_2|$. A balanced network requires

$$\frac{\bar{k}_1}{|E_1|} = \frac{\bar{k}_2}{|E_2|}.$$

Let $\alpha = \frac{\bar{k}_2}{\bar{k}_1}$ represent the relation between the use of each type of links under uniform traffic. Thus, α also represents the relation between the amount of links of each type (global, local) for a balanced network, $\Delta_2 = \alpha \cdot \Delta_1$. In order to approximate α, it can be observed that for global links $\bar{k}_2 = \frac{ab-a}{ab-1} \approx 1$. The average distance in local links $\bar{k}_1$ can be derived from the number of possible minimal paths between two groups, ignoring the communication internal to a single group, as follows. There are a^2 pairs of source/destination vertices among two given groups; t vertices of each group have a direct global edge to the other group and $a - t$ do not. Hence,

- t pairs are connected by a direct global edge, which is their minimal path, *g*.
- $t(a - t)$ pairs begin at a vertex with a global edge to the other group and finish in one without such edge. Their minimal path is *gl*.
- $(a - t)t$ pairs begin at a vertex without a global edge to the other group but finish in one with such edge. Their minimal path is *lg*.
- $t(t - 1)$ pairs begin and finish at vertices with global edges between the groups, but they are different. Their two possible minimal paths are *lg* and *gl*.
- $(a - t)(a - t)$ pairs begin and finish at vertices without direct global edges. The t minimal paths are *lgl*.

Thus, ignoring the traffic local to a group, $\bar{k}_1 \approx (t^2 - t - 2ta + 2a^2)a^{-2}$. Removing low order terms it becomes $\bar{k}_1 \approx 1 + (\frac{t}{a} - 1)^2$ and α can be approximated as

$$\alpha \approx \frac{1}{1 + (\frac{t}{a} - 1)^2}. \tag{3.2}$$

t	Limit for b using $\alpha = 1$ in (3.3)	b for a balanced network, according to (3.4)	Limit for b using $\alpha = 1/2$ in (3.3)	Actual network example
1	13	8.7	7.5	can. dragonfly $b = 9$
2	7	5.8	4.0	$b = 5$, $\Delta_2 = 2$
3	5	4.8	3.0	$b = 5$, $\Delta_2 = 3$
4	4	4.0	2.5	Hamming $K_4 \square K_4$.

Table 3.1: Examples of dimensioning the number of groups b of a network with $a = 4$ routers per group, for different levels of trunking t as in Figure 3.5. Networks with less groups (middle column) require more trunking to be balanced.

Thus, in the Hamming graph $t = a$ and $\alpha = 1$, whereas in the canonical dragonfly $t = 1$ and α tends to $\alpha \to 1/2$ for $a \to \infty$. The approximate dragonfly balancing conditions presented in Section 3.4 ($2\Delta_2 \approx \Delta_1$ or $a = 2\Delta_2$) no longer hold when the network employs trunking, since α becomes larger than $1/2$ so the ratio between global and local router ports needs to increase.

The parameter α and its relation with the number of edges of each type in a balanced network is:

$$\alpha = \frac{\bar{k}_2}{\bar{k}_1} = \frac{|E_2|}{|E_1|} = \frac{\frac{tb(b-1)}{2}}{b\frac{a(a-1)}{2}} = \frac{t(b-1)}{a(a-1)}. \tag{3.3}$$

From the expressions of α in 3.3 and 3.2, the following balancing condition is obtained:

$$b = 1 + \alpha\frac{a(a-1)}{t} \approx 1 + \frac{1}{1 + (\frac{t}{a} - 1)^2}\frac{a(a-1)}{t}. \tag{3.4}$$

The balancing condition (3.4) can be related with the cardinal equation (3.1):

$$a\Delta_2 = t(b-1) = a(a-1)\alpha. \tag{3.5}$$

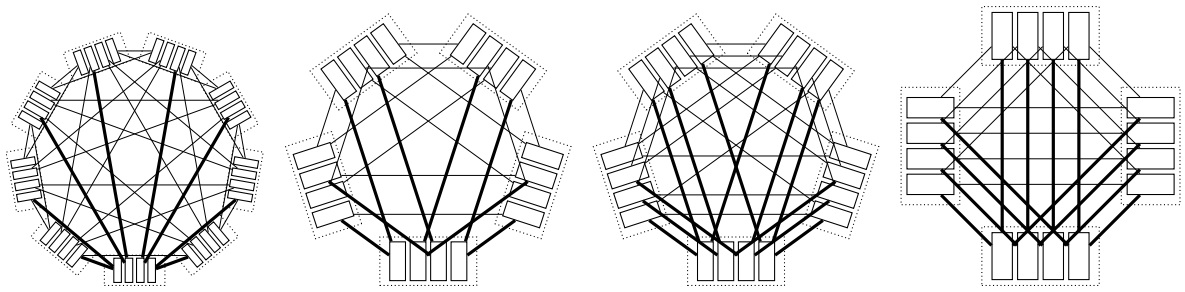
In the extreme case of Hamming graphs, $t = a$ and $\alpha = 1$, and hence the balancing condition is $b = a$ or equivalently $\Delta_2 = a - 1$. This was already known since it is the case of the Hamming graph being edge-transitive.

Table 3.1 shows in the middle column several dimensioning examples for groups of $a = 4$ routers, and in the sides the valid range for the number of groups b that keep $\alpha \in [\frac{1}{2}, 1]$. Since the result from the balancing equation (3.4) is not necessarily integer, an approximation with integral values is presented on the right. The corresponding topologies can be seen in Figure 3.5. It can be observed that for $t = 1$ (no trunking) the balancing condition is close to the lower limit given for $\alpha = 1/2$ on its right, whereas for $t = a = 4$ (maximum trunking) the result is close to the upper limit for $\alpha = 1$ on its left. Also, it is clear that the less groups of a dragonfly are present, the higher the trunking level is required for the topology to be balanced.

3.5.2 Arrangements for Dragonflies with Global Trunking

Trunking increases the number of possible arrangements of a dragonfly network. Subsection 3.4.1 introduced several possible global link arrangements for dragonflies without trunking. Those same configurations can be directly applied when using LAG, this is, multiple parallel links between each pair of linked routers. This section extends the arrangements presented in Subsection 3.4.1 to use trunking with disjoint pairs of routers for parallel links, to maximize fault tolerance. We denote such configurations as "extended."

In general, building a trunked dragonfly with an arbitrary arrangement requires:



(a) $t = 1$ (b) $t = 2$ (c) $t = 3$ (d) $t = 4$

Figure 3.5: Dragonfly networks with extended palmtree arrangement; a=4 routers per group and b groups, according to Table 3.1.

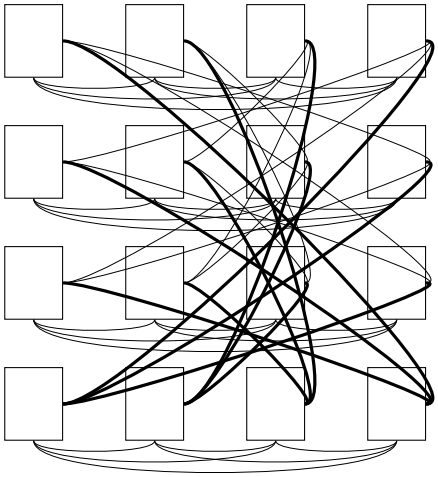


Figure 3.6: Palmtree arrangement for $t = a = 4$; vertices organized in rows and columns.

i) In each group, for each router select Δ_2 (generally different) groups. Among all the routers of the group, each other group must have been selected exactly t times.

ii) For every pair of groups A, B, find in A the t routers which have selected B and in B the t routers which have selected A. Therefore, there are $t!$ ways to add the t global links between the two collections of routers found.

The consecutive arrangement presented in Subsection 3.4.1 is generated adding global edges in a greedy way. However, for $t > 1$ any greedy strategy ends connecting some router to several other routers of the same remote group (which is denoted as multichassis-LAG). Since this section searches for solutions that connect disjoint pairs of routers for maximum fault tolerance, no extension of the consecutive arrangement is presented.

Extended Palmtree Arrangement

A generalization of the palmtree arrangement is defined here for any trunking level which obeys equation (3.1). This configuration employs disjoint pairs of routers for each parallel link between groups. The router x of group y is connected by global links to the following

Δ_2 routers:

$$\{\big(a-x-1, \text{rem}(y+1+\text{rem}((a-x-1)\Delta_2+k-1, b-1), b)\big) \mid k \in \{1,\ldots,\Delta_2\}\}.$$

As in the base case, $(x,y) \mapsto (x, \text{rem}(y+1,b))$ and $(x,y) \mapsto (a-x-1, \text{rem}(b-y,b))$ are automorphisms of the extended palmtree. Hence, there are at most $\lfloor \frac{a}{2} \rfloor$ isomorphism classes. The graphs in Figure 3.5 employ such arrangement and, as stated before, they correspond to the examples in Table 3.1. The graph obtained for $a = b$ is very similar to the Hamming graph but not isomorphic to it; this is the case of the last graph in Figure 3.5. A representation of such graph with nodes organized in rows and columns is presented in Figure 3.6, providing a visual comparison with the Hamming graph presented in Figure 3.1. It is remarkable that a *lg* path exists for any pair of routers with this last arrangement, enabling a deadlock-free DOR routing.

Extended Circulant-based Arrangement

Subsection 3.4.1 discussed the construction of canonical dragonflies as subgraphs of the Hamming graph, by finding a decomposition of the complete graph K_b into a regular subgraphs. When using trunking, the construction relies on finding a decomposition of $t > 1$ copies of a complete graph, this is, of a multigraph with t edges between every pair of vertices.

The arrangements composed of multiple circulant graphs from Subsection 3.4.1 can be easily extended to the case of global trunking, under the restrictions of equation (3.1), even Δ_2 and odd b. Specifically, the following connectivity pattern generates a dragonfly with trunking t: vertex i in group j is connected to vertices i in groups $j \pm (\text{rem}(\frac{\Delta_2}{2}i+k, \frac{b-1}{2})+1) \pmod b$ for every integer $k \in \{0,\ldots,\frac{\Delta_2}{2}-1\}$.

3.6 Deadlock-free Adaptive Routing in Dragonflies with Trunking

As discussed in Section 3.4, distance-based deadlock-free routing mechanisms proposed for dragonflies require as many VCs as hops allowed through a given type of network link. Such implementations can be costly and complex, and tie the number of VCs with the maximum path length. However, Hamming graphs allow for deadlock avoidance mechanisms based on route restrictions (DOR). Such a mechanism does not require VCs.

Section 3.5 showed how Hamming graphs and dragonflies with trunking can be seen as members of the same family. In this section, three alternative routing mechanisms are introduced for dragonflies with global trunking, based on a variation of the route restriction mechanism employed in Hamming graphs. A DOR mechanism is equivalent to coloring all the links in the network with one of two colors, according to their dimension, and following paths that obey a certain color order. Similarly, the proposed mechanisms impose a selection of the global links in the path, from those t specified by the trunking level. They rely on coloring the routers, which is possible when the global link configuration is an extended palmtree or a subgraph of the Hamming graph (as defined in Subsections 3.5.2 and 3.5.2), what highlights the importance of a careful selection of the global link connectivity.

DOR can be safely used with trunking level $t = a$. With $a > t \geq 2$, cyclic dependencies in minimal routing can be avoided by deciding which of the t global links to use each time, based on two router colors and without relying on VCs; as it will be seen shortly, it requires

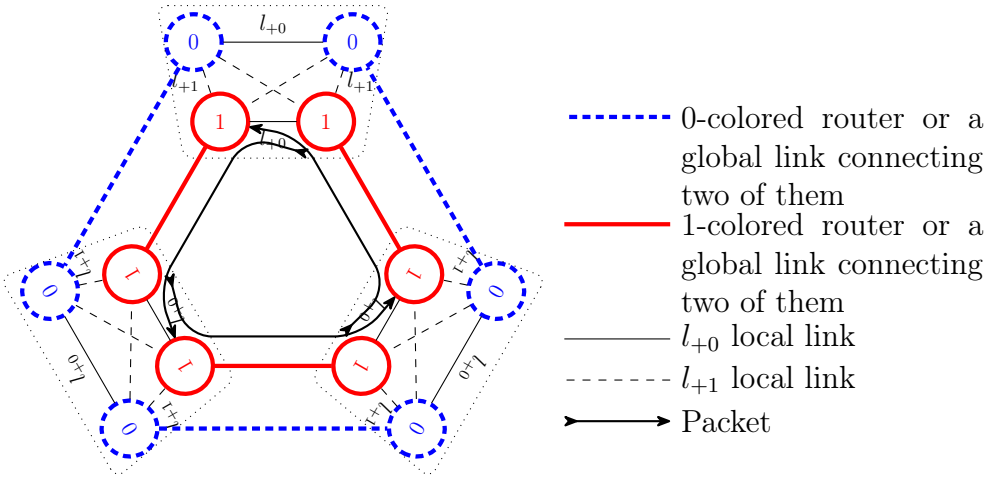


Figure 3.7: Coloring of routers with 0 or 1 and the local links with +0 or +1. The cyclic dependency presented would be avoided using the color-ordering rules, since at least one of the messages must follow the l_{+1} local channels.

$t \geq 2$. For adverse traffic patterns, a variant of Valiant routing (which sends traffic to an intermediate network router) can be implemented without VCs, requiring $t \geq 4$. These two mechanisms are oblivious. Finally, an adaptive mechanism can be implemented, which selects between the minimal or Valiant paths depending on network conditions, requiring again $t \geq 4$. These three mechanisms are detailed next and evaluated in Section 5.7.

3.6.1 Oblivious Minimal Deadlock-free Routing for $t \geq 2$

In this subsection a deadlock-free routing mechanism denoted *2-color minimal* is introduced for dragonflies with $t \geq 2$ global links between pairs of groups. Deciding which of the t links to use for each packet can prevent deadlock; $t = 2$ will be assumed from here onwards although the mechanism is still valid for larger values of t. However, in such cases the proposals of the next subsection present several advantages.

Every router in the network will be colored with one of two colors, say 0 and 1. Considering an even number of routes per group a, half of them receive each color. Global links should be arranged so they only connect vertices with the same color, which implies a restriction in the global link arrangement. The *extended palmtree* for $a \geq 4$ and any subgraph of the Hamming graph for $a \geq 2$ satisfy this restriction since they divide vertices into several classes. Local links are labelled according to the difference of the color of their endpoints, modulo 2. Thus there are "+0" and "+1" local links, depending on whether they connect vertices with the same or different color, respectively. They will be denoted l_{+0} and l_{+1}. A simple three-group example is presented in Figure 3.7.

The routing mechanism will vary depending on the respective colors of the source and destination routers. Routes with source and destination of different colors will need to employ up to one l_{+0} and one l_{+1} local links. The l_{+0} link always will be selected in the source group and the l_{+1} in the destination group. Implicitly, this forces the selection of the global link to be used, which will have the same color as the source router. This routing restriction prevents dependencies from l_{+1} to l_{+0} local links, which furthermore implies that any possible cyclic dependencies are completely composed either of l_{+0} local links or of l_{+1} local links. For routes in which endpoints have the same color, the path

must contain two l_{+0} or two l_{+1} local links. Selecting which one is employed is done in a careful way to avoid deadlock. Our mechanism employs the l_{+0} local links when the destination group index is larger than the source index and the l_{+1} local links otherwise. Again, this implies a selection of the global link to traverse. Alternative orderings between the groups can be used, as long as they guarantee that directed cycles do not appear in the global topology.

The proposed mechanism is deadlock-free by construction: multiple paths between routers with different color never form cycles because they follow local links in an ascending order, and paths between routers of the same color never form cycles because they employ different links when the group index is increased or decreased. An example is presented in Figure 3.7, in which three paths between routers of different groups always employ l_{+0} local links. Such cycle is forbidden with the proposed mechanism, since at least one of the paths will decrease the group index and thus will be forced to employ the l_{+1} links. Finally, it should be noted that under uniform traffic all links are used similarly. There are $(\frac{a}{2})^2$ "+1" and $2\binom{a/2}{2} = \frac{a}{2}(\frac{a}{2}-1)$ "+0" local links per group. Their ratio tends to 1 for large a. Global link usage is completely balanced, according to the color of the source and destination routers.

3.6.2 Oblivious Minimal and Non-minimal Deadlock-free Routing for $t \geq 4$

Non-minimal Valiant routes like *lgllgl* are required to balance load and avoid bottlenecks under adverse traffic patterns. This section introduces a non-minimal routing for dragonflies with $t \geq 4$ global links between pairs of groups denoted as "4-color nonminimal," which does not need VCs for deadlock-freedom. Additionally, by traversing only the first or the second half of the allowed path, routes with a single global hop can be employed. Such routing will be denoted as "4-color minimal," despite using in some cases one extra local hop. This minimal routing is less restrictive than the previous mechanism for $t = 2$, what will be patent in the performance results of Section 5.7.

Like in the previous subsection, this new mechanism relies on a coloring of the routers that allows to classify and order the local and global links considering a directed graph. Unlike the previous mechanism in which the order was only relevant for local links, in this case the link order will be strict. Considering the possible paths, this requires 4 colors for routers, 2 colors for global links and 8 colors for local links; 6 colors for local links are not enough to generate a balanced use of the network links, as it will be seen later. The four router colors will be labelled with one number and one letter $\{0A, 0B, 1A, 1B\}$. Every global link joins routers of the same color, what is possible for the extended palmtree with $a \geq 8$ and for subgraphs of the Hamming graph with $a \geq 4$. Global links will receive one of two labels, A or B, the same as of their endpoints. By contrast, local links receive one of eight labels. A local link from a router labelled xP to a router labelled yQ will be labelled as $+zPQ$ where $z \equiv y - x \pmod 2$, $P, Q \in \{A, B\}$ and $x, y, z \in \{0, 1\}$.

In order to provide a deadlock-free routing, an ordering of the links is required. That is, if α and β are two classes of links with $\alpha \prec \beta$ (α preceding β), then in every possible route there will be at most one link of each class and then the link of class α will appear earlier in the route than the link of class β. The partial order of global links will always be $g_A \prec g_B$, as it is required in a complete graph K_b. Considering local links, the routing employs the complete ordering $l_{+0AA}, l_{+0BA} \prec g_A \prec l_{+1AA} \prec l_{+1AB} \prec l_{+0AB} \prec l_{+1BB} \prec g_B \prec l_{+1BA}, l_{+0BB}$, which allows for the paths shown in Figure 3.8 in which every node

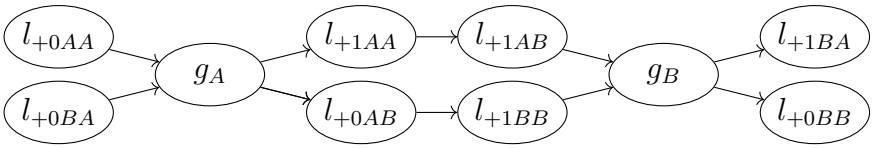


Figure 3.8: A precedence of links using $t = 4$ which allows for routes *lgl* and *lgllgl*. Allowed paths flow from left to right, and parallel routes represent different alternatives, one of which is chosen depending on the labels of the source and destination routers.

represents a link in the path.

Any pair of nodes can communicate using this ordering. The first local link is selected between l_{+0AA} or l_{+0BA} depending on the label A or B of the source router. Similarly, the last local link allows to select the A/B label of the destination router and the middle branch allows to select the change $+0/+1$ of the whole path. For example, a route from $0A$ to $0A$ must be $l_{+0AA}, g_A, l_{+0AB}, l_{+1BB}, g_B, l_{+1BA}$. This ordering restricts the class of the middle router, $0B$ in the previous example, which illustrates the restriction of routes applied. However, for any pair of $0A$ source and destination routers in different groups, this mechanism allows to select any of the $0B$ routers in the network as the intermediate router of the Valiant path. Similar routes can be calculated for the other 15 color combinations of source-destination pairs. Thus, it is similar to Valiant [Val82] but with the intermediate node restricted to a fourth of the total nodes.

The same link ordering can be used for minimal routing. Depending on the labeling of the source and destination, either the first 3 links or the last 3 links will be used. For example, packets going minimally from a $0A$ router to a $1A$ router will need to use a route l_{+0AA}, g_A, l_{+1AA} (first half); and to go minimally from a $0A$ to a $0A$ the path is l_{+1AB}, g_B, l_{+1BA} (second half). A priori, one could expect a small loss of performance when using this routing, since some packets which could minimally route as *gl* increase their paths in routes *lgl* to satisfy the coloring criteria. However, as it will be observed in Section 5.7, the proposed deadlock-free routing algorithms perform similarly to the references and in some cases outperform them. It is also interesting to remark that the minimum routing mechanism for $t \geq 4$ performs better than the one for $t \geq 2$.

As both minimal and non-minimal routes are allowed with the same ordering of links, adaptive routing can be employed by selecting one of them at the source. This requires the use of some decision mechanism, such as UGAL [Sin05] and using congestion information from neighbors as in [JKD09].

3.7 3-level Dragonflies

While dragonfly networks provide a very competitive scalability, larger networks can be built if the number of hierarchy levels is increased at the cost of a longer diameter. Alternatively, very large systems can be built based on moderate-radix routers (such as the integrated routers discussed in the introduction) if multiple levels are employed. This section explores the properties of 3-level dragonflies. More levels could be considered but the analysis would be similar and as it will be seen, the scalability grows very quickly with the number of levels, making configurations with more than 3 levels unlikely.

For 3-level dragonflies, links can be considered as local (l or 1), medium (m or 2) and global (g or 3). For notation, a 1-level group contains a routers, a 2-level group contains

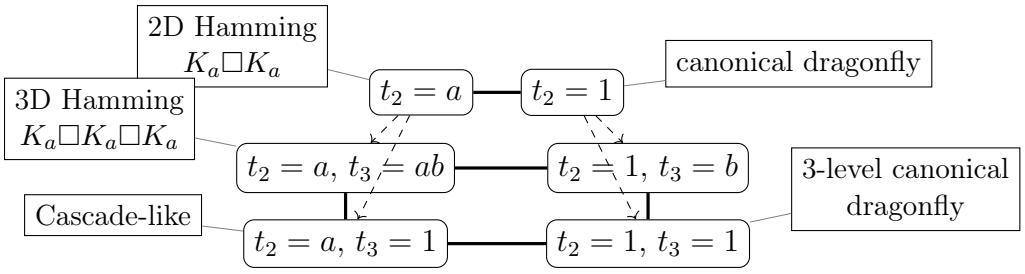


Figure 3.9: Classification of 3-level networks. Nodes correspond to extreme cases. Solid lines correspond to changes in one of the trunking levels. Dotted arrows represent the increase from two to three dimensions, where a trunking level for the new dimension must be chosen.

b 1-level groups and there are c 2-level groups in the whole network. The degree will be extended to $\Delta = \Delta_1 + \Delta_2 + \Delta_3$. Two trunking levels can be considered in this case: t_2 will be the number of 2-links between every pair of 1-groups and t_3 will be the number of 3-links between every pair of 2-level groups.

The network average distance can be decomposed as $\bar{k} = \bar{k}_1 + \bar{k}_2 + \bar{k}_3$. Similar to the 2D trunked dragonfly studied in Section 3.5.1, balancing conditions can be derived from a calculation of the relations between the average distance on each type of link. It can be defined as $\alpha = \frac{\bar{k}_2}{\bar{k}_1}$ and $\beta = \frac{\bar{k}_3}{\bar{k}_2}$. The equations of size and balance (3.5) generalize easily:

$$a\Delta_2 = t_2(b-1) \approx a(a-1)\alpha \text{ and } ab\Delta_3 = t_3(c-1) \approx t_2 b(b-1)\beta.$$

From which the following expressions of the degrees are obtained:

$$\Delta_2 \approx (a-1)\alpha \approx \Delta_1\alpha \text{ and } \Delta_3 \approx (a-1)\alpha\beta \approx \Delta_2\beta.$$

In 3-level dragonflies a medium-link arrangement and a global-link arrangement can be defined. Any combination could be chosen such as (random, palmtree) or (random, random), where the first element of the tuple indicates the medium-link arrangement and the second element the global-link arrangement The definition for the global-link arrangement equals the one in the 2-level case only when $t_2 = 1$, otherwise it needs some adaption.

In this 3-level case, minimal routes are in general *lmlglml*, thus requiring up to 4 VCs. The classical Valiant [Val82] (using an intermediate router) duplicates the route and would need up to 8 VCs. Shortened versions as in [KDSA08] can be defined; using an intermediate 1-level group the routes would be *lmlglmlmlglml* requiring only 7 VCs; and using as intermediate a 2-level group routes would be *lmlglmlglml* requiring only 6 VCs. However, only the original Valiant routing makes traffic completely uniform. This large number of VCs can be reduced by increasing one or both of the trunking levels, and applying the studied coloring techniques.

Considering two levels, a family of topologies between the Hamming graph and the dragonflies was built in Section 3.5 by modifying the parameter t. This is depicted by the horizontal line on top of Figure 3.9. With three levels, there are two parameters (t_2 and t_3) that can be modified, what extends the design space to a plane, represented in the lower part of the same Figure. Some of the most remarkable properties of this family of networks are presented in Table 3.2. There are three corner cases which are very relevant, being the first of them the canonical 3-level dragonfly without any trunking. The opposite

name	t_2	t_3	balancing conditions	link use relations	routers	compute nodes	general route
2-levels					ab	$ab\Delta_0$	
canonical dragonfly	1	-	$b = 1 + a(a-1)/2$	$\alpha \approx 1/2$	$\approx a^3/2$	$(R^4 + 4R^3 + 12R^2)/2^6$	*lgl*
Hamming $K_a \square K_a$	a	-	$a = b$	$\alpha = 1$	a^2	$(R+2)^3/3^3$	*lg*
3-levels					abc	$abc\Delta_0$	
3-level canonical dragonfly	1	1	$b = 1 + a(a-1)/2$, $c = 1 + b(b-1)/2$	$\alpha \approx \beta \approx 1/2$	$\approx a^7/16$	$R^4(R+2)^4/2^{14}$	*lmlglml*
Cascade-like	a	1	$a = b$, $c-1 = a^2(a-1)/2$	$\alpha = 1$, $\beta \approx 1/2$	$\approx a^5/2$	$(R^6 + 12R^5 + 54R^3)/(2^2 3^6)$	*lmglm*
—	1	b	$c - 1 = b - 1 \approx a(a-1)/3$	$\alpha \approx 1/3$, $\beta = 1$	$\approx a^5/3^2$	$R^6/(2^6 3^3)$	*lmlgl*
Hamming $K_a \square K_a \square K_a$	a	ab	$a = b = c$	$\alpha = \beta = 1$	a^3	$(R+3)^4/2^8$	*lmg*

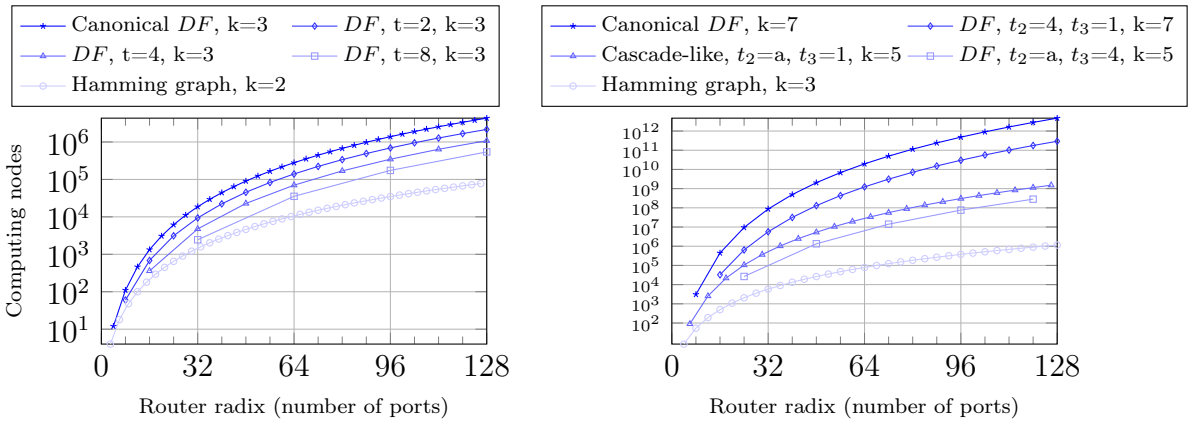
Table 3.2: Characteristics of the extreme cases (respect to trunking) of 2D and 3D balanced dragonflies. a, b and c routers per dimension. Δ_0 compute nodes per router. Routers with R ports (radix). Number of compute nodes approximate.

case is the 3D Hamming graph $K_a \square K_b \square K_c$, which is balanced for $a = b = c$. Notably, according to this classification there exists another 3-level corner configuration (using $t_2 = a$) which employs 2D Hamming graphs in the two lower levels, but no trunking in the highest level. This is equivalent to a 2-level dragonfly in which a Hamming graph is used for the local group topology, as in Cray Cascade [FBR+12]. It is due to the fact that the Hamming graph can be seen as a 2-level dragonfly as discussed in Section 3.5. Interestingly, their design combines route-restriction and distance-based deadlock-avoidance mechanisms (DOR in the 2D Hamming and increasing order of VCs otherwise).

The remaining corner case in the design space (the one with no name in Table 3.2) employs $t_3 = b$, as many global 3-level links as 2-level groups. With such trunking and a proper arrangement, a global link leads directly to the destination 1-level group, shortening minimal routes to *lmlgl*. This leads to $\beta = 1$ and $\alpha \gtrsim 1/3$. Larger values of trunking, up to $t_3 = ab$ could shorten paths to *lmlg*, but this clearly overdimensions the network.

In a n-level network, up to $\Delta_0 = \Delta_n$ compute nodes can be connected per router with maximum throughput. Larger values $\Delta_0 > \Delta_n$ lead to oversubscribed networks and lower values to waste of the network maximum bandwidth. For these concentration values, routers with radix $R = \sum_{i=0}^{n} \Delta_i = \Delta_0 + \Delta$ are required. Then for 2-level networks it is obtained $R = (a-1)(1+2\alpha)$ and for 3-level balanced networks $R = (a-1)(1+\alpha+2\alpha\beta)$. Table 3.2 summarizes the maximum number of compute nodes in a network ($abc\Delta_0$) for a given router radix R, along with balancing conditions and minimal routes employed in each case.

Figure 3.10 depicts the system size for different router radices and trunking levels, considering 2 and 3 levels. Notice the logarithmic vertical axis. Figure 3.10a corresponds to the upper line in Figure 3.9. The 2D Hamming graph ($t = a$, diameter $k = 2$) and the canonical 2-level dragonfly topology ($t = 1, k = 3$) are extreme cases. Between them, there are are multiple alternatives with variable trunking and smaller size than the canonical dragonfly. As discussed before, trunking is required to build systems smaller than the maximum achievable size for a given router radix. Figure 3.10b represents the scalability of certain designs in the lower rectangle of Figure 3.9, scaling from a 3D Hamming graph



(a) 2-level Dragonflies with trunking level t.

(b) 3-level Dragonflies with trunking levels t_1, t_2.

Figure 3.10: Scalability of different network configurations.

($t_2 = a$, $t_3 = a^2$, $k = 3$) to a Cascade-like dragonfly topology ($t_2 = a, t_3 = 1, k = 5$) and then to a canonical 3-level dragonfly without trunking (with diameter $k = 7$). These figures clearly highlight two issues: the trade off between diameter, degree and scale (the $d - k$ problem discussed in the introduction) and the need of global trunking to build systems that do not reach the maximum size for a given router and diameter, which can be in the order of millions of nodes.

3.8 Conclusions

Hamming graphs and dragonflies have been extensively studied in the technical literature. However, Hamming graphs have been revisited multiple times without recognizing its previous existence, whereas the dragonfly topology definition was very loose and not completely specified. This chapter has characterized topologically both networks, including their balancing conditions and provided precise definitions for the dragonfly topology. The relation between both graphs has been studied. With a proper global link arrangement, canonical dragonfly topologies are subgraphs of Hamming graphs. On the other hand, Hamming graphs can be seen as an extreme case of a dragonfly network with trunking, showing that both networks are actually part of the same broader family.

Based on this classification, the typical deadlock-free DOR mechanism used in Hamming graphs has been adapted to dragonflies with trunking, based on a coloring and ordering of the network resources. Trunking $t = 2$ allows for 3-hop paths while trunking $t = 4$ allows for 6-hop paths and traffic randomization, in both cases without a restriction on the number or use of virtual channels in the system. Evaluations in Section 5.7 will show that performance results are competitive with alternative mechanisms based on VCs, but they allow for implementations with more VCs to prevent Head of Line Blocking and increased performance, or less VCs to reduce implementation cost. The overall cost of this routing mechanism can be obviously higher than an equivalent VC-based routing, because it requires more router ports rather than more VCs. However, in many cases the required trunking is already employed to build a balanced dragonfly of a given size or for adding fault tolerance, so it would imply no extra cost. Finally, this routing would allow to leverage existing 2-level dragonfly router designs to multi-level dragonflies, thus increasing the maximum achievable system size with the same router design.

Chapter 4

Almost Optimal Lattice Graphs and Related Lee Codes

This chapter shows how the techniques developed for lattice graphs can offer deep insight into Lee codes. Reciprocally, good codes can be translated back into good topologies.

Section 4.1 details the relation between lattice graphs and linear Lee codes and gives some illustrative examples. In Section 4.2 all quasi-perfect codes over Gaussian and Eisenstein–Jacobi integers given by ideals are built. It is also shown that the codes over Gaussian integers generalize Lee codes over a bidimensional space. Section 4.3 builds a family of 2-quasi-perfect Lee codes with arbitrarily large dimension with density very close to the ones of perfect codes. Hence, it is an approach to the Golomb–Welch conjecture on the existence of perfect Lee codes.

4.1 The Relations Among Linear Lee Error Correcting Codes and Lattice Graphs

The lattice graphs introduced in Chapter 2 induce linear Lee-codes and *vice versa*. In this chapter this view of lattice graphs is adopted. Thus, this first section starts the chapter explaining this relation.

Since Lee codes are the target of our study, the natural space to be considered is the finite integer lattice $\mathbb{Z}_p^n$. However, for convenience, also the infinite lattice $\mathbb{Z}^n$ will be considered. Therefore, a *code* $\mathcal{C}$ will be a subset of either $\mathbb{Z}_p^n$ or $\mathbb{Z}^n$. This code is said to be *linear* or *lattice-like* if it is a subgroup of the corresponding space.

In the space $\mathbb{Z}^n$ it will be used the *Manhattan distance*. For any two words $\mathbf{v}, \mathbf{w} \in \mathbb{Z}^n$ its Manhattan distance is defined as:

$$D(\mathbf{v}, \mathbf{w}) = \sum_{j=1}^{n} |v_i - w_i|.$$

On the other hand, the *Lee distance* will be the metric when considering $\mathbb{Z}_p^n$. For $\mathbf{v}, \mathbf{w} \in \mathbb{Z}_p^n$ its Lee distance is defined as

$$D(\mathbf{v}, \mathbf{w}) = \sum_{j=1}^{n} \min\{|s| \mid s \equiv v_i - w_i \pmod{p},\ s \in \mathbb{Z}\}.$$

Since the Lee distance becomes the Manhattan distance for $p = \infty$, there will be no opportunity for confusion. In both cases the weight of a word v is defined as its distance

to the zero vector, which will be denoted as $|\mathbf{v}| = D(\mathbf{v}, \mathbf{0})$. For any positive integer r, the *Lee sphere* of radius r is defined as all the points with weight less or equal to r, that is:

$$B_r^n = \{\mathbf{v} \mid |\mathbf{v}| \leq r\}.$$

Note that, for any dimension $n \geq 1$, the cardinal $|B_2^n| = 2n^2 + 2n + 1$ [GW70] and that $|B_r^n| = |B_n^r|$.

A code $\mathcal{C}$ is said *t-error correcting* if t is the greatest integer such that for any word $\mathbf{w}$ there is at most one codeword $\mathbf{c} \in \mathcal{C}$ with $D(\mathbf{w}, \mathbf{c}) \leq t$. A code $\mathcal{C}$ is said *r-covering* if r is the smallest integer such that for any word $\mathbf{w}$ there is at least one codeword $\mathbf{c} \in \mathcal{C}$ with $D(\mathbf{w}, \mathbf{c}) \leq r$. Then, a code that is both t-error correcting and t-covering is said to be *perfect*. Golomb and Welch in [GW70] conjectured that there only exist perfect Lee codes for $t = 1$ or $n = 2$. Therefore, the existence of quasi-perfect codes must be studied since they are the best alternative to the perfect ones. Thus, a code that is t-error correcting and $(t + 1)$-covering is said to be *t-quasi-perfect*.

Codes and tilings are closely related as it is manifested in many papers. Given a linear code $\mathcal{C}$, the tessellation induced by the Voronoi regions of the codewords can be considered. The Voronoi region of a codeword is composed by the words that are closer to it than to other codewords. Since the code is linear, the tessellation is *congruent*, that is, all the tiles have the same shape and size. If $\mathcal{C}$ is a t-perfect error correcting code then the tiles obtained are translations of the Lee sphere of radius t. Otherwise, for a t-correcting and r-covering code, each induced tile contains a Lee sphere of radius t and it is contained in a Lee sphere of radius r. Reciprocally, the centers of the congruent tiles of a lattice-like tessellation can be used to define a linear code.

Once the tessellation induced by the linear code is obtained, then this tessellation can be used to define a Cayley graph. The set of the vertices of the graph are the words inside the tile centered at codeword 0. To define the adjacency, two different situations can be considered. For the vertices inside the tile, two vertices are adjacent if they are at a distance 1. In the case of vertices in the boundary of the tile, two vertices v and w are adjacent if there is a tile center c such that $D(u, c + v) = 1$. Note that, since the tessellation comes from a linear code, the graph is a undirected Cayley graph over the Abelian group $\mathbb{Z}^n/\mathcal{C}$. Now, if t is the greatest integer such that the Lee sphere B_t^n is contained in the tile, then this graph has $|B_t^n|$ vertices at distance t or less. By analogy to the concept of correction in codes, this value t will be referred as the *error correction capacity* of the graph. If r is the smallest positive integer such that B_r^n contains the tile, then the graph has diameter r. Given a Cayley graph it is straightforward to obtain a lattice-like congruent tilling. The tile can be defined by the representation in minimum distances of the set of vertices. Then, the tiling of the space is induced by the adjacency of the peripheral vertices.

This graph theoretical study of perfect codes can be seen as the reverse of the degree-diameter problem for Cayley graphs over Abelian finite groups [MŠ13]. In this problem, for a given diameter, graphs with the maximum possible number of vertices are searched. Specifically, for a positive integer t, graphs providing t-covering codes but without caring about the correction are looked for. Note that in this case, the order of the graphs obtained is lower than the cardinal of the corresponding sphere $|B_t^n|$. Therefore, in the present chapter graphs providing t-correcting codes and enforcing additionally $(t + 1)$-covering have been constructed. In our case, the order of the Cayley graphs is always greater than the cardinal of the sphere $|B_t^n|$. The degree-diameter problem for $t = 2$ and $t = 3$ has been considered in [MŠŠ12, Vet13]. In that papers families of graphs with smaller number of

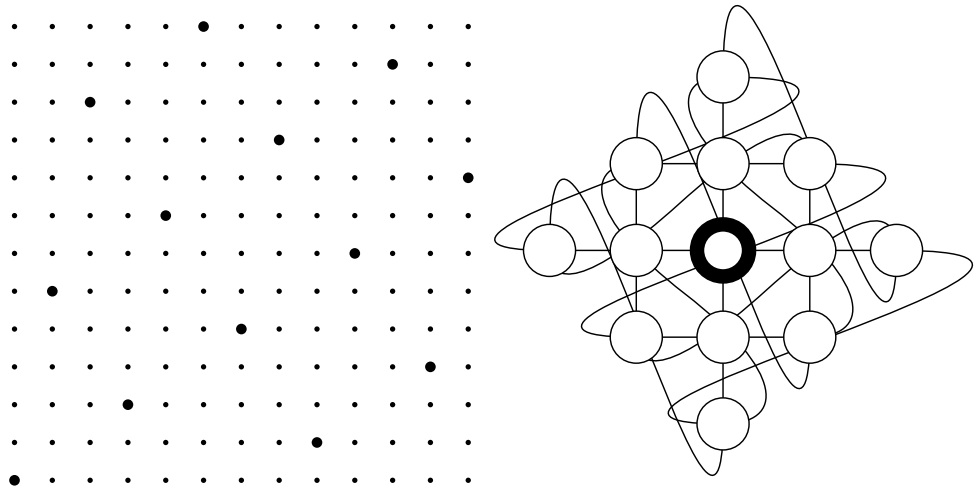


Figure 4.1: A 2-perfect code over $\mathbb{Z}_{13}^2$ and its associated lattice graph.

vertices than the sphere cardinal were given. Specifically, one of the graph constructions by Macbeth *et al.* [MŠŠ12] is given for infinitely many degrees $2n$ of graphs of diameter 2 and $\frac{3}{2}(n^2-1) = \frac{3}{4}|B_2^n| - \frac{3}{2}n - \frac{9}{4}$ vertices. Then, Vetrík [Vet13] constructs graphs with diameter 3 and $\frac{9}{128}(2n+3)^2(2n-5)$ vertices, which is asymptotically $\frac{27}{64}|B_3^n|$; it is remarkable that these graphs have error correction capacity 1 instead of the hoped 2, and thus they do not induce quasi-perfect codes. Note that a Cayley graph attaining the degree-diameter bound will induce a perfect code and *vice versa*.

For illustration of the graph–code relation, let us consider the following examples.

Example 4.1.1. *Perfect linear error correcting codes of dimension 1 become cycles. For example, the code $\mathcal{C} = \{5k \mid k \in \mathbb{Z}\} \subset \mathbb{Z}_q$ is a perfect linear 2-error correcting code over $\mathbb{Z}_q$ if q is multiple of 5. The graph obtained is the cycle of length 5, or equivalently, the lattice graph generated by the matrix $M = (5)$.*

Example 4.1.2. *Perfect linear 1-error correcting codes become complete graphs. Take for example Golomb and Welch perfect Lee code with dimension $n = 3$ and arity $q = 2n+1 = 7$. The lattice graph associated is generated by the matrix*

$$M = \begin{pmatrix} 7 & 2 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

It is clear that $\mathcal{G}(M)$ is isomorphic to the circulant graph $C_7(\pm 1, \pm 2, \pm 3)$, which in turn is isomorphic to the complete graph K_7, since $\pm 1, \pm 2 \pm 3$ is the whole set of possible adjacencies. With the matrix, the code can be expressed as $\mathcal{C} = \{M\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^3\}$, a 1-perfect linear 7-ary Lee code of length 3.

Example 4.1.3. *Perfect linear error correcting codes of dimension 2 become dense midimews (or Gaussian). In dimension $n = 2$, perfect codes of correction t are associated to the lattice graph generated by the matrix*

$$M = \begin{pmatrix} t+1 & -t \\ t & t+1 \end{pmatrix}$$

or its transpose. This graph is isomorphic to the Gaussian graph generated by $t + 1 + ti$ and to the dense midimew. The case $t = 2$ is shown in Figure 4.1.

It is remarkable that as a by-product of constructing good Lee-codes, good solutions are obtained for the resource distribution problem over lattice graphs. The *resource placement* problem consist on finding the optimal location for sources of some resource, trying to minimize the distance of every user to one of the resources. Thus the resources are the codewords and the minimum distance from an user to the closest resource is the covering radius. The distribution of the resources is optimal if the code is perfect. Specifically, for a lattice graph G, let $\mathcal{C}$ the associated linear Lee code to G. Then for any subgroup $\mathcal{C}' \subset \mathcal{C}$, $\mathcal{C}'$ gives a code over G with the same distance properties. For example, this was developed for tori in [BB96] and for Gaussian graphs in [MBS+08].

In the following, Section 4.2 will characterize quasi-perfect codes given by ideals over Gaussian and Eisenstein–Jacobi integers. The graphs related to those codes have few more vertices than the dense midimew and one more diameter. Finally, Section 4.3 will give a construction of 2-quasi-perfect Lee codes over arbitrarily large dimension, where the related graphs have twice the number of vertices than the theoretical perfect graph (conjectured not to exist), diameter 3 instead of 2, and many of them are Ramanujan graphs.

4.2 2D Quasi-Perfect Codes from Cayley Graphs over Integer Rings

The problem of searching for perfect codes has attracted great attention since the paper by Golomb and Welch in which the existence of these codes over Lee metric spaces was considered. Since perfect codes are not very common, the problem of searching for quasi-perfect codes has also a great interest. In this aspect, also quasi-perfect Lee codes have been considered for two and three dimensional Lee metric spaces. In this section constructive methods for obtaining quasi-perfect codes over metric spaces modeled by means of Gaussian and Eisenstein–Jacobi integers are given. The obtained codes form ideals of the integer ring thus preserving the property of being geometrically uniform codes. Moreover, they are able to correct more error patterns than the perfect codes that may properly be used in asymmetric channels. Therefore, the results in this section complement the constructions of perfect codes previously done for the same integer rings. Finally, decoding algorithms for the quasi-perfect codes obtained in this section are provided and the relationship of the codes and the Lee metric ones is investigated.

The remaining of the section is organized as follows. Subsection 4.2.1 details the related literature about quasi-perfect codes for the Lee-metric and quadrature amplitude modulation (QAM). In Subsection 4.2.2, some basic concepts from number theory that are necessary to define quotient rings of Gaussian and Eisenstein–Jacobi integers are presented. Moreover, graphs over quotients of these integer rings and codes over them are considered. Also, perfect and quasi-perfect codes over graphs are defined and geometrically uniform codes are constructed over them. In Subsection 4.2.3, an extension of perfect codes is made by presenting quasi-perfect codes over quotient rings of Gaussian integers. In addition, a complete characterization of quasi-perfect codes being ideals of the ring is given. Similarly to the previous subsection, in Subsection 4.2.4 quasi-perfect ideal codes over quotient rings of Eisenstein–Jacobi integers are characterized. In Subsection 4.2.5 the relation between the quasi-perfect codes presented in Subsection 4.2.3 and previously known Lee metric quasi-perfect codes is considered. Necessary and sufficient conditions for a group code being an ideal are given. As a consequence, the constructive method

presented in this section gives new quasi-perfect codes over two-dimensional Lee spaces. In Subsection 4.2.6 some decoding algorithms for the codes presented in this section are obtained. The decoding algorithms take advantage of the algebraic structure of the codes, that is, that they form ideals. Finally, in Subsection 4.2.7 some conclusions are drawn.

4.2.1 Related Work

Geometrically uniform codes were proposed by Forney [FJ91]. This class of codes encompasses the Slepian Group codes and the lattice codes by allowing the elements of the generator group be arbitrary isometries of the Euclidean space $\mathbb{R}^n$, instead of orthogonal transformations or translations when considering the previous classes separately. A space signal code is defined as *geometrically uniform* if for any two code sequences, there exists an isometry that takes a code sequence to the other while leaving the code invariant. Such a code has desirable symmetry properties such as: the Voronoi regions are congruent, the distance profile is the same for any codeword, each codeword has the same error probability, and the generator group is isomorphic to the permutation group acting transitively on the codewords.

In [GW70] Golomb and Welch define close-packed codes or perfect codes by the use of polyominoes, where each codeword has a decision region, its Voronoi region, given by Lee spheres of radius t. These regions satisfy the property that there is a group acting transitively on them that cover the torus and consequently, the resulting code is geometrically uniform. In [CMAPJ04] flat tori were used with a similar objective, as well as signal sets of the QAM-type considered as perfect coset codes with the induced distance from the Euclidean metric.

In [Hub94] and [Hub93] quotients of the Gaussian and Eisenstein–Jacobi (EJ) integer rings were proposed for modeling QAM-type and hexagonal signal constellations. Later, in [MBGG05, MBG07, MBG09, MMB06] a new metric over these spaces was introduced. This metric, similar to the Lee metric, is defined by means of the Gaussian and Eisenstein–Jacobi graphs, which are Cayley graphs over the integer rings modeling the signal constellation. In [MBS$^+$08] and [FB10], the main distance-related properties of Gaussian and Eisenstein–Jacobi graphs were characterized, providing closed expressions for their diameter and average distance. On the other hand, perfect codes were constructed as being ideals of the integer rings, thus solving the theoretical problem over the graphs known as the *perfect dominating set* of the vertices. This sections presents a complete characterization of quasi-perfect codes over Gaussian and EJ graphs that are ideals. Some preliminar work was done in [QPJ10, QPJ11].

The construction of quasi-perfect group codes was also considered in [AB03b] for the Lee metric. This kind of codes have shown to have different practical applications, as in phase modulated and multilevel quantized-pulse-modulated channels [GW70], [Ber68]. Moreover, they constitute the solution to the optimal resource allocation in toroidal interconnection networks as it was considered in [AB05] and [AB03a] for the two and three dimensional cases. In addition, decoding algorithms for Lee-distance quasi-perfect codes were presented in [AB03b] and [HA06]. Also, in [HG14] the authors presented some quasi-perfect codes for $n = 3$ and a few radii.

The aim of this section is to provide the construction of quasi-perfect codes over the Gaussian and Eisenstein–Jacobi integers, which besides preserving the geometrically uniform structure of the codes they are able to correct more error patterns than the perfect codes. Moreover, since in [MMB06] the relationship between perfect codes for

the two-dimensional Lee spaces and perfect codes over Gaussian integers was shown, the relation between the construction given in this section with the quasi-perfect Lee codes is also investigated. It will be shown that new quasi-perfect codes over two-dimensional spaces can be obtained from the methods given here. The new quasi-perfect codes not only are groups but also form ideals over the integers rings.

Finally, decoding algorithms are given for the presented codes that take advantage of the ideal structure of the constructions. The algorithms that will be considered have some geometrical similarities to the one presented in [HA06] for the Lee metric although they decode different code constructions.

4.2.2 Preliminary Results

This subsection is organized into three parts. In the first one, quotients of Gaussian and Eisenstein–Jacobi rings, which will be used to design signal constellations, are introduced. In the second one, Cayley graphs over these quotient rings are considered in order to define metrics over the previous rings. Finally, the third one defines perfect and quasi-perfect codes over these graphs.

Quotient of integer rings

Next, basic results from number theory that are important to the development of the remaining sections are presented. More detailed information can be found in [Sam67], [Hun74] and [HW79].

Let $\mathbb{K}$ be a number field with degree n and $\sigma_1, \sigma_2, \ldots, \sigma_n$ the monomorphisms of $\mathbb{K}$ into $\mathbb{C}$. For any $\alpha \in \mathbb{K}$ the norm of α is defined as

$$\mathcal{N}(\alpha) = \prod_{i=1}^{n} \sigma_i(\alpha).$$

Since the σ_i's are monomorphisms, it follows that

$$\mathcal{N}(\alpha\beta) = \mathcal{N}(\alpha)\mathcal{N}(\beta),$$

for any $\alpha, \beta \in \mathbb{K}$. In addition, if $\alpha \neq 0$, then $\mathcal{N}(\alpha) \neq 0$.

Now, given a number field $\mathbb{K}$, let $I_{\mathbb{K}}$ be the ring of integers of $\mathbb{K}$ and $I_{\mathbb{K}}\alpha = (\alpha)$ the ideal of $I_{\mathbb{K}}$ generated by α. Then, the following result is obtained.

Proposition 4.2.1. *[Sam67, Proposition 1, p.52] Let $0 \neq \alpha \in I_{\mathbb{K}}$. Then*

$$\mathcal{N}(\alpha) = card\left(\frac{I_{\mathbb{K}}}{I_{\mathbb{K}}\alpha}\right),$$

that is, the quotient ring $\frac{I_{\mathbb{K}}}{I_{\mathbb{K}}\alpha}$ has $\mathcal{N}(\alpha)$ elements.

The codes presented in this section will be constructed over signal constellations modeled by Gaussian and Eisenstein–Jacobi integers. Therefore, let us first consider the number field $\mathbb{K} = \mathbb{Q}[i] = \{a + bi \mid a, b \in \mathbb{Q}\}$, where $i = \sqrt{-1}$. The ring of integers of $\mathbb{Q}[i]$ is $\mathbb{Z}[i]$, called the ring of the *Gaussian integers* and denoted by $\mathbb{Z}[i] = \{a + bi \mid a, b \in \mathbb{Z}\}$. If $\alpha = a + bi \in \mathbb{Z}[i]$ then its *norm* is given by

$$\mathcal{N}(\alpha) = \alpha\bar{\alpha} = (a + bi)(a - bi) = a^2 + b^2,$$

where $\bar{\alpha}$ is called the *conjugate* of α. That is, the norm is given by a quadratic form $\mathcal{N}$ such that

$$\begin{array}{rccc} \mathcal{N}: & \mathbb{Z}[i] & \longrightarrow & \mathbb{Z}^+ \\ & a+bi & \longmapsto & a^2+b^2. \end{array}$$

Now, let $\mathbb{Z}[\omega]$, called the ring of the *Eisenstein–Jacobi integers* and denoted by $\mathbb{Z}[\omega] = \{a+b\omega \mid a,b \in \mathbb{Z}\}$, where $\omega = \frac{1+\sqrt{-3}}{2}$. Note that ω is such that $\omega^2 - \omega + 1 = 0$. Hence, if $\alpha = a+b\omega \in \mathbb{Z}[\omega]$ then its norm is given by

$$\mathcal{N}(\alpha) = \alpha\bar{\alpha} = (a+b\omega)(a-b\omega^2) = (a+b\omega)((a+b)-b\omega) = a^2+b^2+ab,$$

that is, in this case its norm is given by a quadratic form $\mathcal{N}$ such that

$$\begin{array}{rccc} \mathcal{N}: & \mathbb{Z}[\omega] & \longrightarrow & \mathbb{Z}^+ \\ & a+b\omega & \longmapsto & a^2+b^2+ab. \end{array}$$

If (α) denotes the ideal of $\mathbb{Z}[\rho]$ generated by α, where either $\rho = i$ or $\rho = w$, then the quotient ring generated by such an ideal is

$$\mathbb{Z}[\rho]_\alpha = \frac{\mathbb{Z}[\rho]}{(\alpha)},$$

where $\alpha \in \mathbb{Z}[\rho]_\alpha$. From Proposition 4.2.1 it follows that:

Theorem 4.2.2. *[MBG07], [MBG09] Let $0 \neq \alpha \in \mathbb{Z}[\rho]$ with $\rho \in \{i, \omega\}$. Then, $\mathbb{Z}[\rho]_\alpha$ has $\mathcal{N}(\alpha)$ elements.*

Moreover, using the third ring isomorphism theorem, [Hun74], it can be easily inferred the following consequence of the previous result.

Corollary 4.2.3. *[MBG07], [MBG09] Let $0 \neq \alpha \in \mathbb{Z}[\rho]$ with $\rho \in \{i, \omega\}$.*

i) If $\beta \in \mathbb{Z}[\rho]$ divides α, then the ideal $(\beta) \subset \mathbb{Z}[\rho]_\alpha$ has order $\mathcal{N}(\alpha)/\mathcal{N}(\beta)$;

ii) If $\beta \in \mathbb{Z}[\rho]$ does not divide α and $\gamma = gcd(\alpha, \beta)$, then the ideal $(\beta) \subset \mathbb{Z}[\rho]_\alpha$ is generated by γ and has order $\mathcal{N}(\alpha)/\mathcal{N}(\gamma)$.

Example 4.2.4. *Given $\alpha = 3+4i$ then its norm is $\mathcal{N}(\alpha) = 25$. Hence, from Theorem 4.2.2, $\mathbb{Z}[i]_{3+4i}$ has 25 elements, which are obtained from the quotient of the ring $\mathbb{Z}[i]$ by the ideal $(3+4i)$, or equivalently, by taking $\mathbb{Z}[i]$ modulo $(3+4i)$. Hence, $\mathbb{Z}[i]_{3+4i} = \{0, 1, 2, 3, -1, -2, -3, i, 2i, 3i, -i, -2i, -3i, 1+i, 1+2i, 1-i, 1-2i, -1-i, -1-2i, -1+i, -1+2i, 2+i, -2+i, 2-i, -2-i\}$.*

On the other hand, if $\alpha = 3+4\omega$, its norm is $\mathcal{N}(\alpha) = 37$ and the induced quotient ring $\mathbb{Z}[\omega]_{3+4\omega}$ has 37 elements. In Figure 4.2 both quotients are graphically represented.

The ring $\mathbb{Z}[\rho]_\alpha$ is a field if and only if α is a prime of $\mathbb{Z}[\rho]$. It is worth remembering that Gaussian primes fall into two categories:

- $\alpha = p$ for some $p \in \mathbb{Z}$ prime over the integers satisfying $p \equiv 3 \pmod 4$. Its norm is $\mathcal{N}(\alpha) = p^2$.
- $\alpha = a+bi$ with norm $\mathcal{N}(\alpha) = a^2+b^2 = p$ for some integer prime p. This integer must satisfy $p = 2$ or $p \equiv 1 \pmod 4$.

In the special case of $\mathbb{Z}[i]_p$ for some prime p, the norm functions become a function into $\mathbb{Z}_p$ and $\mathcal{N}(\zeta) = 0$ for $\zeta \in \mathbb{Z}[i]_p$ if and only if ζ is a zero divisor.

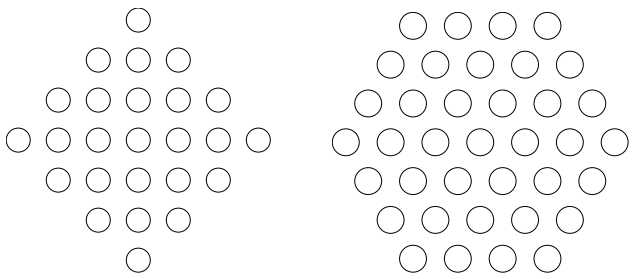


Figure 4.2: Signal constellations obtained as $\mathbb{Z}[i]_{3+4i}$ and $\mathbb{Z}[\omega]_{3+4\omega}$

Graphs over Gaussian and Eisenstein–Jacobi integer rings

Since the codes presented in this section are obtained from quotients of Gaussian and Eisenstein–Jacobi integers, metrics over these rings must be defined. In this direction, Gaussian and Eisenstein–Jacobi graphs (EJ-graphs) are defined as Cayley graphs over the quotient rings, so the metric induced by these graphs will be the one considered for the code construction. Hence, Gaussian and Eisenstein–Jacobi graphs are Cayley graphs where the corresponding adjacency sets are the units of the integer rings.

Definition 4.2.5. *Let $0 \neq \alpha \in \mathbb{Z}[\rho]$, where $\rho \in \{i, \omega\}$.*

- *If $\rho = i$ then the Gaussian graph generated by α is defined as*
$$G_\alpha = Cay(\mathbb{Z}[i]_\alpha; \{\pm 1, \pm i\}).$$
- *If $\rho = w$ then the Eisenstein–Jacobi graph generated by α is defined as*
$$EJ_\alpha = Cay(\mathbb{Z}[w]_\alpha; \{\pm 1, \pm w, \pm w^2\}).$$

As it has been remarked in the previous subsection, the order of the graphs is given by the norm of its generator. Clearly, Gaussian graphs are regular of degree 4 and EJ-graphs have degree 6. Since they have been defined as Cayley graphs, they result in vertex-symmetric graphs, that is, for any pair of vertices there is an automorphism that sends one into the other. As a consequence, the distance distribution of the vertices can be determined by counting the number of vertices at each distance from a central vertex, which is usually selected to be zero. The complete determination of these distributions has been done in [MBS+08] and [FB10] for Gaussian and EJ-graphs, respectively. As a consequence, the *diameter* of the graph, that is, the length of the longest shortest path has been exactly determined. Next, the results summarizing the distance distributions are presented in order to be self-contained.

Theorem 4.2.6. *[MBS+08] Let $0 \neq \alpha = a + bi \in \mathbb{Z}[i]$ and $0 \leq a \leq b$. Let $T = \frac{a+b}{2}$ and for any positive integer t, let $W(t)$ be the number of vertices in G_α at a distance t. Then*

$$W(t) = \begin{cases} 1 & \text{if } t = 0 \\ 4t & \text{if } 1 \leq t < T \\ 2(b-1) & \text{if } t = T < b \\ 2b-1 & \text{if } t = T = b \\ 4(b-t) & \text{if } T < t < b \\ 1 & \text{if } T < t = b \text{ and } a \equiv b \pmod 2 \\ 0 & \text{if } b < t. \end{cases}$$

Theorem 4.2.7. *(Fixed from [FB10]) Let $0 \neq \alpha = a + b\omega \in \mathbb{Z}[\omega]$ and $a \geq b \geq 0$. Let $T = \frac{a+b}{2}$ and $M = \frac{2a+b}{3}$. For any positive integer t, let $W(t)$ be the number of vertices in EJ_α at a distance t. Then*

$$W(t) = \begin{cases} 1 & \text{if } t = 0 \\ 6t & \text{if } 1 \leq t < T \\ 3a - 3 & \text{if } t = T < M \\ 3a - 1 & \text{if } t = T = M \\ 18(M - t) & \text{if } T < t < M \\ 2 & \text{if } T < t = M \text{ and } a \equiv b \pmod 3 \\ 0 & \text{if } M < t. \end{cases}$$

Example 4.2.8. *In Example 4.2.4, the set of vertices is $V = \mathbb{Z}[i]_{3+4i}$ and the set of edges as in the previous definition, is shown in Figure 4.3. The diameter of the graph is 3 since every vertex is at distance less than or equal to 3 from the central vertex. Moreover, as can be seen this graph has the maximum number of vertices for diameter 3, or equivalently, it is dense[1]. On the other hand, the graph $EJ_{3+4\omega}$ has the set of vertices $V = \mathbb{Z}[\omega]_{3+4\omega}$ and the adjacency is completed as shown in Figure 4.3. In this case the diameter of the graph is also 3, and the graph is also dense.*

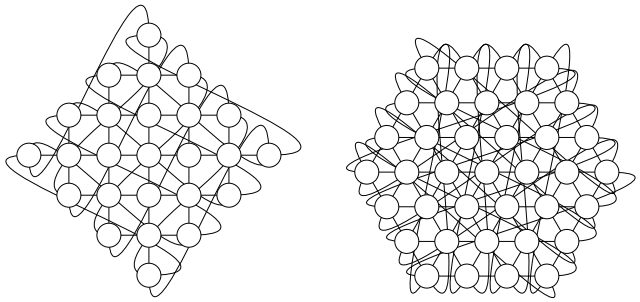


Figure 4.3: The graphs G_{3+4i} and $EJ_{3+4\omega}$

Now, to define a metric over the integer rings considered in this section it is only needed to consider the distance induced by its corresponding Gaussian or Eisenstein–Jacobi graph.

Definition 4.2.9. *[MBG07, MSBG08] Let $0 \neq \alpha \in \mathbb{Z}[\rho]$, where $\rho \in \{i, \omega\}$. The distance in $\mathbb{Z}[\rho]_\alpha$ is the distance induced by the associated Cayley graph G_α or EJ_α. Thus, if $\eta,\ \tau \in \mathbb{Z}[\rho]_\alpha$, the graph distance can be computed as:*

i) $D_\alpha(\eta, \tau) = \min\{|x| + |y| \mid x, y \in \mathbb{Z} \text{ such that } \tau - \eta \equiv x + yi \pmod \alpha\}$.

ii) $D_\alpha(\eta, \tau) = \min\{|x| + |y| + |z| \mid x, y, z \in \mathbb{Z} \text{ such that } \tau - \eta \equiv x + y\omega + z\omega^2 \pmod \alpha\}$.

Remark 4.2.10. *Note that the distance between any two vertices is the length of any shortest path between them. Then, the diameter of the graph gives the maximum distance in the metric space. As a consequence, signal constellations corresponding to dense graphs contain the maximum number of signal points for a given maximum distance.*

[1]Note that this is a different definition of dense graph from the usual one, which refers to the number of edges instead of the number of vertices.

Geometrically uniform codes over quotient rings

Once the metric spaces to be considered for quasi-perfect codes constructions have been established, classical definitions of codes will be provided in this subsection.

Given a graph G and distance D, a *code* in G is a nonempty subset $\mathcal{C}$ of $V(G)$. The *Voronoi region* $\mathcal{V}_\eta$ associated with $\eta \in \mathcal{C}$ is the subset of the elements in V for which η is the closest point in $\mathcal{C}$, that is,

$$\mathcal{V}_\eta = \{\tau \in V \mid D(\eta, \tau) = D(\tau, \mathcal{C})\}.$$

From this, the *covering radius* of the code is defined as

$$t = \max\{D(\eta, \mathcal{C}) \mid \eta \in V\}.$$

Let $B_t(\eta) = \{\tau \in V \mid D(\eta, \tau) \leq t\}$ denote the ball of radius t centered at η. Then the covering radius is the least number t such that the balls of radius t centered at the points of $\mathcal{C}$ cover V. Then,

$$\delta = \min\{D(\eta, \tau) \mid \eta, \tau \in \mathcal{C}, \eta \neq \tau\},$$

is the *minimum distance* of $\mathcal{C}$, with $\delta \leq 2t + 1$. The equality holds, that is $\delta = 2t + 1$, when the balls of radius t centered at the points of $\mathcal{C}$ partition V. A code satisfying this property is called *perfect* and it is said to correct t errors. Next, perfect and quasi-perfect codes, which are the target of this section, are defined.

Definition 4.2.11. *Let $G = (V, E)$ be a graph and D denote its distance. Let $\mathcal{C} \subset V$.*

i) $\mathcal{C}$ is a t-quasi-perfect code if

- *For every pair of different codewords $c, c' \in \mathcal{C}$ it follows that $B_t(c) \cap B_t(c') = \emptyset$.*
- *For every vertex $v \in V$ there exists $c \in \mathcal{C}$ such that $D(c, v) \leq t + 1$.*

ii) $\mathcal{C}$ is a t-perfect code if for every vertex $v \in V$ there exists a unique codeword $c \in \mathcal{C}$ such that $D(c, v) \leq t$.

Perfect codes being ideals for the Gaussian and Eisenstein–Jacobi graphs were obtained in [MBGG05] and [MBG07]. These codes have the property of being generated by elements with maximum norm for a given diameter, or equivalently, as codes associated with dense graphs. The following result summarizes how to obtain perfect codes over Gaussian and EJ-graphs.

Theorem 4.2.12. *[MBG07, MSBG08] Let $0 \neq \alpha = a + b\rho \in \mathbb{Z}[\rho]$, $\rho \in \{i, \omega\}$ and t be a positive integer.*

i) If $\rho = i$ and $\beta = t + (t + 1)i$ (or $\overline{\beta}$) divides α, then the ideal (β) (resp. $(\overline{\beta})$) is a t-perfect code over G_α.

ii) If $\rho = \omega$ and $\beta = t + (2t + 1)\rho$ (or $\overline{\beta}$) divides α, then the ideal (β) (resp. $(\overline{\beta})$) is a t-perfect code over EJ_α.

Note that since the construction is made by means of ideals of the integer ring, the resulting codes are not only perfect but also geometrically uniform. In fact, it is straightforward that any ideal over the quotients generates a code over the graph, as it is proved in the following result.

t	0	1	2	3	4	5	6	7	8	9
$W(t)$	1	4	8	12	16	20	24	12	8	4

Table 4.1: Distance distribution of G_{3+10i}.

Corollary 4.2.13. *Let $0 \neq \alpha, \beta \in \mathbb{Z}[\rho]$, $\rho \in \{i, \omega\}$. Then, if β divides α, then the ideal (β) forms a geometrically uniform code over $\mathbb{Z}[\rho]_\alpha$. Moreover, let $\beta = a + b\rho$,*

- *If $\rho = i$ then the code can correct every error pattern of weight t, for $t < \frac{|a|+|b|}{2}$*
- *If $\rho = \omega$ then the code can correct every error pattern of weight t, for $t < \frac{|a|+|b|}{2}$.*

Proof. The ideal generates a geometrically uniform code straightforwardly. On the other hand, the error correction capacity is obtained as a consequence of Theorems 4.2.6 and 4.2.7. □

Example 4.2.14. *Given $\alpha = 16 + 17i$ then $\mathcal{N}(\alpha) = 545$. Hence, from Theorem 4.2.2, $\mathbb{Z}[i]_{16+17i}$ has 545 elements. Now, $\alpha = 16 + 17i$ may be written as $16 + 17i = (-i)(1 + 2i)(3 + 10i)$. Therefore, $\beta = 3 + 10i$ generates a geometrically uniform code over G_α that corrects every error pattern of weight $t = 6$. Moreover, the distance distribution of the graph G_β can be directly inferred from Theorem 4.2.6, which is shown in Table 4.1. Therefore, the code generated by β corrects 12 error patterns with $t + 1 = 7$ errors, 8 with $t + 2 = 8$ and 4 with $t + 3 = 9$, resulting in the 24 errors. This geometrically uniform code is shown in Figure 4.4. As it can be checked, the code is not a perfect code neither a quasi-perfect code.*

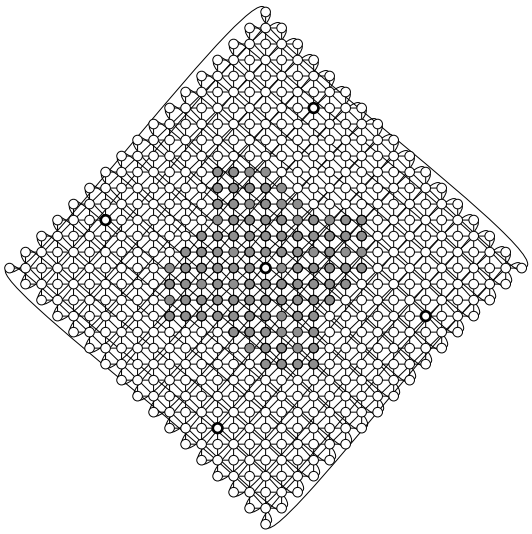


Figure 4.4: Geometrically uniform code generated by $3 + 10i$ over G_{16+17i}

In the next subsections the problem of characterizing quasi-perfect codes over Gaussian and EJ-graphs is considered, thus complementing the previous works on perfect codes over Gaussian and EJ-graphs. The codes considered will also be generated as an ideal, thus obtaining geometrically uniform codes. Also, being ideals will simplify the decoding procedures. Finally, by using Theorems 4.2.6 and 4.2.7 the distance distribution of the codewords can be calculated as it was done in previous example.

4.2.3 Quasi-Perfect Codes over Quotient Rings of Gaussian Integers

In this subsection a constructive method for obtaining quasi-perfect codes over Gaussian integer rings is presented. As it has been mentioned before, given the signal constellation $\mathbb{Z}[i]_\alpha$ equipped with the Gaussian metric, it is enough to choose β a divisor of α, such that β is either of the form $t+(t+1)i$ or $t-(t+1)i$, to obtain a perfect code. The code is defined as the ideal generated by the divisor and the signal constellation is covered by the fundamental region whose covering radius has maximum value of t. That is, the fundamental regions consist of Lee spheres of radius t. Thus, perfect codes are obtained by translations of Lee spheres of radius t. As observed in [GW70], a Lee sphere with radius r has $2r^2+2r+1$ cells. By Corollary 4.2.13, any β being a divisor of α, defines a new code, although not necessarily a perfect code, but with the property of covering the signal constellation by identical fundamental regions. In this section the characterization of a divisor β such that the ideal generates a quasi-perfect code is done. In this aim, some distance properties of Gaussian graphs will be needed. In Theorem 4.2.6 the vertices distance distribution has been presented and as consequence, it can be obtained in the following result:

Corollary 4.2.15. *Let $\alpha = a+bi \in \mathbb{Z}[i]$ and consider G_α. Then,*

- *The value $t = \lfloor \frac{|a|+|b|-1}{2} \rfloor$ gives the radius of the maximum Lee sphere contained in the Voronoi region associated to G_α.*
- *If $\mathcal{N}(\alpha)$ is odd, the value $k = \max\{|a|,|b|\}-1$ is the maximum distance from any word to the center of the Voronoi region. If $\mathcal{N}(\alpha)$ is even, the value $k = \max\{|a|,|b|\}$ is the maximum distance from any word to the center of the Voronoi region.*

Proof. This corollary is a consequence of Theorem 4.2.6. □

As a consequence a constructive method for quasi-perfect codes that gives a complete characterization can be obtained as presented in Theorem 4.2.16.

Theorem 4.2.16. *Let $\alpha \in \mathbb{Z}[i]$ and t be a positive integer. Let $\beta \in \{(t-1)+(t+2)i, t+(t+1)i, (t+1)+(t+1)i\}$. Then,*

- *If β divides α, then the ideal (β) forms a t-quasi-perfect code over $\mathbb{Z}[i]_\alpha$.*
- *If $\overline{\beta}$ divides α, then the ideal $(\overline{\beta})$ forms a t-quasi-perfect code over $\mathbb{Z}[i]_\alpha$.*

In both cases the code can correct every error pattern of weight t and $\mathcal{N}(\beta)-(2t^2+2t+1)$ error patterns with weight $t+1$. Moreover, these are the unique ideals that form quasi-perfect codes over $\mathbb{Z}[i]_\alpha$.

Proof. Let us consider the first item since the other one can be demonstrated in a similar way. Now, for every $\beta = x+yi \in \mathbb{Z}[i]_\alpha$ it is of the form $\delta_{t,h} = t+(t+1)i+(-h+hi)$ or $\delta'_{t,h} = (t+1)+(t+1)i+(-h+hi)$ with integers t,h, depending on the parity of its norm. It is enough to consider the values $t = \frac{x+y-1}{2}$ and $h = \frac{y-x-1}{2}$ for β with odd norm and $t = \frac{x+y-2}{2}$ and $h = \frac{y-x}{2}$ for β with even norm. Hence, it can be assumed that $y \geq x \geq 0$, which implies $t,h \geq 0$.

First, consider the case that $\beta = \delta_{t,h} = (t-h)+(t+1+h)i$ divides α and define $\mathcal{C} = (\delta_{t,h})$. Now, given $c, c' \in \mathcal{C}$ two different codewords, it has to be proved that $B_t(c) \cap B_t(c')$ is

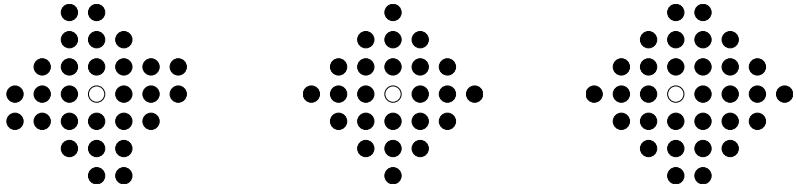


Figure 4.5: The 3 tiles of the 3-quasi-perfect codes over $\mathbb{Z}[i]$.

an empty set. Clearly, this can be obtained from the first item of Corollary 4.2.15 since $\lfloor \frac{|t-h|+|t+h+1|-1}{2} \rfloor = t$ gives the radius of the maximum Lee sphere contained in the Voronoi region associated to G_α.

Now, let us consider $v \in \mathbb{Z}[i]_\alpha$. Then, by Corollary 4.2.15 it is obtained that $k = \max\{|t-h|, |t+1+h|\} - 1 = t + h$ is the maximum distance from any word to the center of the Voronoi region. Therefore, for any $v \in \mathbb{Z}[i]_\alpha$ there must exist a codeword $c \in \mathcal{C}$ such that $D_\alpha(v, c) \leq t + 1$. Now, $k \leq t + 1$ if and only if $0 \leq h \leq 1$, thus obtaining the first two values for β in the theorem. Moreover, $h = 0$ gives us the t-perfect code.

Second, let us assume that $\beta = \delta'_{t,h} = (t+1) + (t+1)i + (-h + hi)$. The proof is similar to the previous case, however for the even cases of Gaussian generators. Note that $\lfloor \frac{|t+1-h|+|t+1+h|-1}{2} \rfloor = t$ gives the radius of the maximum Lee sphere contained in the Voronoi region associated to G_α. Now, since $k = \max\{|t+1-h|, |t+1+h|\} = t + 1 + h$ it follows that this is the maximum distance from any word to the center of the Voronoi region. Then, clearly $k \leq t + 1$ if and only if $h = 0$.

To conclude with the proof just note that the norm of β is cardinal of the Voronoi region generated by β and $2t^2 + 2t + 1$ is the number of vertices at a distance less or equal to t.

□

The distance distribution of these 3 quasi-perfect codes is shown in Figure 4.5 for error correction capacity $t = 3$. Note that the one in the middle corresponds with the perfect code and that vertex zero is highlighted in white.

Example 4.2.17. *Let us consider $h = 1$, $t = 3$, which implies $\delta_{3,1} = 2 + 5i$. Hence, for any multiple α of δ a Gaussian ring with a 3-quasi-perfect code is obtained. Let us consider for example $\alpha_1 = -8 + 9i = (1 + 2i)(2 + 5i)$. Then, the ideal $(2 + 5i) \in \mathbb{Z}[i]_{-8+9i}$ forms a 3-quasi-perfect code with $\mathcal{N}(1 + 2i) = \frac{\mathcal{N}(-8+9i)}{\mathcal{N}(2+5i)} = 5$ codewords. Now, if α_2 is a multiple of the previous generator, that is, $\alpha_2 = -9 + 21i = (3 + 3i)(2 + 5i)$, then the ideal $(2 + 5i) \in \mathbb{Z}[i]_{-9+21i}$ forms again a 3-quasi-perfect code however with $\mathcal{N}(3 + 3i) = \frac{\mathcal{N}(-9+21i)}{\mathcal{N}(2+5i)} = 18$ codewords. A graphical representation of both sets over their corresponding Gaussian graphs is shown in Figure 4.6.*

Remark 4.2.18. *Note that the uniqueness is strongly obtained from the condition of being an ideal. If this condition is relaxed, codes that are not obtained from the construction given in Theorem 4.2.16 can be more or less straightforwardly constructed. Example 4.2.19 shows one of these possible codes. A special case of quasi-perfect codes being groups but not ideals, will be discussed in Subsection 4.2.5.*

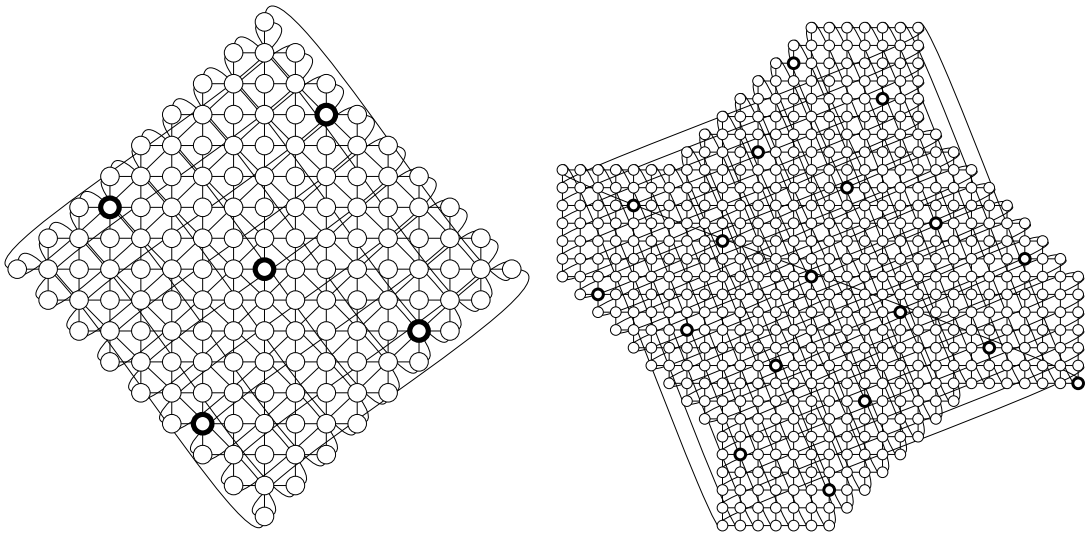


Figure 4.6: 3-quasi-perfect codes generated by $2 + 5i$ over G_{-8+9i} and G_{-9+21i}.

Example 4.2.19. *Let us consider $\alpha = 23 \in \mathbb{Z}[i]$. The metric space induced by the corresponding Gaussian graph has $\mathcal{N}(\alpha) = 23^2 = 529$ elements. Let us consider the subset formed by the 46 elements depicted in Figure 4.7 as bolded points. It can be checked that this subset forms a quasi-perfect code over $\mathbb{Z}[i]_{23}$, but it neither forms an ideal nor a group of the ring. Moreover, the obtained code is not geometrically uniform. Note that $\mathbb{Z}[i]_{23}$ is isomorphic to the two-dimensional Lee space $\mathbb{Z}_{23} \times \mathbb{Z}_{23}$. More details about the relationship between quasi-perfect codes over two-dimensional Lee spaces and the ones presented in this section will be discussed in Subsection 4.2.5.*

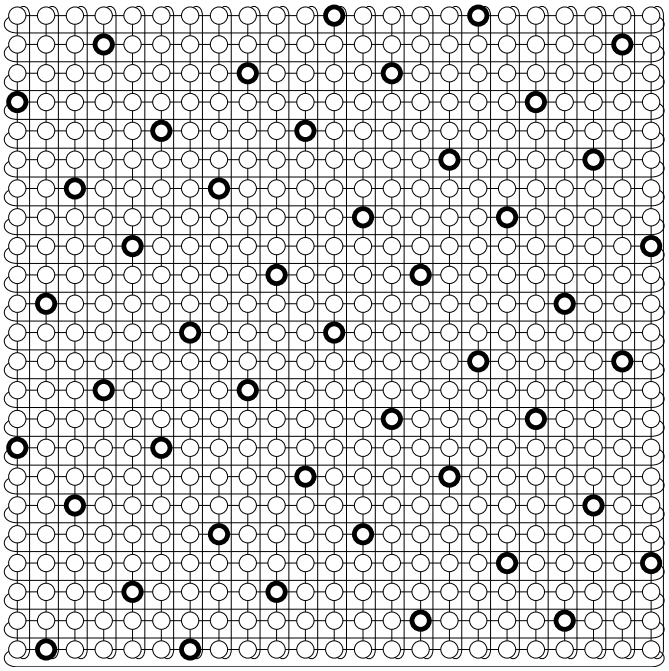


Figure 4.7: A non geometrically uniform quasi-perfect code over $\mathbb{Z}[i]_{23}$

4.2.4 Quasi-Perfect Codes over Eisenstein–Jacobi Integer Rings

In this subsection, the characterization of quasi-perfect codes over EJ-graphs generated by ideals is considered. Analogously to the Gaussian case, perfect codes over EJ-graphs can be obtained over Eisenstein–Jacobi integers modulo α by choosing β a divisor of α of the form $(t+1)+t\omega$ or its conjugate, as it was shown in [MSBG08]. Then, the ideal generated by such a divisor forms a perfect code and the signal constellation is covered by fundamental regions whose covering radius has maximum value of t, that is, the fundamental regions are hexagons with radius t. Thus, perfect codes are obtained by translations of the hexagons with radius t that have $3t^2+3t+1$ cells. Therefore, if a different divisor is considered, the generated code is not perfect anymore. In Theorem 4.2.21 the adequate values for a divisor such that it generates a quasi-perfect code are stated. Moreover, it is also shown that the divisors provided are the only ones if the wanted code has to be an ideal.

In order to construct such codes, some distance properties of EJ-graphs should be considered first. The following result is a consequence of the distance distribution of EJ-graphs given in [FB10] and summarized in Theorem 4.2.7.

Corollary 4.2.20. *Let $\alpha = a+b\omega \in \mathbb{Z}[\omega]$ and consider EJ_α. Then,*

- *The value $t = \lfloor \frac{|a|+|b|-1}{2} \rfloor$ gives the radius of the maximum Lee sphere contained in the Voronoi region associated to EJ_α.*
- *The value $k = \lfloor \max\{|\frac{2a+b}{3}|, |\frac{a+2b}{3}|, |\frac{a-b}{3}|\} \rfloor$ is the maximum distance from any word to the center of the Voronoi region.*

The next result that gives a complete characterization of ideals over EJ-graphs that form quasi-perfect codes.

Theorem 4.2.21. *Let $\alpha \in \mathbb{Z}[\omega]$ and let t be a positive integer. Let $\beta \in \mathbb{Z}[\omega]$ be such that $\beta|\alpha$ and $\beta \in \{(t+1)+t\omega, (t+1)+(t+1)\omega, (t+2)+t\omega, (t+2)+(t-1)\omega, (t+3)+(t-1)\omega, (t+3)+(t-2)\omega, (t+4)+(t-3)\omega, \}$. Then, the ideal $\mathcal{C} = (\beta)$ is a t-quasi-perfect code over $\mathbb{Z}[\omega]_\alpha$. Moreover, these are the only (up to units multiplication and conjugation) t-quasi-perfect codes being an ideal and $\beta = (t+1)+t\omega$ generates a perfect code.*

Proof. Let us consider $\beta = x+y\omega$ such that $x \geq y \geq 0$, $2x+y \equiv p \pmod 3$, $0 \leq p < 3$ and $x+y-1 \equiv q \pmod 2$, $0 \leq q < 2$. Then, by Corollary 4.2.20 the β looked for are such that:

$$\left\lfloor \frac{2x+y}{3} \right\rfloor \leq \left\lfloor \frac{x+y-1}{2} \right\rfloor + 1.$$

Now, two cases are considered separately:

i) $\lfloor \frac{2x+y}{3} \rfloor = \lfloor \frac{x+y-1}{2} \rfloor$;

ii) $\lfloor \frac{2x+y}{3} \rfloor = \lfloor \frac{x+y-1}{2} \rfloor + 1$.

For the first case, it follows, for some integers $0 \leq p < 3$, $0 \leq q < 2$, that:

$$\frac{2x+y-p}{3} = \frac{x+y-1-q}{2}.$$

As a consequence $x = y-3+2p-3q$, which gives the following possible values for $x-y$:

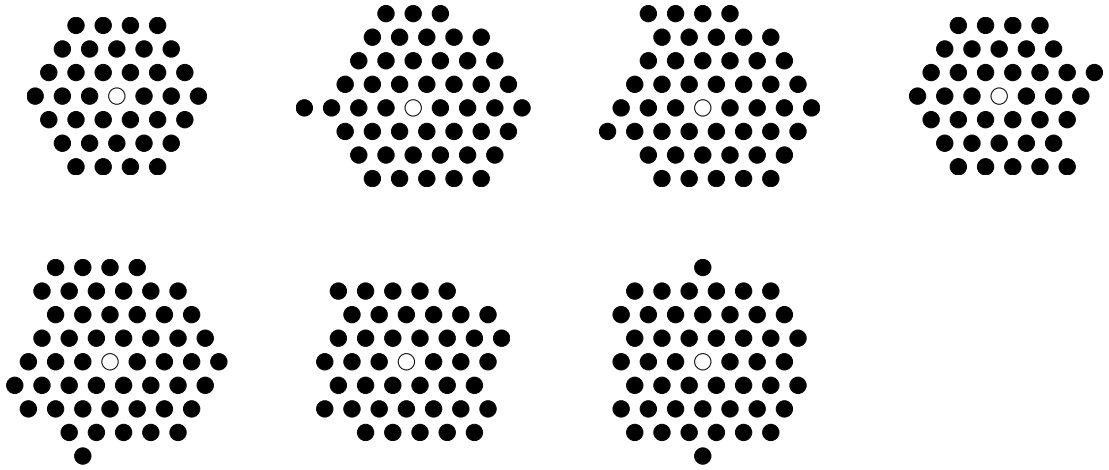


Figure 4.8: The 7 tiles of the 3-quasi-perfect codes over $\mathbb{Z}[\omega]$.

$q \backslash p$	0	1	2
0	-3	-1	1
1	-6	-4	-2

Then, considering the different cases and taking into account that $x \geq y \geq 0$, the only possibility is $x = y + 1$ for $p = 2, q = 0$. Note that this value gives the perfect code.

Now, for the second case, it follows that $x = y + 3 + 2p - 3q$, which gives the following values:

$q \backslash p$	0	1	2
0	3	5	7
1	0	2	4

Then, considering the different cases and taking into account that $t = \left\lfloor \frac{x+y-1}{2} \right\rfloor$ the remaining of the given divisors are obtained. □

The distance distributions of the 7 quasi-perfect codes obtained in Theorem 4.2.21 are shown in Figure 4.8 for error correction capacity $t = 3$. Note that the tile situated on the left upper corner corresponds with the perfect code and that vertex zero is highlighted in white.

Example 4.2.22. *Let us consider $t = 2$ and $\alpha = ((t+1) + t\omega)((t+3) + (t-1)\omega) = (3+2\omega)(5+\omega) = 13+15\omega$. This EJ-integer generates a hexagonal signal constellation with $\mathcal{N}(\alpha) = 13^2 + 15^2 + 13 \cdot 15 = 589$ points. Now, if the code is defined using the first divisor of α, that is, $\mathcal{C}_1 = (3 + 2\omega)$, then this ideal constitutes a 2-perfect code with 31 codewords. On the other hand, if the code is defined using the other divisor, that is $\mathcal{C}_2 = (5 + \omega)$, then the ideal is in this case a 2-quasi-perfect code with 19 codewords. Both codes correct all the error patterns for $t = 2$, but the latter code also corrects error patterns for $t + 1 = 3$.*

Remark 4.2.23. *Again, note that the uniqueness of such codes is conditioned by the restriction of being ideals. Therefore, codes that are neither ideals nor groups of the EJ-integers can be easily constructed as it was done for the Gaussian integers case.*

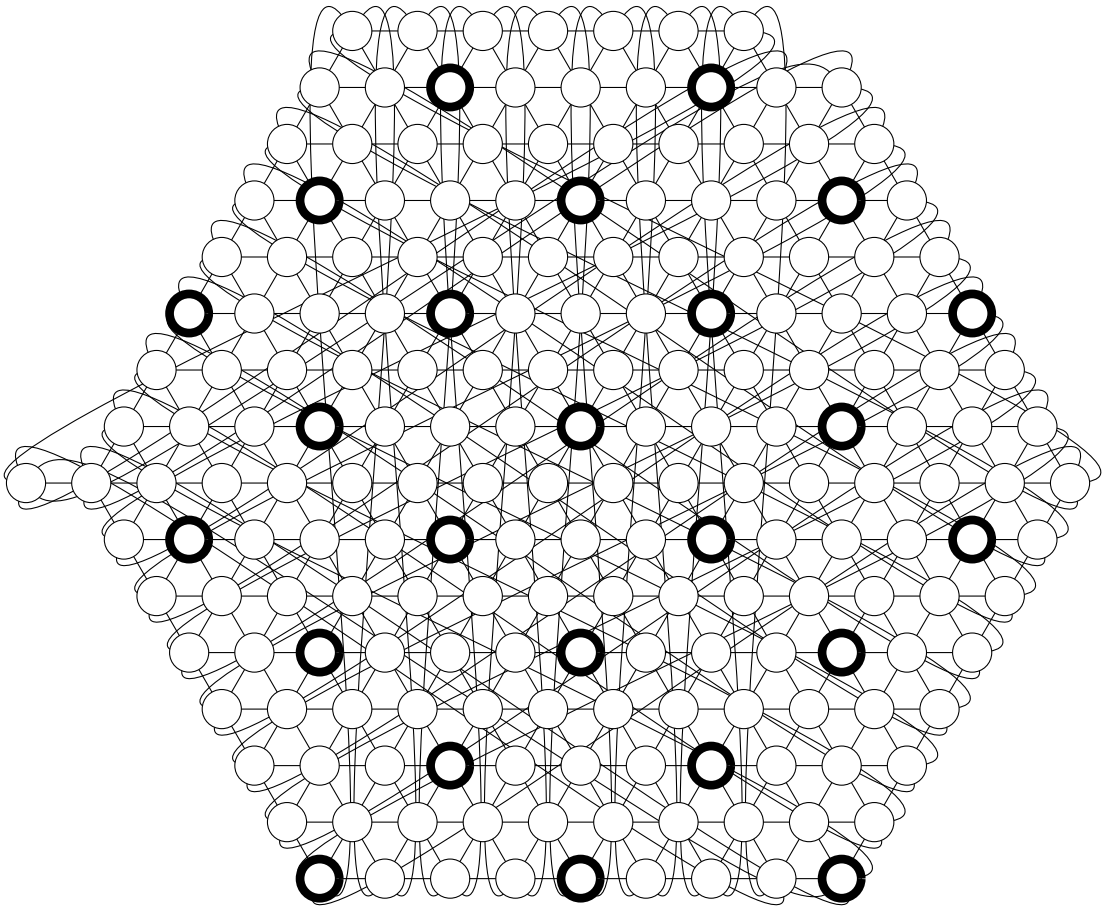


Figure 4.9: Group quasi-perfect code $\mathcal{C} = \langle 1 + 2\omega \rangle$ over $G_{8+8\omega}$.

Example 4.2.24. *In Figure 4.9 the code generated as a group* $\mathcal{C} = \langle 1+2\omega \rangle$ *over the graph* $G_{8+8\omega}$ *is represented. As it can be checked, this code is not an ideal since* $1 + 2\omega \in \mathcal{C}$ *but* $(1 + 2\omega)\omega = -2 + 3\omega \notin \mathcal{C}$.

4.2.5 2-Dimensional Quasi-Perfect Codes for the Lee Metric

Quasi-perfect Lee distance codes over $\mathbb{Z}_K^2$ were considered in [AB03b]. Given positive integers K and t, the proposed code $\mathcal{C}_t$ in $\mathbb{Z}_K^2$ is the one generated by the matrix $G = [t, t+1]$, that is, the code is defined as a group over $\mathbb{Z}_K^2$ as follows:

$$\mathcal{C}_t = \langle t, t+1 \rangle = \{(st \pmod K, s(t+1) \pmod K) \mid 0 \leq s < K\}.$$

In [AB03b] the authors established the conditions over K such that $\mathcal{C}_t$ is a t-quasi-perfect code in $\mathbb{Z}_K^2$. Moreover, two different decoding schemes are provided in the same paper. Later, in [HA06] an optimized decoding algorithm is presented for the same family of codes. In this subsection this construction will be referred as the *quasi-perfect group code construction*.

As it was proved in [MMB06], certain perfect Lee codes over two dimensional spaces can be obtained as subcases of perfect codes being ideals over the Gaussian integers. The main idea under this result is that $\mathbb{Z}_K^2$ and $\mathbb{Z}[i]_K$ are isomorphic as groups such that

the two corresponding metrics coincide. That is, the Lee distance and the one induced by the Gaussian graph are the same metric, since the underlying graphs are isomorphic. Therefore, Theorem 4.2.16 can be applied also to obtain quasi-perfect Lee codes over $\mathbb{Z}_K^2$. Moreover, it makes sense to consider the relationship between the two families of codes, that is, the one given by the quasi-perfect group codes construction and the one presented in this section.

As a first approach to the determination of the connection between the two types of codes, note that the codes constructed using Gaussian integers are indeed ideals over the Gaussian integer rings, while the codes defined by AlBdaiwi and Bose in [AB03b] are just group codes over the Gaussian integers. Therefore, in the general case, both codes do not coincide. As an example, let us consider the Lee space $\mathbb{Z}_{29}^2$. Clearly, this space can be seen as $\mathbb{Z}[i]_{29}$ with the Gaussian graph's metric. In Example 4.2.25, two different constructions of quasi-perfect Lee codes over this space are considered, the first one being a group and the second one being an ideal, both over the same Gaussian integer ring.

Example 4.2.25. *Let us consider $\mathbb{Z}_{29}^2 \cong \mathbb{Z}[i]_{29}$. Expressed in the notation to Gaussian integers, in [HA06] it was shown that the code given by the group $\mathcal{C}_1 = \langle 3+4i \rangle \subset \mathbb{Z}[i]_{29}$ is a 3-quasi-perfect code, as shown in Figure 4.10. Note that the code has 29 codewords.*

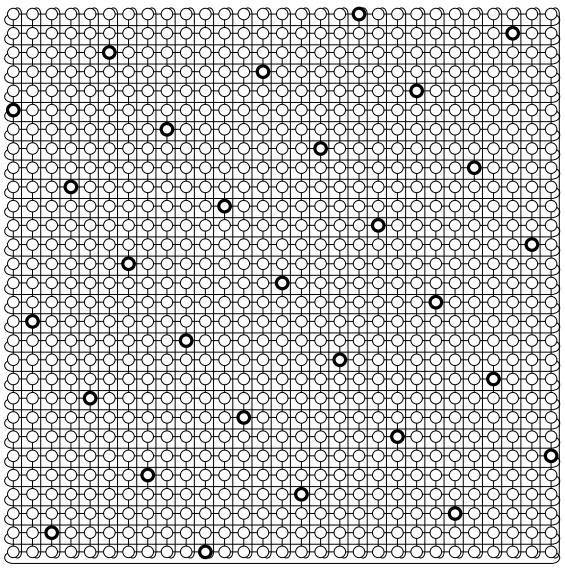


Figure 4.10: A quasi-perfect code over $\mathbb{Z}_{29}[i]$ being a group but not an ideal

On the other hand the ideal $\mathcal{C}_2 = (2+5i)$ over $\mathbb{Z}[i]_{29}$ is by Theorem 4.2.16 again a 3-quasi-perfect code with 29 codewords, since $\mathcal{N}(2-5i) = \mathcal{N}(\frac{29}{2+5i}) = 29$.

Now, if the distance properties of the codes are considered, it can be seen that both codes have maximum distance 25 and average distance 15. However, the codes are different and let us illustrate it. In the first group code $\mathcal{C}_1$, for every codeword there exist two codewords at distance 7 and two codewords at distance 8 from it. On the other hand, in the code being an ideal $\mathcal{C}_2$, for every codeword there are exactly 4 codewords at distance 7. The next codewords are obtained at distance 10, as it can be seen in Figure 4.11.

What comes out from the previous example is that both constructions are different in general. Moreover, the question of when both the construction of quasi-perfect group codes and the one presented in this section coincide is directly connected with the study of

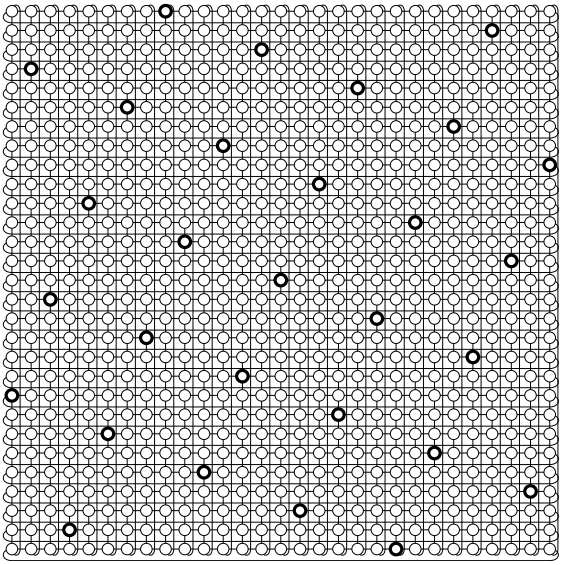


Figure 4.11: A quasi-perfect code over $\mathbb{Z}_{29}[i]$ being an ideal

the situation in which both algebraic structures match up, that is, whenever an additive group of the Gaussian integers is also an ideal over the same ring. In this direction, it is proved the following lemma, which characterizes these situations. Let $\gcd_{\mathbb{Z}[i]}$ denote the greatest common divisor over the Gaussian integers and $\gcd_{\mathbb{Z}}$ the greatest common divisor over the ring of integers.

Lemma 4.2.26. *Let $\alpha, \beta \in \mathbb{Z}[i]$ and consider $\langle\beta\rangle = \{n\beta \mid n \in \mathbb{Z}\} \subseteq \mathbb{Z}[i]_\alpha$. Let $\frac{\alpha}{\gcd_{\mathbb{Z}[i]}(\alpha,\beta)} = a + bi$. Then, $i\beta \in \langle\beta\rangle$ if and only if $\gcd_{\mathbb{Z}}(a, b) = 1$.*

Proof. Let us denote $\delta = \gcd_{\mathbb{Z}[i]}(\alpha, \beta)$, $\alpha' = \frac{\alpha}{\delta} = a + bi$ and $\beta' = \frac{\beta}{\delta}$.

First, it will be proved that if $i\beta \in \langle\beta\rangle$ then $\gcd_{\mathbb{Z}}(a, b) = 1$. Since $i\beta \in \langle\beta\rangle$ it follows that $i\beta \equiv n\beta \pmod{\alpha}$ for some integer n. Thus, there exists $\gamma \in \mathbb{Z}[i]$ such that $\beta(i - n) = \alpha\gamma$ and $\beta'(i - n) = \alpha'\gamma$ with $\gcd_{\mathbb{Z}[i]}(\alpha', \beta') = 1$. Since α' divides $\beta'(i - n)$ and α' and β' are coprimes, it follows that α' divides $i - n$. This entails that $\mathcal{N}(\alpha')$ divides

$$(i - n)\bar{\alpha}' = (i - n)(a - bi) = (-na + b) + (a + nb)i.$$

As a consequence the following Diophantine system is obtained:

$$\begin{aligned} -na + b &= \mathcal{N}(\alpha')p \\ a + nb &= \mathcal{N}(\alpha')q \end{aligned}$$

for integers p, q. Now,

$$\begin{aligned} \mathcal{N}(\alpha') = a^2 + b^2 &= a(\mathcal{N}(\alpha')q - nb) + b(\mathcal{N}(\alpha')p + na) \\ &= aN(\alpha')q - anb + bN(\alpha')p + bna = \mathcal{N}(\alpha')(aq + bp). \end{aligned}$$

By simplification it is obtained that $1 = aq + bp$, which implies $\gcd_{\mathbb{Z}}(a, b) = 1$.

To prove the converse, let us assume $\gcd_{\mathbb{Z}}(a, b) = 1$, that is, there exist integers p, q such that $aq + bp = 1$. Let γ be $(p + qi)\beta'$. Then, $\alpha\gamma = \alpha'\delta(p + qi)\beta' = \beta(p + qi)(a + bi) = \beta\big((pa - qb) + (qa + pb)i\big)$. As $qa + pb = 1$ and $n = qb - pa \in \mathbb{Z}$ it follows that $\alpha\gamma = \beta(-n + i)$. Hence $\beta i \equiv \beta n \pmod{\alpha}$ and $\beta i \in \langle\beta\rangle$, which concludes the proof. □

As a consequence of the previous lemma, it is expected that both constructions, although different in the general case, yield the same codes in certain cases. In the following example, such a situation is considered, that is, a quasi-perfect Lee group code is given that results in an ideal code over the Gaussian integers. However, as it can be seen in that example, the same quasi-perfect code cannot be obtained from both constructions separately.

Example 4.2.27. *Let us consider* $N = 89^2 = (5+8i)(5-8i)$ *and the ring* $\mathbb{Z}[i]_{89}$. *In this ring a quasi-perfect code that is a group and also an ideal is being constructed. Therefore, the example illustrates that the previously known construction given in [AB03b] and the one presented in this section are not completely disjoint in the quasi-perfect case. Hence, let us consider* $\beta = 27+28i = (-1+4i)(5-8i)$ *the generator of both codes. Note that* $\gcd_{\mathbb{Z}[i]}(N,\beta) = 5-8i$, *with* $\frac{N}{5-8i} = 5+8i$ *and* $\gcd(5,8) = 1$, *fulfilling the hypothesis of the previous Lemma 4.2.26. Now, it is enough to realize that*

$$\langle 27+28i\rangle = (27+28i) = (5-8i),$$

where $5-8i = (6-7i)+(-1-i)$, *which gives a 6-quasi-perfect ideal code using Theorem 4.2.16, with* $t = 6$ *and* $h = 1$.

Remark 4.2.28. *Note that in the previous example, the group code is generated by an element of the form* $t+(t+1)i$ *but its correction capacity is not equal to* t. *Moreover, it can be straightforwardly obtained that the only group codes from the construction in [AB03b] that coincide with the ideal quasi-perfect codes considered in this section are in fact the perfect codes generated by* $\beta = t+(t+1)i$. *As a consequence, the generators given in Theorem 4.2.16 provides many new examples of quasi-perfect Lee codes over two-dimensional spaces.*

4.2.6 Decoding Algorithms

In this subsection decoding algorithms for the quasi-perfect codes over Gaussian and EJ-integers obtained in this section are presented. The algorithms take advantage of the algebraic structure of the codes. Hence, the procedures use the fact that the codes form ideals over the corresponding integer ring to perform the decoding process.

Decoding algorithms for the Lee-distance quasi-perfect codes were presented in [AB03b] and [HA06]. AlBdaiwi and Bose's algorithm in [AB03b] makes a strong use of the cyclic nature of the group codes that they consider. They construct a subset of the code with cardinality $2t+1$ and correct by the closest codeword. Therefore, the algorithm can be straightforwardly adapted to decode our codes when these codes are additive cyclic groups over the integer rings, as it has been considered in Lemma 4.2.26. A more efficient algorithm also for decoding quasi-perfect Lee codes was presented by Horak and AlBdaiwi in [HA06]. In this case, the received symbol is corrected by the closest codeword among at most 4 codewords. The algorithms that are proposed in this section have some geometrical similarities to this last one although they decode different code constructions. Finally, in [ABF12] and [FB10] general algorithms for minimum distance calculations are given, which can be used for decoding in Gaussian and Eisentein–Jacobi lattice constellations. However, in this section a different approach is given, which tries to minimize integer multiplications and divisions.

The decoding procedures, both for the Gaussian and the EJ-integer rings, are presented in Algorithms 6 and 7, respectively. However, both methods follow the same idea. Since

the codes are defined by means of ideals, the codewords are multiples of the generator of the ideal. Hence, given a received word, finding the nearest codeword that corrects it is equivalent to finding the quotient that results from the Euclidean division of the received word by the ideal generator. Moreover, the correctness of the algorithms is guaranteed by Theorem 4.2.36, which is obtained as a consequence of the following lemmas.

Notation 4.2.29. *As it can be seen in the algorithms' description, $|\beta|$ will denote the common Manhattan weight for a given Gaussian or EJ-integer. Let us also denote by $[\cdot]$ the rounding operator, with $[a+b\rho]=[a]+[b]\rho$. Then, the quotient and the remainder will be denoted as* $\operatorname{quot}(\alpha,\beta)=\left[\frac{\alpha\bar{\beta}}{\mathcal{N}(\beta)}\right]$ *and* $\operatorname{rem}(\alpha,\beta)=\alpha-\operatorname{quot}(\alpha,\beta)\beta$. *It can be checked that $\alpha=\beta\operatorname{quot}(\alpha,\beta)+\operatorname{rem}(\alpha,\beta)$ with $|\operatorname{rem}(\alpha,\beta)|<\mathcal{N}(\beta)$, which provides a Euclidean division algorithm for $\mathbb{Z}[\rho]$, $\rho=i,\omega$.*

Lemma 4.2.30. *Let $\alpha_1,\alpha_2,\beta\in\mathbb{Z}[\rho]$. If $\alpha_1\equiv\alpha_2 \pmod{\beta}$ then $\operatorname{rem}(\alpha_1,\beta)=\operatorname{rem}(\alpha_2,\beta)$.*

Proof. Let $\alpha_2=\alpha_1+\beta\mu$, $\mu\in\mathbb{Z}[\rho]$. Then, it follows that $\operatorname{rem}(\alpha_2,\beta)=\alpha_2-\left[\frac{\alpha_2\bar{\beta}}{\mathcal{N}(\beta)}\right]\beta=\alpha_2-\left[\frac{\alpha_1\bar{\beta}+\beta\mu\bar{\beta}}{\mathcal{N}(\beta)}\right]\beta=\alpha_2-\left[\frac{\alpha_1\bar{\beta}}{\mathcal{N}(\beta)}+\mu\right]\beta=\alpha_2-\left[\frac{\alpha_1\bar{\beta}}{\mathcal{N}(\beta)}\right]\beta-\mu\beta=\alpha_2-\operatorname{quot}(\alpha_1,\beta)\beta-\mu\beta=\alpha_1+\bar{\beta}\mu-\operatorname{quot}(\alpha_1,\beta)\beta-\mu\beta=\alpha_1-\operatorname{quot}(\alpha_1,\beta)\beta=\operatorname{rem}(\alpha_1,\beta)$. □

Algorithm 6: Decoding in $\mathbb{Z}[i]_\alpha$

Data: $\alpha\in\mathbb{Z}[i]$ being the $\mathbb{Z}[i]_\alpha$ generator
$\beta\in\mathbb{Z}[i]$ being the code generator
$t\in\mathbb{Z}^+$ being the code correction capacity
$\gamma\in\mathbb{Z}[i]_\alpha$ being the received symbol
Result: $\theta\in\mathbb{Z}[i]_\alpha$ being the corrected symbol
Compute $q:=\operatorname{quot}(\gamma,\beta)$, $r:=\operatorname{rem}(\gamma,\beta)$;
if $\beta=(t+1)+(t+1)i$ *or* $|r|\leq t$ **then**
  Return $\theta=q\beta$;
else
  Compute:
    $Q=\{(q+h)\beta \pmod{\alpha} \mid h\in\{0,\pm1\pm i\}\}$;
  Find θ such that $|\theta|=\min\{D_\alpha(x,\gamma)\mid x\in Q\}$;
  Return θ ;
end

The next result considers the geometrical location of the remainders obtained by such a division algorithm.

Lemma 4.2.31. *Let $\beta\in\mathbb{Z}[\rho]$. The set*

$$R_\beta=\{\operatorname{rem}(\delta,\beta)\mid\delta\in\mathbb{Z}[\rho]\},$$

is obtained as the integral points of the complex parallelepiped with vertices at $\{\pm\mu,\pm\eta\}$, where $2\mu=\beta(1+\rho)$ and $2\eta=\beta(1-\rho)$.

Proof. Let us define $\mathfrak{a}(a+b\rho)=a$ and $\mathfrak{b}(a+b\rho)=b$. By Lemma 4.2.30 it is obtained that $R_\beta=\{\operatorname{rem}(\delta,\beta)\mid\delta\in\mathbb{Z}[\rho]\}=\{\delta\in\mathbb{Z}[\rho]\mid\operatorname{quot}(\delta,\beta)=0\}$. Hence, $R_\beta=\{\delta\mid|2\mathfrak{a}(\delta\bar{\beta})|\leq\mathcal{N}(\beta)$ and $|2\mathfrak{b}(\delta\bar{\beta})|\leq\mathcal{N}(\beta)\}$. Since $2\mathfrak{a}(\delta\bar{\beta})$ and $2\mathfrak{b}(\delta\bar{\beta})$ are both linear functions it follows that R_β is a convex polyhedron.

Algorithm 7: Decoding in $\mathbb{Z}[\omega]_\alpha$

Data: $\alpha \in \mathbb{Z}[\omega]$ being the $\mathbb{Z}[\omega]_\alpha$ generator
$\beta \in \mathbb{Z}[\omega]$ being the code generator
$t \in \mathbb{Z}^+$ being the code correction capacity
$\delta \in \mathbb{Z}[\omega]_\alpha$ being the received symbol
Result: $\theta \in \mathbb{Z}[\omega]_\alpha$ being the corrected symbol
Compute $q := \text{quot}(\gamma, \beta)$, $r := \text{rem}(\gamma, \beta)$;
if $|r| \leq t$ **then**
  Return $\theta = q\beta$;
else
  Compute:
    $Q = \{(q + h)\beta \pmod{\alpha} \mid h \in \{0, \pm 1 \pm \omega\}\}$;
  Find θ such that $|\theta| = \min\{D_\alpha(x, \gamma) \mid x \in Q\}$;
  Return θ ;
end

One pair of vertices of R_β is $\pm\mu$ with $2\mathfrak{a}(\mu\bar{\beta}) = 2\mathfrak{b}(\mu\bar{\beta}) = \mathcal{N}(\beta)$. Thus, $2\mu\bar{\beta} = 2\mathfrak{a}(\mu\bar{\beta}) + 2\mathfrak{b}(\mu\bar{\beta})\rho = \beta\bar{\beta}(1 + \rho)$, from which it is obtained that $2\mu = \beta(1 + \rho)$.

The other pair of vertices of R_β is $\pm\eta$ with $2\mathfrak{a}(\eta\bar{\beta}) = -2\mathfrak{b}(\eta\bar{\beta}) = \mathcal{N}(\beta)$. Thus, $2\eta\bar{\beta} = 2\mathfrak{a}(\eta\bar{\beta}) + 2\mathfrak{b}(\eta\bar{\beta})\rho = \beta\bar{\beta}(1 - \rho)$, from which it is obtained that $2\eta = \beta(1 - \rho)$. □

As a consequence, the remainders generated by β are located in a parallelepiped R_β. A translation of this parallelepiped was considered in [FB10] for defining a set of representatives of the quotient group. Moreover, the weight of the remainder obtained over the Gaussian integers is bounded by the diameter of the Gaussian graph generated by the divisor. However, this remainder not always minimizes the weight as it is expected in order to perform correction. The next lemma proves these facts.

Lemma 4.2.32. *Let $\beta = c + di \in \mathbb{Z}[i]$.*

$$|\,\text{rem}(\delta, \beta)| \leq \begin{cases} \max\{|c|, |d|\} & \text{if } c \equiv d \pmod 2 \\ \max\{|c|, |d|\} - 1 & \text{if } c \not\equiv d \pmod 2 \end{cases}$$

Proof. Let μ and R_β be as in Lemma 4.2.31. Since $\rho = i$, the vertices of R_β are $\{\mu i^k \mid k \in \mathbb{Z}\}$. Now, as $|\cdot|$ is a linear function, it has to be maximized in a vertex. Thus, $|\,\text{rem}(\delta, \beta)| \leq |\mu i^k| = |\mu| = |\frac{c-d}{2} + \frac{c+d}{2}i| = \max\{|c|, |d|\}$. If 2 does not divide $c + d$, $\text{rem}(\delta, \beta)$ cannot be $\mu \notin \mathbb{Z}[i]$ and as a consequence the strict inequality $|\,\text{rem}(\delta, \beta)| < |\mu|$ is obtained. □

Since the remainder considered in the previous lemma not always minimizes the weight function, slight modifications might be done in order to find the one with minimum weight, as it is shown in the following result.

Lemma 4.2.33. *Let (β) be a t-quasi-perfect code over $\mathbb{Z}[i]_\alpha$ such that $\mathcal{N}(\beta)$ is odd. Then, for every $\delta \in \mathbb{Z}[i]_\alpha$, there exists $\sigma \in \{0, \pm 1, \pm i\}$ such that* $\text{rem}(\delta, \beta) = \arg\min\{|\gamma| \mid \gamma \equiv \delta \pmod \beta\} + \sigma\beta$.

Proof. Let c be the closest codeword to $\delta' = \text{rem}(\delta, \beta)$. If $c = 0$ then let $\sigma = 0$. Otherwise, as c is the closest codeword it follows that $D_\alpha(\delta', c) \leq t + 1$. By Lemma 4.2.32 and the

fact that $2 \nmid \mathcal{N}(\beta)$ necessarily $D_\alpha(\delta', 0) \leq (t+1+h)-1 \leq t+1$. Now, by the triangular inequality $D_\alpha(0,c) \leq D_\alpha(0,\delta') + D_\alpha(\delta', c) \leq 2t+2$. As $c \neq 0$ and (β) is a t-quasi-perfect code, then $2t+1 \leq D_\alpha(c,0) \leq 2t+2$. Finally, $c \in \{\pm\beta, \pm\beta i\}$. □

Lemma 4.2.34. *Let (β) be a t-quasi-perfect code over $\mathbb{Z}[i]_\alpha$ such that $\mathcal{N}(\beta)$ is even. Then, for every $\delta \in \mathbb{Z}[i]_\alpha$, $\operatorname{rem}(\delta, \beta) = \arg\min\{|\gamma| \mid \gamma \equiv \delta \pmod{\beta}\}$.*

Proof. By uniqueness, it can be assumed that $\beta = (t+1) + (t+1)i$. For any $\delta' \in \mathbb{Z}[i]$ let $\delta = \arg\min\{|\gamma| \mid \gamma \equiv \delta' \pmod{\beta}\} = a + bi$. If $|\delta| = t+1$, then by Lemma 4.2.32 it follows that $|\operatorname{rem}(\delta, \beta)| = t+1$ with minimum weight. Otherwise, $|\delta| \leq t$. Then,

$$\operatorname{quot}(\delta, \beta) = \left[\frac{(a+b)(t+1)}{2t^2+2t+2}\right] + \left[\frac{(b-a)(t+1)}{2t^2+2t+2}\right] i.$$

As a consequence, $\left[\frac{|(a+b)(t+1)|}{2t^2+2t+2}\right] \leq \left[\frac{|t(t+1)|}{2t^2+2t+2}\right] \leq \left[\frac{1}{2}\right] = 0$ and $\left[\frac{|(b-a)(t+1)|}{2t^2+2t+2}\right] \leq \left[\frac{|t(t+1)|}{2t^2+2t+2}\right] \leq \left[\frac{1}{2}\right] = 0$. Hence $\operatorname{quot}(\delta, \beta) = 0$ and $\operatorname{rem}(\delta, \beta) = \operatorname{rem}(\delta', \beta) = \delta$. □

Although in the Eisenstein–Jacobi integers case the corresponding remainder may have a larger weight, the operations to get the minimum one are similar, as it is shown in the next result.

Lemma 4.2.35. *Let $\beta \in \mathbb{Z}[\omega]$ and $\mathcal{C} = (\beta)$ be a t-quasi-perfect code over $\mathbb{Z}[\omega]_\alpha$. For every δ, there exists $\sigma \in \{0, \pm 1, \pm\omega\}$ such that $\operatorname{rem}(\delta, \beta) = \arg\min\{|\gamma| \mid \gamma \equiv \delta \pmod{\beta}\} + \sigma\beta$.*

Proof. Let us consider R_β, μ and η as in Theorem 4.2.31. It is clear that the result only has to be proved for the boundary of R_β. Then, $\mu = \frac{\beta(1+\omega)}{2}$ is the middle point between β and $\beta\omega$ and $\eta = \frac{\beta(1-\omega)}{2} = -\frac{\omega^2}{2}$ is the middle point between 0 and $-\beta\omega^2$.

Let us prove first that all the points in the segment from μ to η can be corrected by 0 or $\beta\omega$. In this direction, note that μ can be corrected by $\beta\omega$. Since the other points in the segment are farther away from β, no point in the segment is corrected by β. Similarly, η can be corrected by 0 and the other points are farther away from $-\beta\omega^2$.

By an analogous reasoning regarding the other segments it follows that the only codewords that correct points in R_β are $\{0, \pm\beta, \pm\beta\omega\}$. □

Theorem 4.2.36. *Algorithms 6 and 7 are correct.*

Proof. Lemmas 4.2.33 and 4.2.34 guarantee the correctness for $\beta \in \mathbb{Z}[i]$ with odd and even norms, respectively. Lemma 4.2.35 guarantees the correctness of the algorithm over $\mathbb{Z}[\omega]$. □

Remark 4.2.37. *Although the algorithms have similar appearance for Gaussian and Eisenstein integers, the case where $|r| > t$ over the Gaussian integers is an exceptional case (indeed $|r|$ is at most $t+1$). However, in the Eisenstein–Jacobi case this is a usual situation. Moreover, it can be computed that $|r| \leq 3t/2 + 3$.*

Finally, the next example shows the performance of Algorithm 7 over a particular scenario.

Example 4.2.38. *Let us consider $\alpha = 14 + 9\omega \in \mathbb{Z}[\omega]$ and $\beta = 5 + \omega \in \mathbb{Z}[\omega]$. Since $\alpha = (3+\omega)\beta$ and $\beta = (t+3) + (t-1)\omega$ for $t = 2$ then $\mathcal{C} = (5+\omega)$ is a t-quasi-perfect code over $\mathbb{Z}[\omega]_\alpha$ for error correction capacity $t = 2$. Now, let us assume that the symbol*

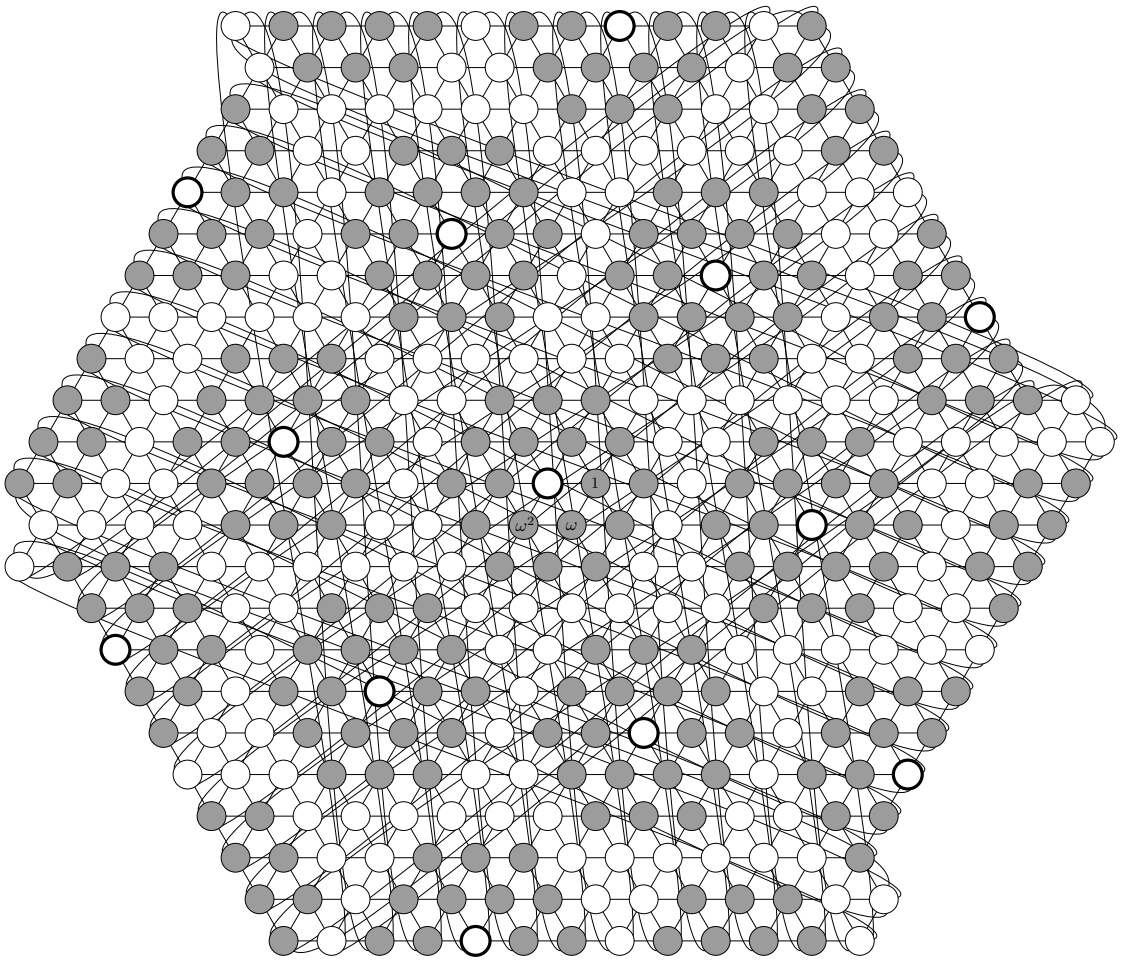


Figure 4.12: A 2-quasi-perfect code over $\mathbb{Z}_{14+9\omega}[\omega]$

$\gamma = 12 - 3\omega$ has been received. First, γ is not a codeword. Then, let us correct γ using Algorithm 7. Applying Euclidean division it is obtained that

$$\gamma = (2 - \omega)\beta + (1 + \omega).$$

Since $|r| = t$ then it can be corrected by $\theta = q\beta = 11 - 4\omega$.

Now, if the received symbol is $\gamma' = 6 + 8\omega$, it is possible to proceed in a similar way. Applying Algorithm 7 results that

$$\gamma' = (1 + \omega)\beta + (2 + \omega),$$

with $|r| = 2 + 1 = t + 1$. The set $Q = \{4 + 7\omega, 9 + 8\omega, -1 + 6\omega, 3 + 13\omega, 5 + \omega\}$. Note that

$$D_\alpha(4 + 7\omega, \gamma) = 3,$$

$$D_\alpha(9 + 8\omega, \gamma) = 3.$$

So both $\theta' = 4 + 7\omega$ and $\theta'' = 9 + 8\omega$ correct the obtained symbol. In Figure 4.12 a graphical representation of the constellation $\mathbb{Z}[\omega]_\alpha$ is shown.

4.2.7 Conclusions

QAM-type and hexagonal signal constellation have been previously modeled by means of quotients of Gaussian and EJ-integers, [Hub94], [Hub93]. Cayley graphs over these rings were proposed in [MBG07] and [MSBG08] to define the so called Gaussian and EJ metrics over these spaces. Moreover, the problem of perfect codes over these quotient rings has been previously considered and such perfect codes were built as ideals over the rings generated by element with maximal norm in the ring [MBGG05].

In this section quasi-perfect codes over Gaussian and EJ-graphs have been considered. Constructive methods for quasi-perfect codes being ideals have been given and the uniqueness of the codes, under the hypothesis of being ideals, has been proved. As a consequence, previously known perfect codes are shown to be the unique ones being ideals over these graphs. Moreover, decoding algorithms for the quasi-perfect codes over Gaussian and EJ-integers have been presented, which also decode the previously known perfect codes.

The relationship between perfect codes over Gaussian graphs and the perfect codes for the two dimensional Lee space was considered in [MMB06]. As it was shown, some quotient rings of the Gaussian integers and the two dimensional Lee space coincide. Thus, the quasi-perfect codes construction given in this section can be also applied to generate new quasi-perfect codes over Lee spaces. Moreover, the connections between quasi-perfect codes and the previously known for the Lee metric [AB03b] have been investigated. It has been shown that both constructions are different in the general case by the characterization of the conditions under which both constructions collapse.

Finally, it can be guessed that the procedures used in this section may be extended to other rings, resulting in the construction of new quasi-perfect codes associated with different signal constellations.

4.3 Quasi-Perfect Lee Codes of Radius 2 and Arbitrarily Large Dimension

A construction of 2-quasi-perfect Lee codes is given over the space $\mathbb{Z}_p^n$ for p prime, $p \equiv \pm 5 \pmod{12}$ and $n = 2[\frac{p}{4}]$. It is known that there are infinitely many such primes. Perfect codes for the Lee-metric were conjectured by Golomb and Welch not to exist, which has been proved for large radii and also for low dimension. The codes found in this thesis are very close to be perfect, which tells about the nature of the conjecture. Some computations show that the related lattice graphs are Ramanujan, which could provide further connections between the fields of coding theory and optimal graph theory.

4.3.1 Introduction

Golomb and Welch conjectured in their seminal paper [GW70] that perfect Lee codes do only exist for spheres of radius $r = 1$ or in Lee spaces of dimension $n = 1, 2$. A constructive result for 1-perfect Lee codes was also given in that paper. Moreover, for a radius sufficiently greater than the space dimension, a negative existence result was given by approximating the problem to the densest tilling of $\mathbb{R}^n$ with cross-polytopes. Afterwards, Molnár enumerated all lattice-like 1-perfect codes in [Mol71]. Later, Post in [Pos75] gave a strong negative result. If a perfect code exists, Post determined an upper bound for its radius, in terms of the dimension, specifically, the radius must fulfill $r < \frac{1}{2}n\sqrt{2} - \frac{3}{4}\sqrt{2} - \frac{1}{2}$ for $n \geq 6$. Later, J. Astola [Ast82] and Lepistö [Lep81] improved

the bound given by Post to a quadratic relation between r and n, which can be considered as an Elias-type bound for Lee codes. Those negative results in [Pos75, Ast82, Lep81], suggest that the conjecture is more difficult for radius 2, as argued by Horak in [Hor09a].

Other authors have considered the conjecture for small dimensions. For example, Gravier *et al.* in [GMP98] proved the non-existence of perfect codes in 3-dimensional Lee spaces, even considering spheres of different radii. Dimension 4 was considered by Špacapan in [Špa07], again with the possibility of spheres of different radii but at least 2. Also, Horak in [Hor09b] and [Hor09a] proved the non-existence of perfect Lee codes in spaces of dimension up to $n \leq 6$. Later, to achieve higher dimensions, Horak and Grošek in [HG14] restricted the problem to linear codes and verified computationally the non-existence of perfect Lee codes for dimension $n \leq 12$ and radius $r = 2$. The status of the conjecture for low values of r and n is depicted in Figure 4.13.

On the other hand, several papers have considered problems around the conjecture that could give some insight. One approach has been to generalize the Lee metric. Huber in [Hub94] gave 1-perfect codes over Gaussian integers and some non-perfect codes with greater correction. In [CMAPJ04] Costa *et al.* considered a relation between tessellations, graphs and codes over flat tori. In [MBG07, MSB$^+$08, MBG09] Martinez *et al.* gave a generalization of the Lee distance by means of a family of Cayley graphs over Cayley–Dickson algebras. Also, the existence of perfect codes being ideals of the algebras was considered. Nishimura and Hiramatsu in [NH08] generalized the Lee distance using a surjective function from $\mathbb{Z}^l$ into a finite field and constructed some non-perfect 2-error correcting codes for this metric.

The existence of Lee codes has also been considered in terms of the size q of the alphabet. AlBdaiwi *et al.* in [AHM09] enumerated all the alphabet sizes q such that there exists a linear 1-perfect Lee code over $\mathbb{Z}^n$. In [AT13] H. Astola and Tabus obtained, for small alphabet size q and dimension n, an upper bound of the number of codewords of error correcting Lee codes.

Recently, a new approach has been taken in terms of *diameter perfect codes*, which were introduced by Ahlswede *et al.* in [AAK01]. A subset $\mathcal{C} \subseteq \mathbb{Z}_q^n$ is a diameter perfect code if there exists an anticode $\mathcal{A}$ such that $|\mathcal{C}||\mathcal{A}| = q^n$. This concept generalizes perfect codes since diameter perfect codes with minimum distance being odd are in fact the perfect codes. Etzion in [Etz11] built diameter perfect codes of minimum distance 4. Later, Horak and AlBdaiwi [HA12] enumerated the arities q such that there are 4-diameter perfect codes over $\mathbb{Z}_q$. Araujo *et al.* in [ADH14] presented a generalization of diameter perfect Lee codes, together with a new conjecture that extends the conjecture by Golomb and Welch. Etzion *et al.* in [EVY13] built Lee codes for large dimension by means of weighing matrices[2].

In the present section a explicit construction of linear quasi-perfect Lee codes of radius 2 for arbitrarily large dimension is given. As it will be shown, these codes are very close to be perfect since they have half of the density of potential perfect codes. In the authors opinion, the existence of these quasi-perfect codes, hints that maybe a perfect code could exist for low radius; and if they do not exist then the proof must be of a very different nature than the proofs in previous papers dealing with the conjecture.

These quasi-perfect 2-error correcting Lee codes will be defined by means of Cayley graphs over Abelian finite groups. The degree of the graph will be the double of the dimension of the Lee space. The order of the graph will be in inverse relation to the density of the quasi-perfect code. Thus, the main contribution of the section is presented

[2]A matrix W is a weighing matrix of weight w if its entries belong to $\{0, \pm 1\}$ and $WW^t = wI$.

n \ r	1	2	3	4	5	6	7	8	9	10
1	t	t	t	t	t	t	t	t	t	t
2	G	G	G	G	G	G	G	G	G	G
3	G	G, Gr*	P, Gr*	P, Gr*	P, Gr*	P, Gr*	P, Gr*	P, Gr*	P, Gr*	P, Gr*
4	G	P, S_c^*	P, S_c^*	P, S_c^*	P, S_c^*	P, S_c^*	P, S_c^*	P, S_c^*	P, S_c^*	P, S_c^*
5	G	H0	P	P	P	P	P	P	P	P
6	G	H1	P	P	P	P	P	P	P	P
7	G	$H2_c^l$	P	P	P	P	P	P	P	P
8	G	$H2_c^l$		P	P	P	P	P	P	P
9	G	$H2_c^l$			P	P	P	P	P	P
10	G	$H2_c^l$			P	P	P	P	P	P
11	G	$H2_c^l$				P	P	P	P	P
12	G	$H2_c^l$				P	P	P	P	P
13	G						P	P	P	P
14	G						P	P	P	P
15	G							P	P	P

A perfect Lee code is known.

It is known that there is no perfect Lee code.

*: Even with different radii
$_c$: Computer based proof
^{l}: Only the linear case is known
t: Trivial
G: Golomb and Welch. 1970. [GW70]
P: Post. 1975. [Pos75]
Astola. 1982. [Ast82] and Lepistö. 1981. [Lep81]. Too small to show.
Gr: Gravier *et al.* 1998. [GMP98]
S: Špacapan. 2007. [Špa07]
H0: Horak. 2009. [Hor09b]
H1: Horak. 2009. [Hor09a]
H2: Horak and Grošek. 2014. [HG14]

Figure 4.13: Cases in which Golomb–Welch conjecture is proved.

in the next result.

Theorem 4.3.1. *For any prime $p \geq 7$ such that $p \equiv \pm 5 \pmod{12}$ there exists a linear 2-quasi-perfect p-ary Lee code over $\mathbb{Z}_p^n$, where $n = 2\left[\frac{p}{4}\right]$ and with p^{n-2} codewords.*

Note that the notation $[a]$ stands for the closest integer to the rational number a. As an example of the codes obtained in previous result, let us consider the following:

Example 4.3.2. *Let $n = 4$, $p = 7$. Then, the code over $\mathbb{Z}_7^4$ defined by the parity-check matrix*

$$\begin{pmatrix} 1 & 0 & 2 & -2 \\ 0 & 1 & 2 & 2 \end{pmatrix}$$

results in a 2-quasi-perfect 7-ary Lee code over $\mathbb{Z}_7^4$. This code has $p^{n-2} = 49$ codewords. It is known that perfect codes do not exist in this case since the sphere packing bound is $\frac{7^4}{41} \approx 58.56$.

As a consequence of Dirichlet's theorem on arithmetic progressions, there are infinitely many primes p such that $p \equiv 5 \pmod{12}$ and infinitely many primes such that $p \equiv -5 \pmod{12}$. Thus, for any constant c, there is a prime $p \equiv \pm 5 \pmod{12}$ such that the dimension $n = 2\left[\frac{p}{4}\right]$ is greater than c. As a consequence of this and Theorem 4.3.1, it is obtained that:

Corollary 4.3.3. *There are infinitely many $n \in \mathbb{N}$ such that there exists a 2-quasi-perfect Lee code over a n-dimensional Lee space.*

As it will be seen later, the result is constructive, and any application that requires the use of Lee-codes can benefit from it. For example, Roth and Siegel in [RS94] considered BCH Lee codes and their application to constrained and partial-response channels. Using space embeddings, Jiang *et al.* in [JSB10] gave a method to construct Charge-Constrained Rank-Modulation codes (CCRM codes) from Lee error-correcting codes, which could be employed for flash memories. H. Astola and Stankovic in [AS12] considered Lee codes to build decision diagrams.

In the rest of the section a family of Cayley graphs over Gaussian integers will be considered. Thus, the family is defined for the additive group of the quotient ring $\mathbb{Z}[i]/p\mathbb{Z}[i]$ as follows.

Definition 4.3.4. *Given an integer prime p, let us define the Cayley graph*

$$\mathcal{G}_p = Cay(\mathbb{Z}[i]/p\mathbb{Z}[i]; H),$$

where

$$H = \{\beta \in \mathbb{Z}[i]/p\mathbb{Z}[i] \mid \mathcal{N}(\beta) = 1\}.$$

Moreover, the adjacency in the graph is determined by the elements with unitary norm. In the following sections, it will be proved that $\mathcal{G}_p$ induces a 2-quasi-perfect Lee code over $\mathbb{Z}_p^n$ under some conditions. Therefore, it must be determined which primes p are such that $\mathcal{G}_p$ has error correction capacity 2 and diameter 3.

The rest of the section is organized as follows. Subsection 4.3.2 proves that the Cayley graphs selected have error correction capacity 2. In Subsection 4.3.3 those Cayley graphs are shown to attain diameter 3, which concludes that they define 2-quasi-perfect codes. Finally, in Subsection 4.3.4 the results presented in this section are discussed, and some open problems and future lines of research are detailed.

4.3.2 Error Correction Capacity of $\mathcal{G}_p$

As explained in previous section, 2-quasi-perfect Lee codes are going to be obtained by means of Cayley graphs. In particular, it will be determined under which conditions the Cayley graph $\mathcal{G}_p$ over the additive group $\mathbb{Z}[i]/p\mathbb{Z}[i]$ and generating set those elements with unitary norm induces a 2-quasi-perfect code. In this section it will be proved that $p \equiv \pm 5 \pmod{12}$ implies that $\mathcal{G}_p$ has error correction capacity 2 over $\mathbb{Z}_p^n$ for $n = 2[\frac{p}{4}]$. Hence, in the remainder of the section, let us assume that $p > 2$ is a prime integer. Therefore, the natural number $n = 2[\frac{p}{4}]$ fulfills $p = 2n \pm 1$.

First, let us introduce some notation. Given a Gaussian integer $\beta = b_1 + b_2 i \in \mathbb{Z}[i]$, $\bar{\beta}$ will denote its conjugate, that is $\bar{\beta} = b_1 - b_2 i$. Also, $\Re(\beta) = b_1$ will stand for its real part and $\Im(\beta) = b_2$ for its imaginary part. Then, the following formula about the norm of a sum of Gaussian integers will be useful in several points of this section.

Lemma 4.3.5. *For any pair of Gaussian integers $\beta, \gamma \in \mathbb{Z}[i]$,*

$$\mathcal{N}(\beta + \gamma) = \mathcal{N}(\beta) + \mathcal{N}(\gamma) + 2\Re(\beta\bar{\gamma}).$$

Then, the previous result can be used to prove the following technical lemma:

Lemma 4.3.6. *For any $\gamma_1, \gamma_2 \in \mathbb{Z}[i]/p\mathbb{Z}[i]$, if $\mathcal{N}(\gamma_1) = \mathcal{N}(\gamma_2)$ and $\mathcal{N}(1+\gamma_1) = \mathcal{N}(1+\gamma_2)$ then $\gamma_1 \in \{\gamma_2, \bar{\gamma}_2\}$.*

Proof. Since $\mathcal{N}(1 + \gamma_1) = \mathcal{N}(1 + \gamma_2)$, by Lemma 4.3.5 it is obtained that $\Re(\gamma_1) = \Re(\gamma_2)$. Therefore, there are $x, y, z \in \mathbb{Z}/p\mathbb{Z}$ such that $\gamma_1 = x + yi$ and $\gamma_2 = x + zi$. Now, $\mathcal{N}(\gamma_1) = \mathcal{N}(\gamma_2)$ implies that $x^2 + y^2 = x^2 + z^2$. As a consequence, $y^2 = z^2$ and therefore $y \in \{\pm z\}$, which means $\gamma_1 \in \{\gamma_2, \bar{\gamma}_2\}$. □

Corollary 4.3.7. *Let $\beta \in \mathbb{Z}[i]/p\mathbb{Z}[i]$ be such that $\mathcal{N}(\beta) = 1$. Then, $1 + \beta$ is not a proper zero divisor.*

Proof. If $1 + \beta$ is a zero divisor then $\mathcal{N}(1 + \beta) = 0 = \mathcal{N}(1 + (-1))$. By Lemma 4.3.6, $\beta \in \{-1, \overline{-1}\} = \{-1\}$ and $1 + \beta = 0$. □

Let us denote by $G = \mathcal{U}(\mathbb{Z}[i]/p\mathbb{Z}[i])$ the multiplicative group formed by the units of the ring. Then, the set

$$H = \{\beta \in G \mid \mathcal{N}(\beta) = 1\}$$

is clearly a multiplicative normal subgroup of G. It is actually a cyclic group, although this fact will not be used in the proofs. Note that H is the set of adjacencies of $\mathcal{G}_p$, this is, $G = Cay(\mathbb{Z}[i]/p\mathbb{Z}[i]; H)$. For any $\gamma \in \mathbb{Z}[i]/p\mathbb{Z}[i]$, the following notation is introduced:

$$\gamma H = \{\gamma\beta \mid \beta \in H\}.$$

Notice that if $\gamma \in G$, then γH is the *coset* of H in G with respect to γ. Nevertheless, this notation is also defined for elements outside G, *i.e.*, for zero divisors of $\mathbb{Z}[i]/p\mathbb{Z}[i]$.

The following lemma tells us that cosets can be identified by the norms of its elements.

Lemma 4.3.8. *For any $\gamma \in G$,*

$$\gamma H = \{\beta \in \mathbb{Z}[i]/p\mathbb{Z}[i] \mid \mathcal{N}(\beta) = \mathcal{N}(\gamma)\}.$$

Proof. In order to prove the sets equality, it will be first proved that $\gamma H \subseteq \{\beta \in G \mid \mathcal{N}(\beta) = \mathcal{N}(\gamma)\}$. Thus, let us consider $\beta \in \gamma H$ and it has to be proved that $\mathcal{N}(\beta) = \mathcal{N}(\gamma)$. Since $\beta \in \gamma H$, then there exists $\eta \in H$ such that $\beta = \gamma\eta$. Hence $\mathcal{N}(\beta) = \mathcal{N}(\gamma)\mathcal{N}(\eta) = \mathcal{N}(\gamma)$.

Now, let us consider the other inclusion, that is, $\gamma H \supseteq \{\beta \in G \mid \mathcal{N}(\beta) = \mathcal{N}(\gamma)\}$. Therefore, let $\beta \in G$ be such that $\mathcal{N}(\beta) = \mathcal{N}(\gamma)$. Since γ is invertible, $\beta = \gamma(\beta\gamma^{-1})$. Now, as $\mathcal{N}(\beta\gamma^{-1}) = 1$ it is obtained that $\beta \in \gamma H$. □

Theorem 4.3.10 states that the degree of the graph $\mathcal{G}_p$ is $2n$. To prove it some particular cases of the Quadratic Reciprocity Law will be necessary, which are recalled in the following theorem for self-containedness.

Theorem 4.3.9 (Quadratic Reciprocity Laws)**.** *If p is an integer prime, then:*

i) The number of solutions to $-1 = x^2$ in $\mathbb{Z}/p\mathbb{Z}$ is:

- *2 if $p \equiv 1 \pmod 4$,*
- *1 if $p = 2$ and*
- *0 if $p \equiv 3 \pmod 4$.*

ii) The number of solutions to $3 = x^2$ in $\mathbb{Z}/p\mathbb{Z}$ is:

- *2 if $p \equiv \pm 1 \pmod{12}$,*
- *1 if $p = 3$ or $p = 2$ and*
- *0 otherwise.*

Theorem 4.3.10. *For any odd prime integer p, let $n = 2[\frac{p}{4}]$. Then,*

$$|H| = |\{\beta \in \mathbb{Z}[i]/p\mathbb{Z}[i] \mid \mathcal{N}(\beta) = 1\}| = 2n.$$

Proof. It is clear that

$$|H| = |\{(x, y) \mid x, y \in \mathbb{Z}/p\mathbb{Z},\ x^2 + y^2 = 1\}|.$$

Therefore, let us consider the solutions of $x, y \in \mathbb{Z}/p\mathbb{Z}$ of equation $x^2 + y^2 = 1$. First, if $x = 1$ then $y^2 = 0$ whose unique solution is $y = 0$. Let us assume $x \neq 1$ to look for the rest of solutions. Since $x \neq 1$, $x-1$ has inverse and it is possible to define $s = y/(x-1) \in \mathbb{Z}/p\mathbb{Z}$. By considering the intersection of the straight line $y = s(x-1)$ with the curve $x^2 + y^2 = 1$ it is obtained that $x^2 + (s(x-1))^2 = 1$. The only solutions of this equation are $x = 1$ (which has already been considered) and $x = \frac{s^2-1}{s^2+1}$. This second solution for x equals 1 if and only if $p = 2$. Thus, the only solutions with $x \neq 1$ are $x = \frac{s^2-1}{s^2+1}$ and $y = \frac{-2s}{s^2+1}$.

Now, for each possible value of s, there is one solution with this form, that is, p minus the number of solutions of $s^2 + 1 = 0$. By the Quadratic Reciprocity Law (first item of Theorem 4.3.9) there are $p + 1$ solutions if $p \equiv 3 \pmod 4$ and $p - 1$ if $p \equiv 1 \pmod 4$. Thus, for primes of the form $p = 1 + 4k$, there are $p - 1 = 4k = 2n$ solutions and for primes $p = -1 + 4k$ there are $p + 1 = 4k = 2n$ solutions, where $k \in \mathbb{N}$.

Finally, just to ensure that the counted solutions are all different, note that if for a pair s_1, s_2 the same solution (x, y) is obtained, then $s_1 = s_2 = y/(x-1)$. □

Next, it can be easily obtained the following consequence of previous theorem, which will be used in Subsection 4.3.3 to determine the diameter of the graph $\mathcal{G}_p$.

Corollary 4.3.11. *For any odd prime integer p, let $n = 2[\frac{p}{4}]$. If $0 \neq \gamma \in \mathbb{Z}[i]/p\mathbb{Z}[i]$ then $|\gamma H| = 2n$.*

Proof. Firstly, note that if $\gamma \in G$, then γH is a coset, which are widely known to have the same cardinal. Thus, the non-immediate part of the proof lies on the zero divisors. By Theorem 4.3.10, it is straightforward that $|\gamma H| \leq 2n$. Proceeding by *reductio ad absurdum*, let us assume $|\gamma H| < 2n$. Then, there exist $\beta_1 \neq \beta_2$ such that $\gamma\beta_1 = \gamma\beta_2$, thus $\gamma(\beta_1 - \beta_2) = 0$. Since $\gamma \neq 0$ then $\beta_1 - \beta_2$ must be a zero divisor. Now, multiplying by β_1^{-1}, $1 - \beta_2\beta_1^{-1}$ is also a zero divisor. By Corollary 4.3.7, $1 - \beta_2\beta_1^{-1} = 0$ and hence $\beta_1 = \beta_2$, which is a contradiction. □

Before stating the conditions under which $\mathcal{G}_p$ has error correction capacity 2, the following lemma is going to be proved. This lemma determines the number of possible norms among the neighbours of a vertex with a given norm.

Lemma 4.3.12. *For any $c \in \mathbb{Z}/p\mathbb{Z}$, $c \neq 0$, let us consider the set $N_p(c) = \{\mathcal{N}(1+\beta) \mid \mathcal{N}(\beta) = c\} \subset \mathbb{Z}/p\mathbb{Z}$. Then, it is obtained that:*

$$|N_p(c)| = \begin{cases} n+1 & \text{if } c \text{ is a square residue mod } p, \\ n & \text{if } c \text{ is not a square residue mod } p. \end{cases}$$

Proof. In the first case, that is c being a square residue, there must exists $s \in \mathbb{Z}/p\mathbb{Z}$ such that $c = s^2$. By Lemma 4.3.8 and Corollary 4.3.11 there are $2n$ elements with norm c, which are:

$$\{\beta \mid \mathcal{N}(\beta) = c\} = \{s, -s, \beta_1, \beta_2, \ldots, \beta_{n-1}, \bar{\beta}_1, \bar{\beta}_2, \ldots, \bar{\beta}_{n-1}\},$$

for some $\beta_1, \ldots, \beta_{n-1} \in \mathbb{Z}[i]/p\mathbb{Z}[i]$. Then,

$$\begin{aligned} N_p(c) &= \{\mathcal{N}(1+\beta) \mid \mathcal{N}(\beta) = c\} \\ &= \{\mathcal{N}(1+s), \mathcal{N}(1-s), \mathcal{N}(1+\beta_1), \mathcal{N}(1+\beta_2), \ldots, \mathcal{N}(1+\beta_{n-1})\}, \end{aligned}$$

which are different by Lemma 4.3.6. Hence $|N_p(c)| = 2 + (n-1) = n+1$.

For the case of c being a square non-residue let us proceed in a similar way. It is obtained that

$$\{\beta \mid \mathcal{N}(\beta) = c\} = \{\beta_0, \beta_1, \beta_2, \ldots, \beta_{n-1}, \bar{\beta}_0, \bar{\beta}_1, \bar{\beta}_2, \ldots, \bar{\beta}_{n-1}\}.$$

Then

$$\begin{aligned} N_p(c) &= \{\mathcal{N}(1+\beta) \mid \mathcal{N}(\beta) = c\} \\ &= \{\mathcal{N}(1+\beta_0), \mathcal{N}(1+\beta_1), \mathcal{N}(1+\beta_2), \ldots, \mathcal{N}(1+\beta_{n-1})\}, \end{aligned}$$

which are different by Lemma 4.3.6. Hence $|N_p(c)| = n$. □

As it will be noted afterwards, the case $c = 1$ in previous lemma is going to be used to prove the error correction capacity. Later, the fact that n is a lower bound of $N_p(c)$ will be considered to determine the graph diameter.

To finish the section, next theorem establishes the conditions for p such that $\mathcal{G}_p$ has error correction capacity 2.

Theorem 4.3.13. *Let p be a prime integer satisfying $p \equiv \pm 5 \pmod{12}$. Let $n = 2\left[\frac{p}{4}\right]$. Then, the Cayley graph $\mathcal{G}_p$ has error correction capacity 2.*

Proof. As it was explained in previous section, it has to be proved that $\mathcal{G}_p$ contains $|B_2^n| = 2n^2 + 2n + 1$ vertices at distance 2 or less from 0. Clearly, 0 is the unique vertex at distance 0. Now, the set H contains all the vertices at distance 1 and $|H| = 2n$ by Theorem 4.3.10.

The vertices at distance 2 is the set $A = \{\beta_a + \beta_b \mid \beta_a, \beta_b \in H\} \setminus (H \cup \{0\})$. Thus, let us prove that $|A| = 2n^2$. By Lemma 4.3.8 and Corollary 4.3.11, $|A| = 2n \cdot |N_p(1) \setminus \{0, 1\}|$. Since 1 is always a square residue for any p, hence by Lemma 4.3.12, $|N_p(1) \setminus \{0\}| = n$. It remains to be proved that 1 does not belong to $N_p(1)$.

Suppose that there is β with $\mathcal{N}(\beta) = 1$ and $\mathcal{N}(1 + \beta) = 1$. Then, by Lemma 4.3.5, $1 = 2 + 2\Re(\beta)$ and hence $\Re(\beta) = -2^{-1}$. Let $\beta = -2^{-1} + yi$, which implies $1 = \mathcal{N}(\beta) = 2^{-2} + y^2$. Then, $3 = (2y)^2$, which only has solutions for $p = 3$ or $p \equiv \pm 1 \pmod{12}$ by the second item of Theorem 4.3.9. Thus, $|N_p(1) \setminus \{0, 1\}| = |N_p(1) \setminus \{0\}| = n$ and $|A| = 2n \cdot n$, which concludes the proof. □

Remark 4.3.14. *If p is a prime greater than 3 that does not satisfy $p \equiv \pm 5 \pmod{12}$, then $p \equiv \pm 1 \pmod{12}$. In this case, $\mathcal{G}_p$ only contains $2n^2 + 1$ vertices at distance 2 or less from vertex 0. Although it is not a 2-error correcting code, it is very close since only $2n$ syndromes cannot be corrected.*

4.3.3 Diameter of $\mathcal{G}_p$

In this section it will be proved that $\mathcal{G}_p$ has diameter 3 for any prime $p > 5$. The proof will be separated into two subsections, the first one considering the case $p \equiv 3 \pmod 4$ and the second one the case $p \equiv 1 \pmod 4$. Also, from here onwards it will be assumed again that $n = 2[\frac{p}{4}]$. Note that, since $|\mathbb{Z}[i]/p\mathbb{Z}[i]| = p^2 > |B_2^n|$, there are vertices outside the sphere of radius 2, which means that the diameter of the graph is at least 3. As it will be seen next, the proofs proceed by *reductio ad absurdum* by the assumption of the existence of a vertex at a distance 4 from vertex 0, thus reaching a contradiction.

Case $p \equiv 3 \pmod 4$

In this case the proof of the diameter can be easily obtained by using a counting argument. Note that in this case $p = 2n - 1$ and therefore $\mathbb{Z}[i]/p\mathbb{Z}[i]$ is a field.

Theorem 4.3.15. *For any prime p such that $p \equiv 3 \pmod 4$ the graph $\mathcal{G}_p$ has diameter 3.*

Proof. By *reductio ad absurdum* let us assume that there exists a vertex $\gamma \in \mathbb{Z}[i]/p\mathbb{Z}[i]$ at distance 4 of vertex 0. Let $c = \mathcal{N}(\gamma)$. Since γ is so far, it is obtained that $N_p(1) \cap N_p(c) = \emptyset$.

Let us denote by $W_t(0)$ the number of vertices at a distance t from vertex 0. Then, $\{W_t(0) \mid t = 0, \ldots, 4\}$ is the distance distribution of the graph $\mathcal{G}_p$. Now, the cardinals $W_1(0) = |H|$ and $W_4(0) \geq |\gamma H|$ can be calculated by Corollary 4.3.11. Also, by Lemma 4.3.12 it can be computed that $|N_p(1)| = n + 1$ and $|N_p(c)| \geq n$. Thus, the bounds for the distance distribution obtained are summarized next:

$$\begin{array}{ll} W_0(0) = |\{0\}| & = 1 \\ W_1(0) = |H| & = 1 \cdot 2n \\ W_2(0) = 2n \cdot |N_p(1) \setminus \{0, 1\}| & \geq (n-1) \cdot 2n \\ W_3(0) \geq 2n \cdot |N_p(c) \setminus \{c\}| & \geq (n-1) \cdot 2n \\ W_4(0) \geq |\gamma H| & = 1 \cdot 2n \end{array}$$

As a consequence, the total number of vertices satisfies $|\mathbb{Z}[i]/p\mathbb{Z}[i]| \geq 1 + 2n(1 + (n-1) + (n-1) + 1) = 4n^2 + 1 > 4n^2 - 4n + 1 = p^2 = |\mathbb{Z}[i]/p\mathbb{Z}[i]|$, which is a contradiction. □

Case $p \equiv 1 \pmod 4$

Unfortunately, the reasoning of the previous case fails to give us a contradiction if $p \equiv 1 \pmod 4$. Thereof, it will be needed to resort to the tight bound from algebraic geometry obtained in the Hasse–Weil Theorem. Note that, in this case, $p = 2n + 1$ and the ring $\mathbb{Z}[i]/p\mathbb{Z}[i]$ contains zero divisors.

First, let us prove two technical lemmas that analyze what happens with the zero divisors of the ring.

Lemma 4.3.16. *For any proper zero divisor $\zeta \in \mathbb{Z}[i]/p\mathbb{Z}[i]$,*

$$\zeta H = \{x\zeta \mid x \in \mathbb{Z}/p\mathbb{Z},\ x \neq 0\}.$$

Proof. On one hand, by Corollary 4.3.11, the cardinal $|\zeta H|$ is $2n$. On the other hand, $|\{x\zeta \mid x \in \mathbb{Z}/p\mathbb{Z},\ x \neq 0\}|$ has $p - 1 = 2n$ elements. Since both sets have the same size, it is enough to prove one inclusion to show the sets equality. Therefore, let us prove the left to right inclusion.

Let $\beta = a + bi$ be an element of norm 1 and $\zeta = u + vi$ a proper zero divisor, hence of norm 0. As $\zeta \neq 0$ and $\mathbb{Z}/p\mathbb{Z}$ is a field, both u and v are nonzero. Let us define $x = a - b\frac{v}{u} \in \mathbb{Z}/p\mathbb{Z}$. Therefore,

$$\begin{aligned} x\zeta = (a - b\frac{v}{u})(u + vi) = (au - bv) + (av - b\frac{v^2}{u})i = (au - bv) + (av - b\frac{-u^2}{u})i \\ = (au - bv) + (av + bu)i = (a + bi)(u + vi) = \beta\zeta. \end{aligned}$$

Finally, note that x cannot be zero, since it would imply that β were a zero divisor, contradicting $\mathcal{N}(\beta) = 1$. □

The following lemma has its inspiration in Lemma 4.3.12, but with the intention to generalize to the case of zero divisors and to give a stronger result.

Lemma 4.3.17. *For any proper zero divisor $\zeta \in \mathbb{Z}[i]/p\mathbb{Z}[i]$,*

$$\{\mathcal{N}(\beta + \zeta) \mid \mathcal{N}(\beta) = 1\} = \mathbb{Z}/p\mathbb{Z} \setminus \{1\}.$$

Proof. Let $\zeta = u + vi$ be a proper zero divisor. By Lemma 4.3.16 and by making few calculations,

$$\begin{aligned} \{\mathcal{N}(\beta + \zeta) \mid \mathcal{N}(\beta) = 1\} &= \{\mathcal{N}(1 + \beta\zeta) \mid \mathcal{N}(\beta) = 1\} \\ &= \{\mathcal{N}(1 + x\zeta) \mid x \in \mathbb{Z}/p\mathbb{Z},\ x \neq 0\} = \{1 + 2xu \mid x \in \mathbb{Z}/p\mathbb{Z},\ x \neq 0\}. \end{aligned}$$

To finish, note that $y = 1 + 2xu$ with $x \neq 0$ has solution for every value of y except 1. □

The previous lemma indicates that proper zero divisors are neighbours of every vertex at distance 2 from 0, and hence they are at distance 3 from 0. Then, the following lemma gives a polynomial description of the sets $N_p(t)$.

Lemma 4.3.18. *Let $p \equiv 1 \pmod 4$ be a prime in $\mathbb{Z}$. For any $t \in \mathbb{Z}/p\mathbb{Z}$, $t \neq 0$, it is obtained that*

$$N_p(t) = \{x^{-1}(x + 1)(x + t) \mid x \in \mathbb{Z}/p\mathbb{Z},\ x \neq 0\}.$$

Proof. By the first item of Theorem 4.3.9, there exists $r \in \mathbb{Z}/p\mathbb{Z}$ such that $r^2 = -1$. Note that $x^{-1}(x+1)(x+t) = x + tx^{-1} + t + 1$. First, let us prove the left to right inclusion of the sets. In this aim, let $\beta = a + bi$, $\mathcal{N}(\beta) = a^2 + b^2 = t$ for a generic element $\mathcal{N}(1+\beta)$ in $N_p(t)$. Thus, let us check that $x = a + rb$ satisfies $\mathcal{N}(1+\beta) = x + tx^{-1} + t + 1$. By Lemma 4.3.5, $x\mathcal{N}(1+\beta) = x(\mathcal{N}(1) + \mathcal{N}(\beta) + 2\Re(\beta)) = x(t+1) + 2ax$. Hence,

$$\begin{aligned} x(x + tx^{-1} + t + 1) - x\mathcal{N}(1+\beta) &= x^2 + t - 2ax \\ &= t + (a+rb)^2 - 2a(a+rb) \\ &= t + (a^2 + 2rab + r^2b^2) - (2a^2 + 2rab) \\ &= t - a^2 + r^2b^2 \\ &= t - a^2 - b^2 \\ &= 0. \end{aligned}$$

For the right to the left inclusion, let $x \neq 0$ and $y = x^{-1}(x+1)(x+t)$ an element of $\{x^{-1}(x+1)(x+t) \mid x \in \mathbb{Z}/p\mathbb{Z},\ x \neq 0\}$. Now, define $\beta = x + x^{-1}(t-x^2) + 2^{-1}x^{-1}(t-x^2)ri$. Then, by calculation $\mathcal{N}(\beta) = (x + x^{-1}(t-x^2))^2 + (2^{-1}x^{-1}(t-x^2)r)^2 = t$. Moreover, $\mathcal{N}(1+\beta) = 1 + t + 2\Re(\beta) = 1 + t + 2x + x^{-1}(t-x^2) = y$, which ends the proof. □

The intersection between $N_p(1)$ and $N_p(t)$ will be given by the roots of polynomial $P_t(x,y) = y(x+1)^2 - x(y+1)(y+t)$. In order to apply the Hasse–Weil bound, the polynomial must be irreducible. Therefore, let us introduce the following definition and two useful results in Lemma 4.3.20 and Corollary 4.3.21.

Definition 4.3.19. *Given a field $\mathbb{F}$, a polynomial $P \in \mathbb{F}[x,y]$ is called absolutely irreducible if it is irreducible in the algebraic closure of $\mathbb{F}$.*

Lemma 4.3.20. *For any prime p, the polynomial $P_t(x,y) = y(x+1)^2 - x(y+1)(y+t) \in \mathbb{Z}_p[x,y]$ is absolutely irreducible for $t \neq 0, 1$.*

Proof. The polynomial $P_t(x,y) = xy(x-y) + (1-t)xy + y - tx$ has degree 3. If $P_t(x,y)$ is not absolute irreducible, then there exist polynomials $A(x,y)$, $B(x,y)$ with coefficients in the algebraic closure of $\mathbb{Z}/p\mathbb{Z}$ such that $P_t(x,y) = AB$ with $\deg A(x,y) = 2$ and $\deg B(x,y) = 1$. Furthermore, the product of the leading terms of $A(x,y)$ and $B(x,y)$ must be $xy(x-y)$. Let us consider the following three mutually exclusive cases, depending on polynomials $A(x,y)$ and $B(x,y)$

i) Case $A(x,y) = (xy + ax + by + c)$, $B(x,y) = (x - y + d)$. The coefficient of x^2 in $A(x,y) \cdot B(x,y)$ is a and the one of y^2 is $-b$. By hypothesis, both are 0 in $P_t(x,y)$. Then, the coefficient of xy is $d = 1 - t$, the coefficient of x is $c = -t$ and the coefficient of y is $-c = t = 1$. Hence, for $t = 1$ there exists the factorization $P_1(x,y) = (xy - 1)(x - y)$.

ii) Case $A(x,y) = (x(x-y) + ax + by + c)$, $B(x,y) = (y + d)$. Now, the coefficient of x^2 in $A(x,y) \cdot B(x,y)$ is $d = 0$ and the coefficient of y^2 is $b = 0$. Then, the coefficient of xy is $a = 1 - t$, the coefficient of x is $0 = -t$ and the coefficient of y is $c = 1$. Hence, for $t = 0$ there exists the factorization $P_0(x,y) = (x^2 - xy + x + 1)y$.

iii) Case $A(x,y) = (y(x-y) + ax + by + c)$, $B(x,y) = (x + d)$. The coefficient of x^2 is $a = 0$ and the coefficient of y^2 is $-d = 0$. Then, the coefficient of y would be $0 = 1$, which implies that there exists no factorization.

Finally, there are factorizations of $P_t(x, y)$ only for $t = 0$ and $t = 1$, which proves the result. □

Corollary 4.3.21. *The homogeneous polynomial*

$$^hP_t(x, y, z) = xy(x - y) + (1 - t)xyz + (y - tx)z^2$$

is absolutely irreducible for $t \neq 0, 1$.

Proof. If $^hP_t(x, y)$ had a factorization, then its evaluation at $z = 1$ would be a factorization of $P_t(x, y)$, contradicting Lemma 4.3.20. □

Finally, let us conclude the section by proving the main result.

Theorem 4.3.22. *If* p *is a prime such that* $p \equiv 1 \pmod 4$ *and* $p > 5$*, then the diameter of* $\mathcal{G}_p$ *is 3.*

Proof. Let us proceed again by *reductio ad absurdum*. In this aim, let us assume the existence of a vertex γ at distance 4 from 0 in $\mathcal{G}_p$, with p fulfilling the hypothesis of the statement. Let $t = \mathcal{N}(\gamma)$. Note that $t \neq 1$ since vertices with norm equal to 1 are at distance 1. Also, $t \neq 0$ by Lemma 4.3.17. Hence, by Lemma 4.3.8, the vertices with norm in the set $N_p(t) \setminus \{0\}$ are at distance at least 3. Meanwhile, the vertices with norm in $N_p(1) \setminus \{0\}$ are at distance at most 2 from 0. Therefore, the intersection of previous two sets is $N_p(1) \cap N_p(t) = \{0\}$.

Now, using polynomial notation, the previous set equality is equivalent, by Lemma 4.3.18, to the non-existence of solutions to $x^{-1}(x + 1)^2 = y^{-1}(y + 1)(y + t)$ other than $x = -1$. Let us highlight that the solution $x = -1$ corresponds with norm 0. Thus, vertices in H have vertex 0 as their neighbour, while vertices in γH have as some of their neighbours vertices that are proper zero divisors.

The contradiction will be obtained when proving the existence of a solution to the equation $P_t(x, y) = 0$ other than the trivial ones $(x, y) \in \{(0, 0), (-1, -1), (-1, -t)\}$. In this aim, let us define the varieties

$$V_t = \{(x, y) \in (\mathbb{Z}/p\mathbb{Z})^2 \mid P_t(x, y) = 0\},$$

$$X_t = \{(x : y : z) \in \mathbb{P}^2_{\mathbb{Z}/p\mathbb{Z}} \mid {}^hP_t(x, y, z) = 0\},$$

where $\mathbb{P}^2_{\mathbb{Z}/p\mathbb{Z}}$ denotes the projective space of dimension 2 over $\mathbb{Z}/p\mathbb{Z}$. The notation $(x : y : z)$ indicates a projective point, which is the same point as $(\lambda x : \lambda y : \lambda z)$ for any $\lambda \neq 0$. Thus, affine solutions can be recovered by taking $\lambda = z^{-1}$; except for solutions $(x : y : 0)$, which are the points at the infinity.

Hasse–Weil's theorem [Coh07] states that

$$\big||X_t| - (p + 1)\big| \leq 2\sqrt{p},$$

for absolutely irreducible polynomial curves X_t of degree 3. Note that, by Corollary 4.3.21, Hasse–Weil's theorem can be applied to $^hP_t(x, y, z)$. Therefore,

$$|X_t| \geq p + 1 - 2\sqrt{p}.$$

Now, the only 3 projective solutions for $^hP_t(x, y, z) = 0$ with $z = 0$ are $(x : y : z) \in \{(0 : 1 : 0), (1 : 0 : 0), (1 : 1 : 0)\}$. Thus, $|V_t| = |X_t| - 3$, which implies that

$$|V_t| \geq p - 2 - 2\sqrt{p}.$$

As a consequence, those primes p such that $|V_t| \geq 4$ give the contradiction looked for. Clearly, if $p \geq 17$ then,

$$|V_t| \geq p - 2 - 2\sqrt{p} \geq 17 - 2 - 2\sqrt{17} \geq 6.7.$$

Finally, the unique prime $p \equiv 1 \pmod 4$ such that $5 < p < 17$ is 13. In this particular case, it can be computed that $|V_t| \geq 9$ for any t, which concludes the proof. □

Remark 4.3.23. *$\mathcal{G}_5$ has diameter 4 since, vertex $2 + 2i$ and its associates are at distance 4 from vertex 0.*

4.3.4 Discussion

In this final subsection, conclusions of this work and future research are going to be presented. In the first subsection, the main result is rewritten using parity-check matrices. Also, a formal proof of the infiniteness of the constructed family of quasi-perfect codes is given. Some considerations on the density of the codes are addressed. Moreover, other examples of codes presenting greater density and an upper error correction capacity are shown. In the final subsection, the authors aim to exhibit the connections of the graphs considered in the present study with other graph theoretical problems, trying to give a new insight into the perfect Lee codes conjecture formulated by Golomb and Welch more than forty years ago.

Quasi-Perfect Lee codes

As it has been proved in previous Subsections 4.3.2 and 4.3.3, $\mathcal{G}_p$ has error correction capacity 2 and diameter 3, for any prime $p > 5$ and $p \equiv \pm 5 \pmod{12}$.

Dirichlet's theorem on arithmetic progressions asserts that in any arithmetic progression whose initial term is coprime with its increment there are infinitely many primes. As a natural consequence, congruences can be considered as arithmetics progressions, and therefore it can be straightforwardly obtained the following:

Corollary 4.3.24. *There are infinitely many $n \in \mathbb{N}$ such that $p = 2n \pm 1$, $p \geq 7$ prime in $\mathbb{Z}$, $p \equiv \pm 5 \pmod{12}$.*

Then, when applying the previous result it is obtained:

Corollary 4.3.25. *The family of graphs $\mathcal{G}_p$ contains infinitely many graphs with error correction capacity 2 and diameter 3.*

Now, as it was argued in Section 4.1, each of these graphs induces a 2-quasi perfect Lee code. Let us consider $\mathcal{G}_p = Cay(\mathbb{Z}[i]/p\mathbb{Z}[i], \{\beta_1, \ldots, \beta_{2n}\})$, where $\beta_1, \ldots, \beta_{2n}$ are the elements in $\mathbb{Z}[i]/p\mathbb{Z}[i]$ with unitary norm and they are associates of the first $\frac{n}{2}$ elements: $\beta_1, \ldots, \beta_{\frac{n}{2}}$.

The set of generators of the Cayley graph defines the parity-check matrix, that is,

$$M = \left(\beta_1, \ldots, \beta_{\frac{n}{2}}\right).$$

This can be verified by realizing that the word $\mathbf{c}$ is associated to vertex $M\mathbf{c}$ and that $\mathbf{c}$ belongs to the code if and only if it is associated to vertex 0; *i.e.*, the codewords are exactly the words $\mathbf{c}$ such that $M\mathbf{c} = 0$.

However, the code associated to this matrix belongs to the space $(\mathbb{Z}[i]/p\mathbb{Z}[i])^{n/2}$. In order to obtain the Lee code over $(\mathbb{Z}/p\mathbb{Z})^n$ and a parity-check matrix with integer entries, every β has to be substituted by

$$\beta \mapsto \begin{pmatrix} \Re(\beta) & -\Im(\beta) \\ \Im(\beta) & \Re(\beta) \end{pmatrix}.$$

Therefore, the parity-check matrix associated to the graph $\mathcal{G}_p$ is

$$\begin{pmatrix} \Re(\beta_1) & -\Im(\beta_1) & \cdots & \Re(\beta_{\frac{n}{2}}) & -\Im(\beta_{\frac{n}{2}}) \\ \Im(\beta_1) & \Re(\beta_1) & \cdots & \Im(\beta_{\frac{n}{2}}) & \Re(\beta_{\frac{n}{2}}) \end{pmatrix}.$$

Now, let us give some considerations on the quality of the codes constructed. Note that, since the Lee sphere of radius 2 contains $|B_2| = 2n^2 + 2n + 1$ words, then the graph induced by any 2-quasi-perfect linear code has at least $2n^2 + 2n + 1$ vertices. The graphs $\mathcal{G}_p$ constructed in this section have p^2 vertices. Therefore, for the case $p = 2n + 1$, the number of vertices is $p^2 = 4n^2 + 4n + 1 = 2|B_2| - 1$. Also, for the case $p = 2n - 1$, the number of vertices is $p^2 = 4n^2 - 4n + 1 = 2|B_2| - 8n - 1$. Thus, the reached vertices are asymptotically the double of which would be reached in the graph associated to a perfect code. In other words, the density of the codes presented is $\frac{1}{p^2}$.

Although the obtained density is quite good, for some small cases (low dimension), graphs with a smaller number of vertices have been computationally found. Let us consider the following examples.

Example 4.3.26. *Let $n = 8$ be the dimension and $p = 13$. The set of generators of the Cayley graphs will be $H = \pm\{1, 4 + 10i, 8, 7 + 11i\}$. In this case the Cayley graph $\mathcal{G} = Cay(\mathbb{Z}[i]/p\mathbb{Z}[i]; H)$ induces a 2-quasi-perfect code. Note that $\mathcal{G}$ has $p^2 = 169$ vertices, which is just 17% over $|B_2^8| = 145$, the cardinal of the sphere in this dimension.*

Example 4.3.27. *Let $n = 16$ be the dimension. In this case, by extending the search to a different ring, a new graph has been found. The graph is build over the Quaternion integers modulo $p = 5$, being the generator set $H = \pm\{1, 1 + 2i + 3j, 3i + 4j + 1k, 3 + 4i + 3j\}$. In this case, the number of vertices of the graph is $p^4 = 625$, which is 15% over $|B_2^{16}| = 545$.*

The previous small examples suggest that there exist codes very close to be perfect, although general constructions appear to be difficult to find.

From the result by Post [Pos75] it was obtained that there is no perfect code with radius greater than the dimension of the space. Previously to that paper, Golomb and Welch [GW70] had already noted that there cannot be perfect codes with correction greater than a constant that depends on the dimension by the use of the maximum density of packing with cross-polytopes. Clearly, this can be applied to quasi-perfect codes. For every n there exists t_n such there are not t-quasi-perfect codes for $t \geq t_n$. Hence, this might suggest that the radius 2 case is an exceptional one. Nevertheless, a few 3-quasi-perfect codes have been found for small dimensions. Note that in this case the n-dimensional sphere of radius 3 has cardinal $|B_3^n| = \frac{1}{3}(1 + 2n)(3 + 2n + 2n^2)$. The examples that we have found are summarized in Table 4.2. The codes are obtained from Cayley graphs $Cay(\mathbb{Z}[i]/p\mathbb{Z}[i]; H)$, for parameters n, p, H indicated in the table. As it can be seen, the first example is just 31% over the cardinal of the sphere, while the second and third are 79% and 102%, respectively. Any of the three examples can be considered as 3-quasi-perfect codes really near to the perfect one.

In the authors opinion, the construction of an infinite family of graphs containing these codes or similar ones would have a great value, both practical and in a better understanding of the Golomb and Welch conjecture.

n	p	H	p^2	$\lvert B_3^n\rvert$
4	13	$\pm\{1, 3+4i\}$	169	129
6	26	$\pm\{1, 4+4i, 9+11i\}$	676	377
8	41	$\pm\{1, 2+13i, 6+18i, 11+i\}$	1681	833

Table 4.2: Some 3-quasi-perfect Lee codes over $\mathbb{Z}_p^n$.

Related Problems

Other interesting problems from different areas than from Coding Theory could be profited from this study. One example is the degree diameter problem over Abelian groups, as discussed in Section 4.1. In the degree-diameter problem the perfect case is approximated by below—in the number of vertices—while our construction has approximated it by above. Specifically, a construction by Macbeth *et al.* [MŠŠ12] obtains graphs of degree $2n$ and diameter 2 with cardinal approximately $\frac{3}{4}|B_2^n|$, whilst our construction have error capacity 2 and about $2|B_2^n|$ vertices.

Furthermore, the graphs considered in this section seemed to be good expanders. Therefore, the spectrum of some of them was computed and the obtained values exhibit that they are Ramanujan graphs. *Ramanujan graphs* are good expander graphs that attain the spectral bound [DSV03]. More specifically, $\mathcal{G}$ is a Ramanujan graph if and only if for every eigenvalue of its adjacency matrix λ it is hold either $|\lambda| = \deg(\mathcal{G})$ or $|\lambda| \leq 2\sqrt{\deg(\mathcal{G}) - 1}$. For the case of $\mathcal{G}$ being a Cayley graph some interesting properties are known. Let $\mathcal{G} = Cay(G; S)$ for a Abelian group G, then, there is a *group character* (an homomorphism into $(\mathbb{C} \setminus \{0\}, \cdot)$) χ_α for every $\alpha \in G$, and the eigenvalues are given by $\lambda_\alpha = \sum_{\beta \in S} \chi_\alpha(\beta)$. In the case of the group $\mathbb{Z}[i]/p\mathbb{Z}[i]$ the characters are

$$\chi_\alpha(\beta) = e^{\frac{2\pi i}{p}(\Re(\alpha)\Re(\beta)+\Im(\alpha)\Im(\beta))},$$

from which follows the eigenvalues of $\mathcal{G}_p$:

$$\lambda_\alpha = \sum_{\beta,\ \mathcal{N}(\beta)=1} e^{\frac{2\pi i}{p}(\Re(\alpha)\Re(\beta)+\Im(\alpha)\Im(\beta))}.$$

Clearly $\lambda_0 = |H|$; for the other values it seems plausible that Weil's conjectures imply $|\lambda_\alpha| \leq 2\sqrt{p}$ for $\alpha \neq 0$. Noticing in addition that if $p \equiv -1 \pmod 4$ then $p = \deg(\mathcal{G}_p) - 1$ and that if $p \equiv 1 \pmod 4$ then $p = \deg(\mathcal{G}_p) + 1$, it is natural to formulate the following conjecture:

Conjecture 4.3.28. *$\mathcal{G}_p$ is a Ramanujan graph for any prime $p \equiv 3 \pmod 4$.*

This conjecture has been verified for all primes $p < 1000$; the only primes in that range for which $\mathcal{G}_p$ is not Ramanujan are 17, 53 and 541. Moreover, we believe that *most primes* fulfilling $p \equiv 1 \pmod 4$ give also $\mathcal{G}_p$ being Ramanujan graphs. Therefore, the proof of this conjecture and the study of the relation between Golomb and Welch conjecture and spectral analysis will be considered as future work.

Chapter 5

Some Experimental Evaluations

This chapter is devoted to explain the simulation infrastructure utilized and to show the results of simulations related to the previous chapters. In Chapter 2 lattice graphs were studied, giving emphasis to symmetry, and some topologies were proposed as alternatives for actual supercomputers. Simulations will show how those alternatives are competitive. Symmetry will be empirically verified to have a large impact on performance. It will be also studied how applications that are designed for tori can run in other lattice graph with some improvements. In Chapter 3, the dragonfly topology was studied, providing deadlock-free routing algorithms that make use of global trunking and symmetry. In this chapter those routings are evaluated and shown to give similar performance and allowing for more variety in the amount of resources. Symmetry is also evaluated for dragonflies, showing that, at difference than lattice graphs, symmetries do not increase throughput.

Section 5.1 details the network simulator used in the experiments. Some simulation-related concepts are introduced and changes on the simulator are accounted. Section 5.2 explains NASA Advanced Supercomputing Division's (NAS) Parallel Benchmarks (NPB) and the traces obtained from them. Section 5.3 shows how symmetry impacts on the network performance in lattice-graphs. Networks with different values of average distance and symmetry are simulated and their effect on the performance is measured. Section 5.4 makes an study on how the twisted links of the RTT can break initially the locality of some applications and how it can be fixed by modifying the application mapping. For that, it introduces a traffic model generalizing uniform traffic by adding traffic local respect to a torus. Then, it is studied analytically and experimentally that using proper mapping functions the RTT topology usually outperforms the rectangular tori. Section 5.5 evaluates 4-dimensional lattice graphs. Some current Blue Gene systems with mixed radix tori as topologies are compared against symmetric lattice networks of the same size. Section 5.6 shows that the throughput of dragonfly topologies is unaffected by symmetry. In these networks, the load can be well-distributed among the links, even if there are not automorphisms among them. Section 5.7 gives the performance of the deadlock-free routing algorithms introduced in Chapter 3 for dragonfly networks with global trunking. These routing algorithms are shown to have comparable performance to other proposals. Then, it is studied their performance using different numbers of VCs—since more VCs reduce HoLB. Note that previous routings did not admit to change arbitrarily the number of VCs.

5.1 The FSIN simulator

All the experiments in this chapter have been realized using the functional simulator called FSIN [RPM05, NMPR11], which is part of INSEE (Interconnection Network Simulation and Evaluation Environment). FSIN is a time driven simulator of network routers. Each router has several ports, where each port can be an input port, an output port or connect by a link to another router. The time is measured in router cycles. Compute nodes only create and consume packets and are associated by an input and output port to a router. When a packet is created it has a destination compute node. Packets are divided into phits, the minimum data unit that can be manipulated every cycle. Each port has several buffers or virtual channels. Tori and other lattice-graphs employ the Bubble Adaptive Router presented in [CBGV97] and currently used by IBM Blue Gene supercomputers with three virtual channels.

Measures of interest of a simulation are throughput, which is measured in phits/(cycle $\cdot$ node) and average latency of packets, measured in cycles. A simulation run consists of choosing a load to be offered in every router, which is translated in the probability of creating a packet each cycle for each compute node. Then, it begins with about 10,000 cycles of warmup to hopefully arrive a stationary state, where more simulation time does not change fundamentally the state of the network. This is followed by another 10,000 cycles of statistics where the measures are taken.

The default mode of FSIN is to generate packets synthetically following a traffic pattern. Nevertheless, it also has the capability of reading MPI traces, generating packets in the same order and same waits than the application did when it was traced, including computing times. This preserves causal dependencies between messages, although it cannot represent the behaviour that the application would have if packets arrive in different order. For a trace, the most interesting measure is the total time of execution.

The version of FSIN used in this thesis is a local branch deriving from a version of April 5th, 2011, which is still the more recent available from `http://www.sc.ehu.es/ccwbayes/members/jnavaridas/home/simul.html`. From that version, many changes have been made, among which are:

- Bug fixes, miscellaneous features and scripts.
- Option to map several applications to the same router (either traces or synthetic traffic).
- Option to expand the collectives found in traces into point to point messages.
- Several placement functions to be used in the mapping of applications.
- Capability of synthetic traffics to receive parameters.
- A new simulation mode for the simulation of bursts.
- Allow to set different values of buffer size and latency in each dimension. This is motivated by the differences between local and global links in dragonfly networks.
- A basic deadlock/livelock detector, which works correctly for oblivious routings and can be useful sometimes with adaptive routing.
- Allow topologies to have any number of dimensions, with a maximum defined at compile time (`MAX_DIM`).

- Injection ports have been differentiated from their virtual channels.
- Option to make several router arbitrations each cycle.
- Option to arbitrate output ports instead of virtual channels.
- Option to use wormhole routing instead of virtual cut-through.
- New traffic patterns have been implemented:
  - **localuniform:** with a parameter a. Consists of $\frac{a}{100}$ local traffic plus $(1 - \frac{a}{100})$ uniform traffic.
  - **antipodal:** Each node generate traffic to a node in the most distant router. Also known as *furthest-node pairing*.
  - **fixedrandom:** At the beginning of the simulation each node selects a different random node. During the simulation, a node sends packets to its selected destination.
  - **randompairing:** At the beginning the set of compute nodes is partitioned in pairs in a random uniform way. During the simulation, each node communicates with its pair.
  - **centralsymmetric:** Must be defined a center of symmetry for each topology. Each node communicates with its symmetric respect to the selected center. If possible, it uses as center of symmetry the center of the network, or equivalently by $(-1/2, ..., -1/2)$. It is the immediate generalization of the traffic called *diagonal pairing* in [CEH+12].
  - **interval**, with an argument a: The node x sends packets to a random node in $x + 1, 2, \ldots, a$.
  - **corners:** In a mesh each node sends traffic to a random corner. This traffic has proved to be the most prone to deadlock in meshes with wormhole.
  - **row** with two arguments X and Y: Each node sends traffic to a random router in one of the rows from $currentrow + X$ to $currentrow + X + Y - 1$. For dragonflies and $Y = 1$ this traffic is more known as ADV+X.
- Several new topologies have been implemented:
  - **Canonical dragonfly:** Including the global link arrangements: dally, palmtree, coset (a symmetric one) and random.
  - **Lattice graphs:** The generator matrix is introduced as argument.
  - **Hamming graphs:** Although they are lattice graphs, they have enough particularities to have a specific implementation. They include the complete graphs.
  - **extdgfly_A_B_T:** A dragonfly with A routers per group, B groups and T links between every pair of groups.
- New injection modes:
  - **route:** First route, then select injection buffer depending on your next direction.
  - **route adaptive:** As above if there are packets to all destinations. Otherwise randomizes. It adapts to traffic, avoiding HoLB in uniform and congestion on permutations.

5.2 NPB MPI traces

This section studies several applications included in the NAS Parallel Benchmarks (NPB) [BHS+95] and derives the communication pattern from an analysis of the calls to MPI communication primitives. NPB is a set of benchmarks (kernels and pseudo-applications) based on scientific applications typically employed in large parallel systems. Thus, we can expect that the results of this section can be generalized to standard HPC applications. These applications have been largely studied in the scientific literature [Rie06, LH, Lee09, KL98]. This section will focus on their communications, considering separately point-to-point and collective messages and ignoring management messages (Init, rank, size, etc). The reason for this is that, while broadcast or all-to-all messages can contribute to a significant part of the network traffic (or all of it, such as in FT), they do not reflect any relevant communication topology, and their behaviour typically mimics the theoretical topological properties of the underlying physical topology. Thus, our topological study will remark the topology defined by point-to-point primitives and the amount of collective communications, if present.

The five kernels and their communication characteristics are:

IS Integer Sort: Each process communicates with the following one, forming a directed cycle. There is a large amount of broadcast/gather from the root process.

EP Embarrassingly Parallel: There is almost no communication, and all of it is composed of collectives: barriers and allreduce. This program is known for not being typically affected by network bottlenecks.

CG Conjugate Gradient: The dataset is a square matrix with data blocks regularly allocated to processes. It is depicted in the middle of Figure 5.1. Starting from that mesh of processes:

- Each process communicates with the process in the same row with a single-bit change in their address (Hamming distance 1); this is, the nodes in each row form an hypercube. This comprises most messages from the application. They are the red lines in the figure.
- Each node communicates with the transpose in the mesh. If the process matrix is rectangular with aspect ratio 2:1, the transpose is applied to consecutive pairs of processes. These are the blue lines in the figure.

MG MultiGrid V-cycle: 3D torus with some broadcast/gather messages from the root node. Figure 5.2 shows in the left a 3D cube and in the right what the communication becomes after mapping it in a 2D mesh.

FT Fourier Transform: All messages are collectives (mainly broadcast), using barrier-based synchronization between different execution phases.

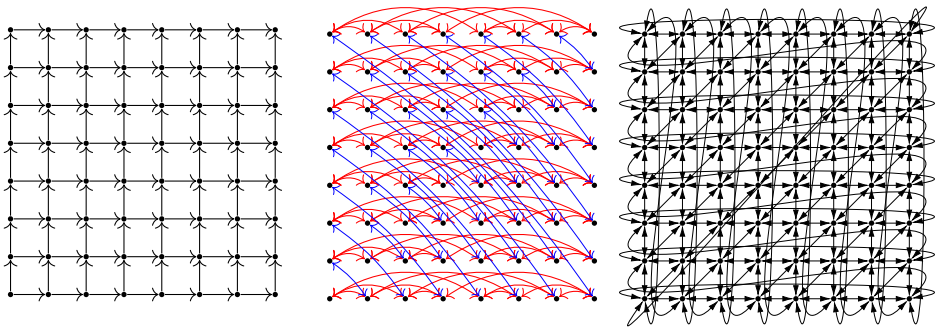


Figure 5.1: Local communications in LU, CG and BT.

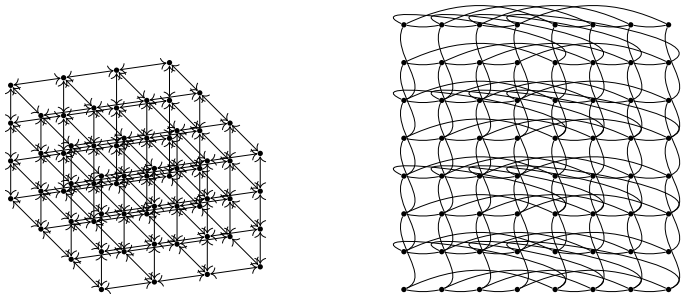


Figure 5.2: Local communication in MG.

There are three pseudo-applications which implement different Computational Fluid Dynamics (CFD) solvers:

BT Block tridiagonal: The logical topology is an square 2D torus with additional diagonal links $\pm(-1, 1)$. There is much more data in one direction than in the opposite, but using the same amount of messages. The right part of Figure 5.1 shows this communication.

SP Pentadiagonal Solver: Same communication graph as BT.

LU LU Solver: Uses a typical wavefront communication mechanism, in which processes are allocated in a 2D mesh and each one communicates to the one on the right and below it. This is depicted in the left of Figure 5.1.

Finally, there is a miscellaneous benchmark with unconstructed computation:

DT Data Traffic: A network traffic benchmark using traffic patterns *Black Hole* (BH), *White Hole* (WH) or Shuffle (SH). The BH model will be the one used.

From the previous list, and considering the amount of data sent on each message, it can be observed that the most frequent communication graphs are 2D or 3D mesh or torus. Generally, this happens because they correspond to the data layout employed with a direct block-assignment partitioning algorithm and near-neighbour communication. The additional edges in the communication graph correspond to application-specific patterns such as the hypercubes in CG, which can be variable in time (because they only happen in certain phases) and send a variable amount of data.

Regarding collective communications, the most frequent primitives are broadcast/scatter from a root node or gather/reduce to this root node, used to distribute tasks among nodes and compile the results. Also, all-to-all is used when each node needs the data from all others in each step.

The Extrae MPI tracing tool [Ext] has been utilized to obtain traces from these benchmarks using problem sizes A, running on 32, 64 and 128 nodes of the Altamira supercomputer based on IBM JS21 blades. The obtained trace files are very large; a total of 84GB of traces have been collected. Some applications require a square number of processes, which limits the traces that have been collected.

When used in simulations, a mapping algorithm must map the trace processes into the simulated nodes. The measure of interest is the execution time (in cycles) of the parallel section from the call to MPI_Init to the call to MPI_Finalize. Simulations will show that the network load differs a lot among NPB. EP and LU have very few network load while that for FT and IS have a considerable load.

5.3 Evaluation of the Impact of Symmetry in the Performance of 2D Lattice Networks

This section explores the effect of symmetry on the performance of lattice graphs. In Subsection 5.3.1 it is given a formula for the throughput of a lattice network depending on a measure of its symmetry. Later, in Subsection 5.3.2, the previous model is verified for 2D lattice graphs and a detailed comparison is done for a selection of networks.

5.3.1 A Simple Performance Model for Networks Based on Lattice Graphs

In this subsection it is introduced a simple performance model based just on two topological parameters: symmetry and average distance. Symmetric 2D lattice graphs are characterized in Appendix A in terms of their generator matrices, showing that the most significant examples of symmetric 2D lattice graphs are Gaussian networks and Kronecker products of cycles. Now, the concept link utilization is introduced for these networks, which measures how symmetric they are. The *link utilization* (LU) is defined as the average usage of the network links under uniform traffic at maximum load.

Since the adjacency pattern of a lattice graph is determined by orthogonal vectors $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$, the network links can be separated into n disjoint sets, each containing the links in the corresponding direction. Therefore, in these networks, the distance of any minimal path between two nodes $\mathbf{v}$ and $\mathbf{w}$ can be decomposed as $D(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^{n} D_i(\mathbf{v}, \mathbf{w})$, where $D_i(\mathbf{v}, \mathbf{w})$ is the distance in the $\mathbf{e}_i$ direction when a packet travels through a shortest path between nodes $\mathbf{v}$ and $\mathbf{w}$. Furthermore, the average distance in $\mathbf{e}_i$ (that is, the average number of $\mathbf{e}_1$ links in the route of packets under uniform traffic) can be defined as

$$\bar{k}_i = \frac{1}{|V(G)| - 1} \sum_{\mathbf{v} \in V(G)} D_i(0, \mathbf{v}).$$

In addition, it is clear that $\bar{k} = \sum_i \bar{k}_i$. Then the average distance of the longest dimension is denoted as

$$\bar{k}_{\max} = \max\{\bar{k}_i \mid i \in \{1, \ldots, n\}\}.$$

In [CMV+10], it is proposed that the link utilization is related to the average distances as follows:

$$LU = \frac{1}{n}\frac{\bar{k}}{\bar{k}_{\max}}.$$

Note that symmetric networks have $\bar{k}_i = \frac{\bar{k}}{n}$ for any i, which implies $LU = 1$, or what is the same, there is a full use of all the links. It is interesting to remark that the maximum throughput achieved by a network under uniform traffic directly depends on the average use of links (LU) and inversely on the network average distance, $\bar{k}$. A random packet traveling between two network nodes has to traverse $\bar{k}$ links at a rate that depends on LU. Hence, in general, the maximum throughput can be computed as

$$l = LU\frac{\Delta}{\bar{k}} = \frac{2}{\bar{k}_{\max}},$$

which generalizes Equation (1.1) to non edge-balanced topologies.

5.3.2 Empirical Performance Evaluation of the Symmetry of 2D Lattice Networks

In this subsection an experimental evaluation is carried out to empirically measure the performance of 2D lattice networks. The simulations have been done for many 2D lattice networks of 360 nodes, a number in which there are a lot of different symmetric and non symmetric 2D lattice networks. Moreover, all the simulations will use uniform traffic, to preserve symmetry. The routing is performed by a uniform selection of the route among all the paths with minimum length. This is of great importance even using minimal routing. If it is not done with such uniformity, there is a risk of breaking symmetry.

In Figure 5.3, the maximum accepted load of the set of the 360 node 2D lattice networks under consideration is represented. In abscissa, the plotted values are of $\frac{1}{\bar{k}_{\max}} = 2\frac{LU}{\bar{k}}$. In ordinates, the throughput is plotted, measured as packets consumed by node per cycle. As can be seen, the empirical values match with the performance model. The accepted load or throughput is summarized by the $\bar{k}_{\max}$ parameter. Therefore, the throughput of a 2D lattice network only depends on its $\bar{k}_{\max}$. Hence, among all the possible 2D lattice networks with the same number of nodes, it would be preferably to choose those with $\bar{k}_1 \approx \bar{k}_2$. The best case $\bar{k}_1 = \bar{k}_2$ is only obtained when the lattice graph is completely symmetric, that is, when it is also edge-transitive. In this sense, the link utilization gives a measure of how symmetric the network is.

Next, it will be evaluated how symmetry and average distance affect network performance for the set of 360 node 2D lattice networks evaluated in the previous experiment. In Figure 5.4, for each network, the average distance in abscissa and the link utilization in ordinates is represented. The symmetric graphs are the ones at the top of the figure and the non-symmetric ones lie nearly on a curve. The best networks in terms of performance are represented by points on the upper left corner. Conversely, the worst networks are represented by points in the bottom right corner. Each line represents networks with the same throughput, where the throughput grows from one line to the next with the growing velocity captured in the density of the lines. The results which are surrounded with a circle correspond to some 2D lattice networks which are going to be considered next with more detail.

Figure 5.5, contains examples of the four networks marked with a circle in Figure 5.4. Two of them have the same throughput, one being symmetric and the other very asymmetric.

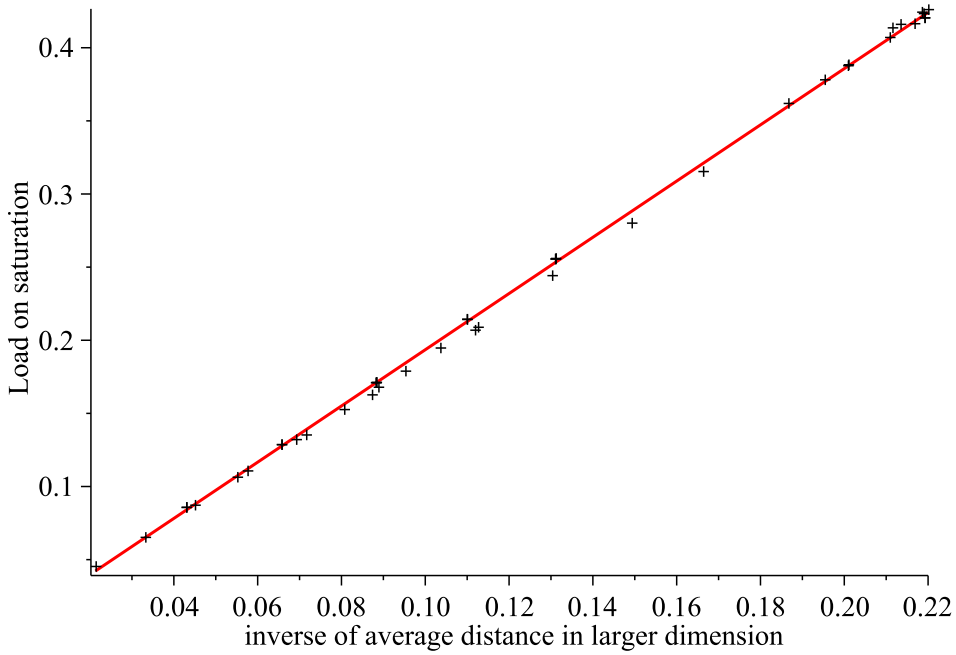


Figure 5.3: Load on saturation measured for empirical values of $\frac{1}{k_{\max}} = 2\frac{LU}{\bar{k}}$ of a wide set of 2D lattice networks of 360 nodes.

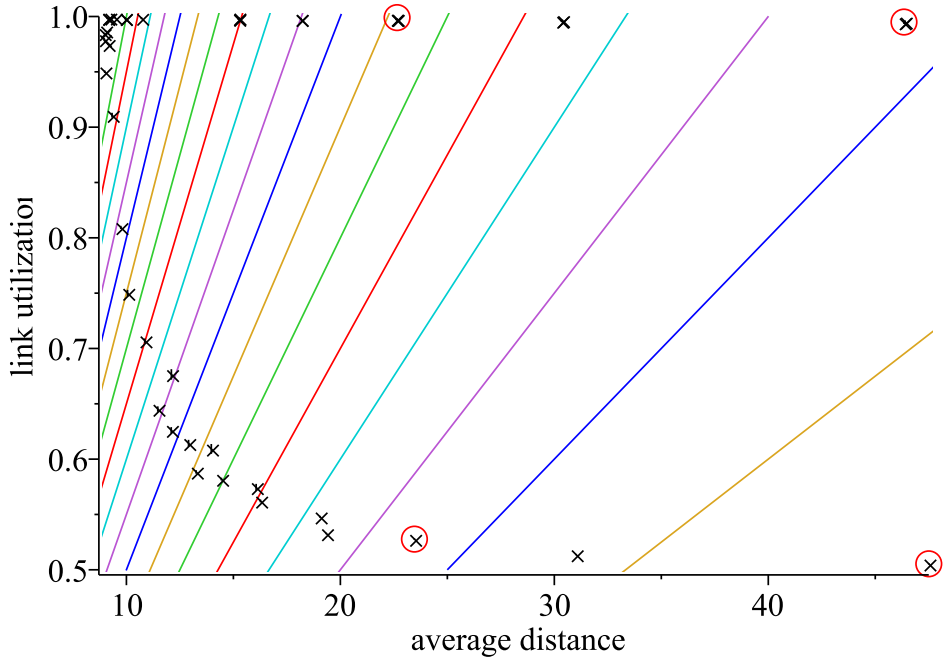


Figure 5.4: Average distance and link utilization of 2D lattice networks of 360 nodes.

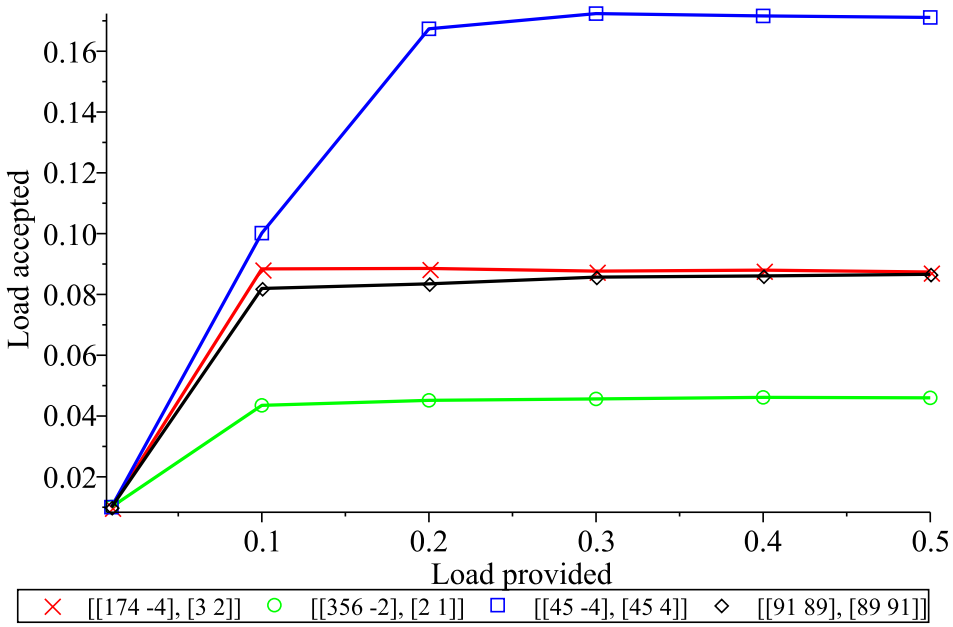


Figure 5.5: Throughput for some the symmetric 2D lattice networks of 360 nodes

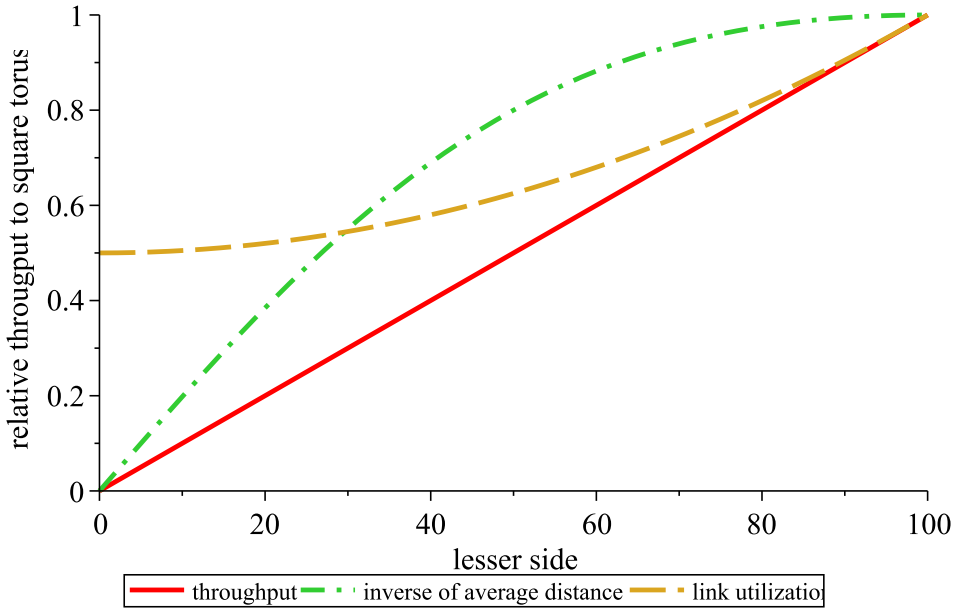


Figure 5.6: Throughput from rectangular torus of 100^2 nodes normalized by the square torus. Separating throughput into link utilization and inverse of $\bar{k}_{\max}$ as in $l \approx \frac{1}{\bar{k}_{\max}} = LU\frac{1}{\bar{k}}$.

The other two are a symmetric one with an average distance similar to the asymmetric one, and finally an asymmetric one with average distance similar to the symmetric one. None of them are good in terms of performance, but they are a clear example of how symmetry and distance impact separately on performance. The selected networks are:

i) $\mathcal{G}(\begin{pmatrix} 91 & 89 \\ 89 & 91 \end{pmatrix})$, with $LU = 1$ and $\bar{k} = 45.131$.

ii) $\mathcal{G}(\begin{pmatrix} 174 & -4 \\ 3 & 2 \end{pmatrix})$, with $LU = 0.527$ and $\bar{k} = 23.317$.

iii) $\mathcal{G}(\begin{pmatrix} 45 & -4 \\ 45 & 4 \end{pmatrix})$, with $LU = 1$ and $\bar{k} = 22.618$.

iv) $\mathcal{G}(\begin{pmatrix} 356 & -2 \\ 2 & 1 \end{pmatrix})$, with $LU = 0.505$ and $\bar{k} = 45.376$.

Networks i) and ii) provide similar throughput because the product of LU by the inverse of the average distance in each network is approximately the same. Network iii) is the best one as it exhibits symmetry and lower average distance. Finally, as expected, Network iv) is the one with lowest performance.

Finally, it is considered an important case of 2D lattice networks: rectangular tori of constant size $N = |\det(M)| = xy$, $x \geq y$. As stated at the beginning of the present section, mixed-radix tori are of great interest for practical reasons. Here, the average distances per dimension are, approximately, $\bar{k}_1 \approx \frac{x}{4}$, $\bar{k}_2 \approx \frac{y}{4}$, thus $\bar{k}_{\max} = \bar{k}_1$, $\bar{k} = \bar{k}_1 + \bar{k}_2 \approx \frac{x+y}{4}$ and therefore, the link utilization is

$$LU = \frac{1}{2}\frac{\bar{k}}{\bar{k}_1} \approx \frac{x+y}{2x}.$$

Since the throughput grows proportionally with the link utilization and inversely with the average distance, follows that

$$l \approx \frac{1}{\bar{k}} LU = \frac{2}{x} = \frac{2y}{N}.$$

Therefore, given a network of size N, the torus with best performance which can be built is the one whose sides have lengths as close as possible. In the case that it is possible, the square torus, which is a completely symmetric graph, is the one showing the best performance.

In Figure 5.6, the relative throughput of a rectangular torus against the square torus of the same size can be seen. All the considered networks have 10,000 nodes. Since that $l \approx \Delta \bar{k}^{-1} LU$, it is possible to decompose the throughput into two factors: the LU factor gives us the performance related to the symmetry, which goes from $\frac{1}{2}$ in the worst case of the completely asymmetric torus degenerated into a cycle, to 1 when the network is the completely symmetric square torus. The other factor, $\bar{k}^{-1}$, gives the performance related to distance. As can be observed, it grows faster when the sides are different, but slower when it is similar to the square tori. The product of the two curves gives us the line that records the performance improvements. Just to finish, it should be remarked that Kronecker products of cycles are clearly superior to tori (Cartesian product of cycles) when used in mixed-radix topologies, as proved in [CMV+10].

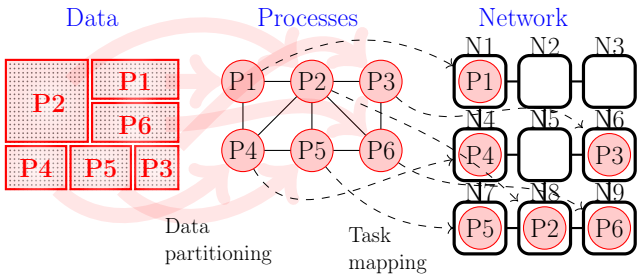


Figure 5.7: Data partitioning and task mapping.

5.4 Mapping Applications on Lattice Graphs

Twisted tori are variants of the torus topology in which a twist is applied to the peripheral links in one or more dimensions [BBK+68, Seq81, YFJ+01]. Different variants of 2D and 3D twisted tori have been studied in the past [CMV+10, CMV+07, VMMB11]. Rectangular tori and meshes are often built for practical reasons of packaging and modularity. The focus of this section is on the Rectangular Twisted Torus (RTT), which is a twisted version of the 2D Rectangular Torus (RT) topology. Its peripheral twist modifies the distance properties of the base topology, reducing the diameter, average distance, and more importantly, balancing the use of the network links in different dimensions. As a consequence, it can achieve a 50% increase in network throughput under uniform traffic.

Traffic from real applications behaves according to the nature of parallel algorithms and depends on the allocation of each logical task in the network and on their communication requirements. The assignment of work to system nodes is a two-step process. First, *data partitioning* divides the program dataset into multiple groups of data to be operated in parallel by each process. The second step is *task mapping*, which assigns each of the processes to an individual computation node. Both steps are illustrated in Figure 5.7.

Depending on the application, processes are arranged according to a certain *logical topology* (or *communication graph*), which reflects the communication pattern between them. The logical topology depends on the data partitioning employed, which is largely dependant on the data structures and the algorithm used by the application. Modifying the data partitioning mechanism to fit the underlying physical topology is generally considered very difficult since it implies modifying the algorithm. On the contrary, task mapping considers the *logical topology* of the application and the *physical topology* of the system to provide an efficient solution that preserves the communications locality as much as possible. Task mapping is a graph embedding problem, in which a guest graph (the logical topology) must be accommodated to a host graph (the physical topology) minimizing an objective cost function such as byte-hop [ASK06], maximum dilation or average dilation [YCM06]. Task mapping is an NP-complete problem [Bok81, KN84], but multiple heuristic mechanisms have been deployed to provide acceptable results [Bha11].

Concentration is a technique typically employed in HPC to reduce system cost and increase scalability. A concentrated system connects multiple computation nodes to a single (higher-radix) switch. This has been routinely employed in fat trees, but also in direct topologies. One example is the Gordon Supercomputer [NS10] which employs a 3D concentrated torus using commodity Infiniband technology. The use of concentration adds a new dimension to the mapping problem, which also needs to consider which logical tasks are concentrated into the same network node.

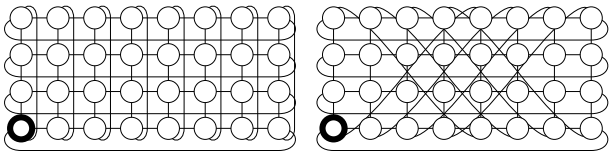


Figure 5.8: RT(4) and RTT(4)

This section presents a comprehensive analysis of the RTT topology under realistic conditions that considers the mappings of HPC applications and the use of concentration. In this way, although twisting is apparently less amenable for task mapping, we can show how the topological advantages of RTTs translates in execution time reductions by choosing the adequate mapping technique. Specifically, the main contributions of this section are the following:

i) An analytical model that estimates the expected performance of both topologies with different mapping algorithms, according to the amount of local and global communications in the system and observing that RTTs should outperform RTs in many scenarios. It is presented the counterintuitive result that when mapping mapping tasks into the RTT, using "twists" in the concentration function, despite not allocating neighbor tasks to the same network node, helps to improve performance.

ii) The theoretical model is validated, considering both standard and concentrated versions of each topology, by simulating synthetic traffic with local and global communications.

iii) Finally, the performance in a real scenario is approached by simulating several benchmarks from the NAS Parallel Benchmark (NPB 3.2) suite [BHS+95].

The topologies considered in this section are the rectangular torus of sides a and $2a$, RT(a), and the rectangular twisted torus RTT(a) (Figure 5.8). Both are lattice-graphs, given by

$$RT(a) = \mathcal{G}(\begin{pmatrix} 2a & 0 \\ 0 & a \end{pmatrix}) \text{ and}$$
$$RTT(a) = \mathcal{G}(\begin{pmatrix} 2a & a \\ 0 & a \end{pmatrix}).$$

It will be required to have a definition of the rectangular mesh $m \times n$:

$$R_{m,n} = \{(x, y) \in \mathbb{Z}^2 \mid 0 \leq x \leq m-1, 0 \leq y \leq n-1\}.$$

Note that both RT(a) and RTT(a) can be seen as a graph over $R_{2a,a}$.

Table 5.1 summarizes approximated distance-properties of both topologies, where the diameter is denoted k, and the average distance $\bar{k}$. We also include the average distance per dimension, so that $\bar{k} = \bar{k}_1 + \bar{k}_2$. Note that RTTs have better distance properties than RTs for the same number of nodes. However, a topological difference with more impact on performance is symmetry. RTTs and RTs are node-symmetric topologies, *i.e.* any node can *observe* the same local environment. Nevertheless, RTs are not edge-symmetric graphs since horizontal links are not equivalent to the vertical ones. For example, note that in a RT(a) horizontal links form cycles of length $2a$ and the vertical links form cycles of

Topology	k	$\bar{k}$	$\bar{k}_2$	$\bar{k}_1$
RT(a)	$\frac{3a}{2}$	$\frac{3a}{4}$	$\frac{a}{2}$	$\frac{a}{4}$
RTT(a)	a	$\frac{2a}{3}$	$\frac{a}{3}$	$\frac{a}{3}$

Table 5.1: Topology distance properties of RT and RTT [CMV+10].

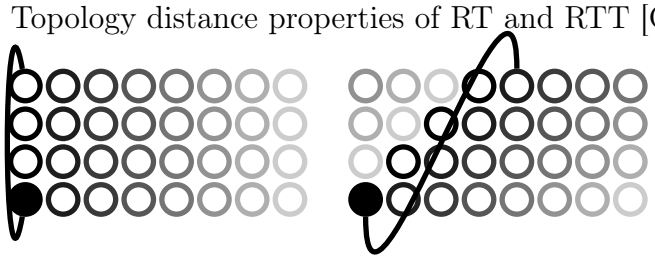


Figure 5.9: Identity and diagonal-shift mapping functions on RT(4).

length a. On the contrary, RTTs are completely symmetric, since the twist in the vertical dimension makes all links locally equivalent [CMB13].

The rest of this section is organized as follows. Subsection 5.4.1 considers the problem of mapping applications, presenting a performance model. Subsection 5.4.2 details the experimental environment and evaluates both topologies with different mapping and concentration functions using the NPB benchmarks. Finally, Subsection 5.4.3 concludes the section.

5.4.1 Task Mapping in Rectangular and Twisted Torus

As presented in the introduction, different mapping and concentration functions will be studied to determine how they impact performance. Many scientific applications rely on structured grid communication patterns which employ both local (near-neighbour) and global communication [Col04, ABC+06]. The study across this section is restricted to 2D topologies for both the application communication pattern (meshes or tori) and the network topology (RT and RTT). The mapping of meshes into both RT and RTT is simple, since the peripheral links do not have an impact on the adjacency of the mesh. Therefore, the focus is on the mapping of 2D torus into RT and RTT. Communication graphs will have the same number of vertices as the physical topology, or a multiple value when using concentration.

The *mapping function* maps each process (or a set of concentrated processes) from the logical topology into one physical network node. Two mapping functions will be considered, depicted in Figure 5.9: the *identity* function id maps the processes grid directly into the internal mesh to preserve internal adjacency; by contrast, the *diagonal-shift* f^d introduces an internal incremental twist to compensate the twisted peripheral links in RTT. The mapping f^d is inspired by the mappings considered for double loop networks in [CS11]. These mapping functions are formally defined for logical and physical topologies of the same size, a rectangle $R_{m,n}$, as follows:

$$id(x, y) = (x, y)$$
$$f^d(x, y) = (\text{rem}(x + y, m), y)$$

In concentrated networks, a *concentrating function* determines which processes are

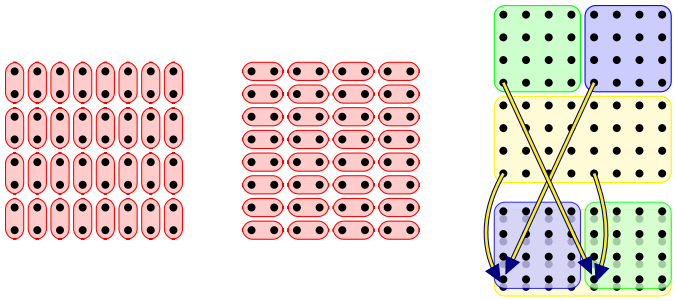


Figure 5.10: Concentration functions $f^v_{c=2}$, $f^h_{c=2}$ and $f^t_{c=2}$ on a 8×8 mesh.

placed on the same network node prior to the mapping function. Let $R_1 = R_{pm,qn}$ and $R_2 = R_{m,n}$ be two rectangles with $m, n, p, q \in \mathbb{N}$. A concentrating function of concentration $c = pq$ sends c processes from R_1 to the same node in R_2. There are several concentration and mapping functions that preserve the communication locality from the logical graph. Specifically, the *horizontal union* f^h_c, the *vertical union* f^v_c and the *twisted union* f^t_c are defined as:

$$f^h_c(x,y) = \left(\left\lfloor \frac{x}{c} \right\rfloor, y\right), \ q = 1$$
$$f^v_c(x,y) = \left(x, \left\lfloor \frac{y}{c} \right\rfloor\right), \ p = 1$$
$$f^t_c(x,y) = \left(\mathrm{rem}(x + \left\lfloor \frac{y}{n} \right\rfloor \frac{m}{2}, m), y - \left\lfloor \frac{y}{n} \right\rfloor n\right)$$

Figure 5.10 represents the three concentration functions with $c = 2$ applied to a 8×8 mesh. Note how the twisted concentration function f^t does not concentrate neighbor nodes; rather, it is designed to compensate for the twist in the peripheral links of RTTs, by preserving adjacency in the logical topology when the identity mapping is employed. Different concentration functions can be combined for $c > 2$. The symbol $\circ$ will denote function composition, *i.e.*, $(f \circ g)(x,y) = f(g(x,y))$.

The next subsections will study the relative performance obtained with each topology (RT and RTT) using the different mapping and concentration functions presented here. In first place, a model will be introduced that considers both the amount of local and global messages that are sent in a generic application. Next, considering this model, expressions are determined for the expected performance with a given topology and mapping in terms of base latency and maximum accepted throughput. Finally, these performance values are calculated for logical torus mapped into RT and into RTT with different combinations of mapping and concentration functions.

Modelling a Generic Application

Now, a simple model for the communications of an application will be considered. Our model considers a variable rate of local and collective (global) communications in the application graph. The greek letter α will denote the proportion of local (l) messages, and $(1 - \alpha)$ the proportion of messages corresponding to collective communications (global, g), assuming the same message size.

Local traffic communicates each process with one of its (up to) four direct neighbours in the application graph, which will be a 2D mesh or torus. Depending on the mapping and

concentration functions, these neighbor nodes could be mapped far away in the physical topology. Collectives communicate a given process (or each of them) with a set (or all) of the other processes in the system. While the behavior of local communications is dependant on the mapping algorithm, it will be assumed that global communications can be averaged as uniform traffic, whose performance only depends on the physical topology. Our model does not consider the less frequent point-to-point messages sent to remote (not neighbour) nodes.

Performance modelling

The performance of a network depends on which metric is most restrictive during the execution of an application. Specifically, both maximum accepted throughput and average latency will be considered. In general, average (base) latency, or average latency at zero load, depends on the average distance of the topology while maximum (base) latency depends on its diameter. Actual latencies in the network will depend on the base latency plus the network contention which is not considered in our simple model.

Under uniform traffic, the maximum throughput in an asymmetric torus depends on the maximum *average distance per dimension* [CMV+10], since longer dimensions will typically saturate earlier. However, when certain mapping and concentration functions are considered, the links in a given dimension may not receive all the same load. One such case are the peripheral links of the RTT when the *id* mapping is employed. In such cases, it is the subset of links that receives the highest load which determines the maximum throughput.

Here the theoretical performance of RT and RTT is studied for applications in which network performance is limited by either throughput or communication latency, considering different mapping and concentration functions and a variable rate of local and global traffic.

Latency Estimation The average distance travelled by the packets in the network will be denoted by τ. Assuming a constant link latency in the network, τ is an indicator of the base latency in the network. τ differs from the topological average distance $\bar{k}$, since τ depends on the communication pattern and the mapping and concentration functions. Note that τ can be divided into two terms, considering the contribution of local and global traffic:

$$\tau = \tau^l + \tau^g = \alpha \cdot d + (1 - \alpha) \cdot \bar{k},$$

where $\tau^l = \alpha \cdot d$ represents the contribution from local messages and depends on the *average dilation* of the mapping function, d. Dilation, that is network distance of adjacent processes in the logical topology, has been employed as an objective function in mapping algorithms. However, in our model it does not directly determine the performance, since the global communications are not affected by the mapping algorithm. Interestingly, τ^g only depends on the physical topology, with its overall value being determined by the average distance, $\bar{k}$, in Table 5.1.

Throughput Estimation Let l be the number of phits sent per cycle by each of the N nodes in the network. Let E be the set of edges (links) in the graph, and $|E|$ its cardinal. If all the links in the network were used in a balanced way, the maximum throughput of

the network (in phits/cycle) would be calculated as:

$$N \cdot l \cdot \tau \leq 2|E| \Rightarrow l \leq \frac{2|E|}{N \cdot \tau}.$$

For example, when all nodes communicate with their four direct neighbours in the network ($\tau = 1$), up to $l = 4$ phits/(node·cycle) can be accepted, since $|E| = 2N$ in both RT and RTT.

By contrast, when the load on different links of the network differs, the set of links that receives the highest load will saturate first and become the bottleneck that limits the maximum throughput accepted by the network. Let $E_1, E_2, \ldots, E_s$ be all the possible different sets of links in the network and τ_j be the average distance that packets traverse across links in E_j. Then, the maximum throughput in the network will be given by

$$max_l = \frac{2}{N} \min \left\{ \frac{|E_j|}{\tau_{E_j}} \right\}, \ E_j \subseteq E.$$

This calculation is valid as long as all nodes are limited by the subsets selected, which happens in all cases considered in this section. For example, it was shown in [CMV+10] that under uniform traffic $\tau_{max} = \max\{\tau_h, \tau_v\}$ serves to calculate the maximum throughput in RTs and RTTs, where $\tau = \tau_h + \tau_v$ represents the division of the average distance on the horizontal and vertical dimensions. This is true because all the links in a dimension (horizontal or vertical) are used in the same proportion under uniform traffic. However, when a mapping algorithm and local traffic are taken into account, the internal and peripheral links within a given dimension can receive different loads. In such case, the maximum throughput is determined by the subset of links receiving the higher load. Therefore, in order to estimate the maximum throughput, it is required to determine the subset of links that will saturate first. Specifically, E_{hi} and E_{vi} will denote the sets of horizontal and vertical internal links, and E_{hp} and E_{vp} the peripheral ones. In the RT and RTT $|E_{hi}| = 2a^2 - a$, $|E_{vi}| = 2a^2 - 2a$, $|E_{hp}| = a$ and $|E_{vp}| = 2a$.

As before, τ_{E_j} can be divided in its local and global components: $\tau_{E_j} = \tau^l_{E_j} + \tau^g_{E_j} = \alpha \cdot \bar{k}_{E_j} + (1 - \alpha)\bar{k}^g_j$, where $\bar{k}_{E_j}$ represents the average number of hops of local packets in E_j. In our model the global communications are approximated by uniform traffic, so $\bar{k}_j$ is independent of the mapping, and can be derived from the values given in Table 5.1 and the specific $|E_j|$. For example, if $E_j = E_{hj}$, then $\bar{k}_{E_j} = \bar{k}_1 \cdot |E_{hj}|/|E| = \bar{k}_1 \cdot \frac{2a^2-a}{2a^2} = \bar{k}_1 \cdot \left(1 - \frac{1}{2a}\right)$.

The next subsections will calculate the expected latency and maximum throughput of applications mapped into RT and RTT. First, standard topologies will be considered, followed by the case of concentration $c = 2$.

Mapping 2D Logical Tori into Standard RT and RTT

Now the previous model is applied to estimate the latency and maximum throughput when the logical topology is a 2D ($2a \times a$) torus with the same number of tasks as nodes in the network. When the physical topology is a RT, the *id* mapping is the only one that makes sense, since f^d would otherwise break the locality. In the RTT, they will be considered both *id* and f^d.

***id* Mapping of 2D Tori into RT** In this case the communication graph coincides with RT(a), so with the mapping function *id* the locality is preserved and the dilation is $d = 1$.

Global traffic follows a uniform distribution with $\bar{k} = \frac{3a}{4}$, so base latency can be calculated from

$$\tau = \tau^l + \tau^g = \alpha \cdot d + (1-\alpha) \cdot \bar{k} = \alpha + (1-\alpha) \cdot \frac{3a}{4}.$$

To determine the maximum throughput the sets of horizontal and vertical links, E_h and E_v are considered. The average distance of local traffic is $(0.5, 0.5)$. As a consequence, $\tau_h^l = \tau_v^l = 0.5\alpha$. Table 5.1 provides the average distances of global traffic in each dimension, so:

$$\tau_h^g = (1-\alpha)\bar{k}_1 = \frac{a}{2}(1-\alpha) \text{ and}$$
$$\tau_v^g = (1-\alpha)\bar{k}_2 = \frac{a}{4}(1-\alpha).$$

Horizontal links E_h are the ones which first saturate with $\tau_h = \frac{1}{2}\alpha + \frac{a}{2}(1-\alpha)$, and the maximum throughput is

$$max_l = \frac{2}{N}\frac{|E_h|}{\tau_h} = \frac{2}{2a^2}\frac{2a^2}{\frac{1}{2}\alpha + \frac{a}{2}(1-\alpha)} = \frac{4}{\alpha + a(1-\alpha)}.$$

This expression shows that under local traffic ($\alpha = 1$) up to 4 phits/(node·cycle) can be accepted, since communication occurs with the four direct neighbors on independent links. Under uniform traffic, the maximum load will be $\frac{4}{a}$.

id **Mapping of 2D Tori into RTT** In this case the locality of the application is broken. The internal and horizontal peripheral links preserve locality. By contrast, peripheral vertical communications which would follow the path $(0, 1)$ in the logical graph are transformed into routes $(0, -(a-1))$ in the physical network (and reciprocally for peripheral hops $(0, -1)$), with maximum dilation $(a-1)$. This happens in a fraction $1/a$ of the vertical local messages in the network, so the average dilation is $d = \frac{1}{2a} \cdot (a-1) + \frac{2a-1}{2a} \cdot 1 = \frac{3}{2} - \frac{1}{a}$. Using the value of $\bar{k}$ from Table 5.1 it is obtained that

$$\tau = \tau^l + \tau^g = \alpha \cdot d + (1-\alpha) \cdot \bar{k} = \left(\frac{3}{2} - \frac{1}{a}\right)\alpha + (1-\alpha)\frac{2a}{3}.$$

Now, some calculation will show which dimension determines maximum throughput. The local load on horizontal links is the same as in the case of *id* mapping on RT, so using the value $\bar{k}_1 = \frac{a}{3}$ it is obtained that

$$max_l \leq \frac{2}{N}\frac{|E_h|}{\tau_h} = \frac{4}{\alpha + \frac{2a}{3}(1-\alpha)}.$$

On vertical links, all the local traffic is sent on the internal links E_{vi}. Similarly to the dilation calculation, the average distance of local traffic on E_{vi} will be $\tau_{vi}^l = (1-\frac{1}{a})\frac{1}{2}\alpha + \frac{1}{a}\frac{(a-1)}{2}\alpha = (1-\frac{1}{a})\alpha$.

Global traffic uses every vertical link equally, which implies that $\tau_{vi}^g = \bar{k}_2\frac{|E_{vi}|}{|E_v|}(1-\alpha) = \frac{a}{3}(1-\frac{1}{a})(1-\alpha)$. With these values one can determine that vertical links impose a lower limit on maximum throughput than horizontal links:

$$max_l = \frac{2}{N}\frac{|E_{vi}|}{\tau_{vi}} = \frac{2}{N}\frac{|E_{vi}|}{(1-\frac{1}{a})\alpha + \frac{a}{3}(1-\frac{1}{a})(1-\alpha)} = \frac{4}{2\alpha + \frac{2a}{3}(1-\alpha)}.$$

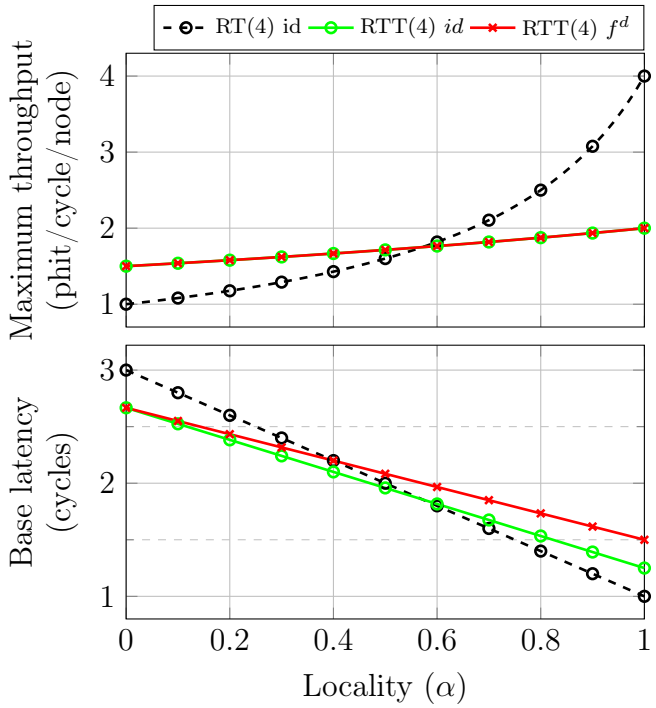


Figure 5.11: Maximum throughput and latency for logical torus mapped on RT(4) and RTT(4)

f^d Mapping of 2D Tori into RTT Using the mapping f^d with the RTT, horizontal locality is preserved and vertical locality suffers dilation 2, with local traffic on vertical links requiring routes $\pm(1,1)$. Average dilation is $3/2$. The distances for local traffic are

$$\tau_h^l = \alpha, \qquad \tau_v^l = \frac{\alpha}{2}, \text{ and} \qquad \tau^l = \frac{3\alpha}{2}.$$

From the values from Table 5.1 follows that $\tau = \tau^l + \tau^g = \frac{3\alpha}{2} + (1-\alpha)\cdot\frac{2a}{3}$ To calculate maximum throughput observe that distances on the horizontal dimension are longer $\tau_h = \tau_h^l + \tau_h^g = \alpha + \frac{a}{3}(1-\alpha)$. Then it is obtained that

$$max_l = \frac{2}{N}\frac{|E_h|}{\tau_h} = \frac{4}{2\alpha + \frac{2a}{3}(1-\alpha)}.$$

Figure 5.11 shows the throughput and latency results when mapping a 2D logical torus into RT and RTT. The identity mapping *id* provides the best results in RTT. With this mapping, the RTT achieves better throughput and latency when global communications dominate with up to 50% throughput improvements. However, the *id* mapping in RTT has a maximum dilation of $a-1$. The diagonal-shift mapping f^d minimizes the maximum dilation on the RTT(4) and it obtains the same throughput as *id* but worse average latency. Both curves intersect in $\frac{a}{a+3}$, which tends to 1 for larger networks. As a consecuence, the RTT will obtain better throughput than RT except for traffic with high locality (α).

Mapping 2D Logical Tori into Concentrated RT and RTT

Now, the case of an application with more processes than routers in the physical topology will be considered. The calculations will be restricted to an application whose local

communication graph is a square torus $2a \times 2a$ and concentration $c = 2$ has to be employed. Cases with larger concentration can be calculated similarly. Two concentration functions can be employed to reduce the vertical dimension, f_2^v and f_2^t. Then, any mapping can be applied, leaving 4 possibilities: f_2^v, $f^d \circ f_2^v$, f_2^t, $f^d \circ f_2^t$, but the latter is omitted because both the concentration f_2^t and mapping f^d are designed to cope with twisted peripheral links. In the RT, f_2^v is the only sensible combination, since it exploits the maximum locality. Their performance is studied next.

f_2^v Concentration of 2D Tori into RT In this case locality is preserved similarly to the *id* mapping in RT, with two vertical neighbour processes mapped into the same network node. From every 8 local communications from each node, 2 are internal to the node, 2 imply a vertical hop and 4 imply a horizontal jump:

$$\tau_h^l = \frac{4}{8}\alpha = \frac{\alpha}{2}, \qquad \tau_v^l = \frac{2}{8}\alpha = \frac{\alpha}{4}, \text{ and} \qquad \tau^l = \frac{3\alpha}{4}.$$

Average dilation is $d = 3/4$, lower than 1 thanks to neighbor nodes being concentrated together. With respect to global traffic, the values in Table 5.1 remain approximately valid when using concentration, so it follows the estimation of average latency:

$$\tau = \tau^l + \tau^g = \frac{3\alpha}{4} + \frac{3a}{4}(1-\alpha).$$

Regarding throughput, it is straightforward that horizontal links are saturated first since both local and global average distances are larger in X than in Y. The same result is obtained as in RT without concentration:

$$\tau_h = \tau_h^l + \tau_h^g = \frac{\alpha}{2} + \bar{k}_1(1-\alpha) = \frac{\alpha}{2} + \frac{a}{2}(1-\alpha), \text{ and}$$

$$max_l = \frac{2}{N}\frac{|E_h|}{\tau_h} = \frac{4}{\alpha + a(1-\alpha)}.$$

f_2^v Concentration of 2D Tori into RTT Again, this case is similar to the *id* mapping in RTT without concentration, with locality preserved in the internal mesh but not in the vertical peripheral links. Now, of each 8 local communications of each node of the first and last rows ($\frac{2}{a}$ from total of rows) 2 are internal to the network node, 4 are $(\pm 1, 0)$, 1 is $(0, \pm 1)$ and 1 $(0, \pm(a-1))$. Therefore, vertical peripheral links are not used by local communications. The nodes in the internal rows $(1 - \frac{2}{a})$ send 4 messages to $(\pm 1, 0)$ and 2 to $(0, \pm 1)$ from each 8 messages. Then, average local distances are

$$\begin{aligned}
\tau_h^l &= \frac{4}{8}\alpha = \frac{1}{2}\alpha, \\
\tau_{vi}^l &= \frac{a}{8}\frac{2}{a}\alpha + \frac{2}{8}\left(1 - \frac{2}{a}\right)\alpha = \frac{a-1}{2a}\alpha, \text{ and} \\
\tau^l &= \tau_h^l + \tau_v i^l = \left(1 - \frac{1}{2a}\right)\alpha.
\end{aligned}$$

Using the global values the average distance in the network can be calculated:

$$\tau = \tau^l + \tau^g = \left(1 - \frac{1}{2a}\right)\alpha + (1-\alpha)\frac{2a}{3}.$$

Local distances are larger in X than in Y, and global distances are balanced in the RTT, so the throughput will be determined by distances in horizontal links. With the value of $\bar{k}_1$ from Table 5.1 is obtained that

$$max_l = \frac{2}{N}\frac{|E_h|}{\tau_h} = \frac{4}{\alpha + \frac{2a}{3}(1-\alpha)}.$$

$f^d \circ f_2^v$ Mapping of 2D Tori into RTT In this case horizontal locality is preserved but vertical locality is modified: two vertical neighbours are mapped into each node, so half of the vertical local messages are internal to the node. The remaining vertical communications suffer dilation 2, similar to the case of the f^d mapping in RTT without concentration. From each 8 local messages of each network node, 6 use horizontal links and 2 use vertical links, so average distances are

$$\tau_h^l = \frac{6}{8}\alpha = \frac{3}{4}\alpha, \text{ and} \qquad \tau_v^l = \frac{2}{8}\alpha = \frac{1}{4}\alpha.$$

Then, $\tau^l = \alpha$ and base latency will be determined by

$$\tau = \alpha + \frac{2a}{3}(1-\alpha).$$

Thus, the network throughput is limited by horizontal traffic by

$$max_l = \frac{2}{N}\frac{|E_h|}{\tau_h} = \frac{2}{\frac{3}{4}\alpha + \frac{a}{3}(1-\alpha)} = \frac{4}{\frac{3}{2}\alpha + \frac{2a}{3}(1-\alpha)}.$$

f_2^t Mapping of 2D Tori into RTT In this case, the f_2^t mapping preserves the neighborhood in the original task graph, but no neighbours are collocated in the same node. The dilation of the network is $d = 1$ in both dimensions, so $\tau_h^l = \tau_v^l = \frac{\alpha}{2}, \tau = \alpha$. The average distance will be determined by

$$\tau = \alpha + (1-\alpha)\frac{2a}{3}.$$

Traffic is balanced, so any dimension is the throughput limiter. Global traffic is $\tau_h^g = \tau_v^g = \frac{a}{3}(1-\alpha)$. Then, the maximum network throughput will be

$$max_l = \frac{2}{N}\frac{|E_h|}{\tau_h} = \frac{2}{\frac{\alpha}{2} + \frac{a}{3}(1-\alpha)} = \frac{4}{\alpha + \frac{2a}{3}(1-\alpha)}.$$

Throughput and average latency results with $c = 2$ are presented in Figure 5.12. In the RTT both the twisted f_2^t or vertical f_2^v concentrations (with *id* mapping) obtain the best results. The latency is better in the latter case, since the vertical concentration puts neighbor processes together, reducing the amount of local communications in the network. However, the f_2^t concentration obtains maximum dilation 1, while in f_2^v it is $a - 1$. Interestingly, both concentrations on the RTT obtain better throughput than the RT for any locality value, and the average base latency is similar in all cases (slightly better for uniform traffic and slightly worse for local traffic).

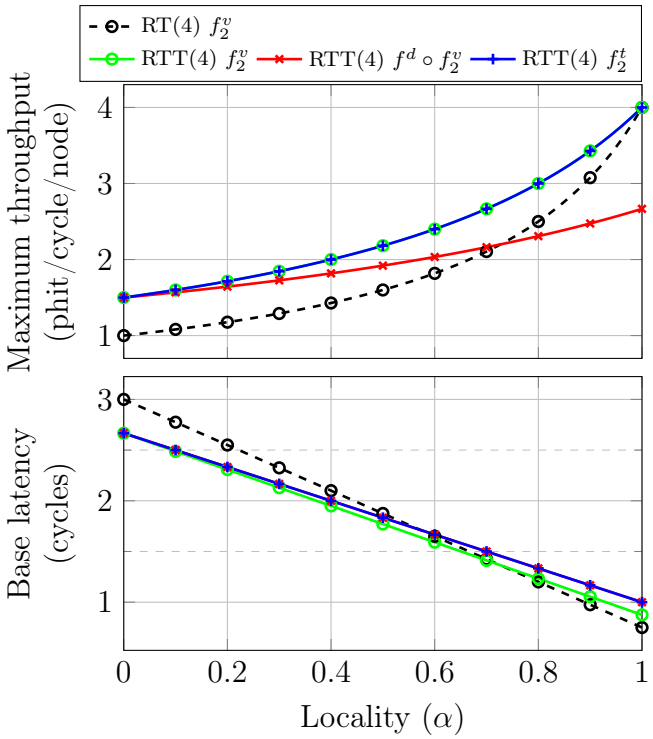


Figure 5.12: Maximum throughput and latency for logical torus mapped on RT(4) and RTT(4) with concentration c=2.

5.4.2 Performance Evaluation

The previous subsection presented an analytical study showing that the RTT can be competitive against the RT. However, it did not consider the impact of other factors such as maximum dilation, remote communications or the network load. In this subsection different RT and RTT configurations using synthetic traffic and real applications from the NPB suite [BHS+95] will be evaluated. It is organized in several parts. First, the traffic model is described, followed by the configuration of the simulator. Then there are two parts for the actual simulation: one for simulation with synthetic traffic and the other for trace-based simulation.

Workloads

In first place, it is presented a simulation of independent traffic sources under random traffic. In this case, a ratio $1-\alpha$ of packets are distributed evenly along the whole network, while a ratio α of packets is sent to neighbor nodes. The inter-injection interval at each node is random following a Poisson distribution chosen as to modulate the provided load in terms of phits/cycle/node. Some parallel applications exhibit traffic patterns in which nodes communicate with their nearest neighbors in a torus topology. This can be either due to the inherent symmetry of the application or because of mapping big data matrices on the network nodes. For that reason, nearest-neighbor (NN) communication patterns are included in this study.

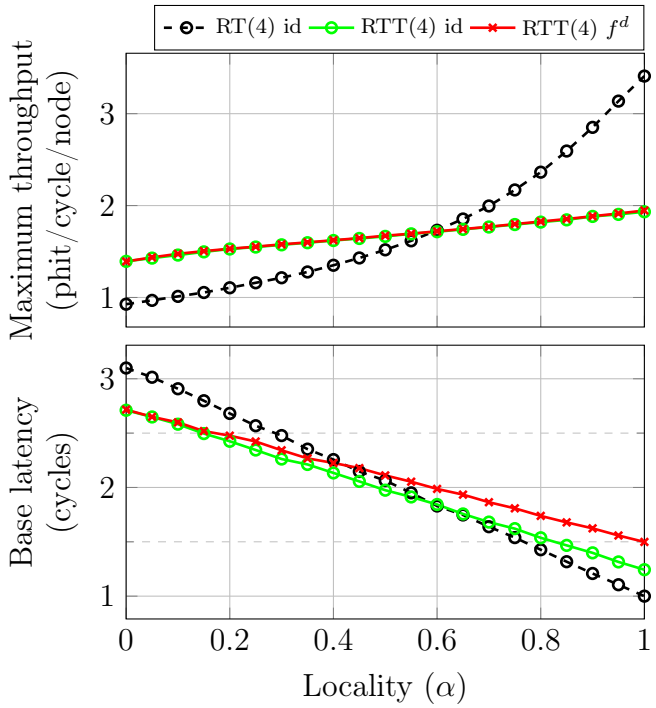


Figure 5.13: Simulation results for latency and maximum throughput for logical torus mapped on RT(4) and RTT(4).

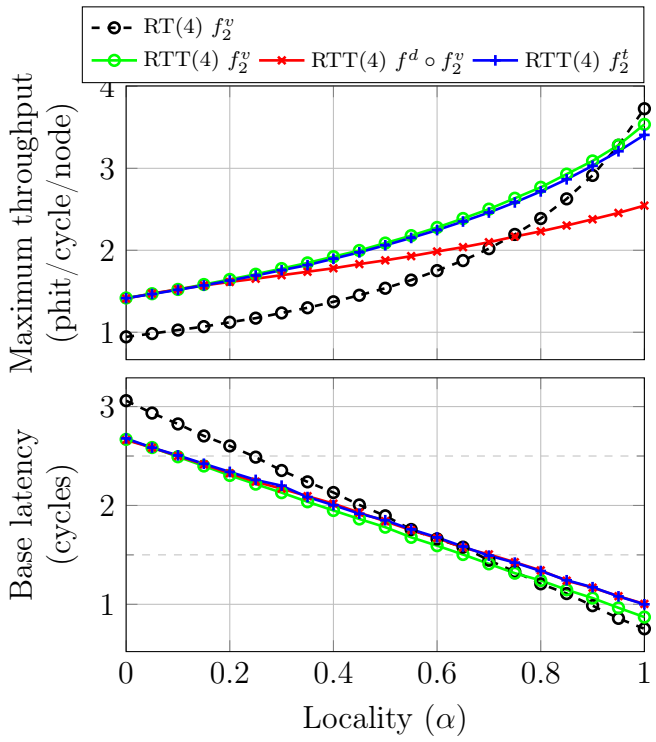


Figure 5.14: Simulation of base latency and maximum throughput for logical torus mapped on RT(4) and RTT(4) with concentration c=2.

Processor Frequency	2 GHz	Virtual Channels	3
Phit size	4 bytes	Routing Mechanisms	adaptive
Packet size	64 phits	Arbitration mechanims	random
Link speed	1 Gbps	Deadlock avoidance	bubble

Table 5.2: Simulation parameters used in experiments about mapping of applications.

Simulation Configuration

Now, a simulation is done of the benchmarks with different sizes and concentration levels employing the FSIN simulator with the parameters presented in Table 5.2. For this study, the router employed is similar to the one implemented in the IBM Blue Gene/L: virtual cut-through switching strategy [KK79], and bubble flow control deadlock avoidance [ABC$^+$05], with an static virtual channel plus two fully adaptive virtual ones. Blue Gene family of supercomputers implements a congestion control mechanism that prioritizes in-transit traffic against new injections, which is also implemented in our router. In our experiments, packets have a fixed length of 64 phits of 4 bytes each.

Synthetic Traffic

Now, the analytical model presented in Subsection 5.4.1 is corroborated with simulation results using synthetic traffic. A synthetic traffic is modelled in which each process communicates with one of his neighbors with probability α. With probability $1 - \alpha$ the packet is sent to a random process, not necessarily one of the four neighbors. The application is mapped to the physical network with different concentrations (1 or 2 processes per node) and different mapping functions.

When measuring the maximum throughput of the network, 4 injectors are used, similar to the Blue Gene/Q chips [CEH$^+$12], and packets with a length of 64 phits. Multiple injectors are required to saturate the network with local traffic, since α close to 1 can provide throughput up to 4 phits/(node$\cdot$ cycle). When measuring minimum latency, a load of 0.01 packets is injected per node per cycle, each of length 1 phit. In this way, the delays due to network congestion and packet consumption time are eliminated, allowing us to measure the minimum average latency to transit the network.

The results of a 16×8 toroidal application over a 16×8 network are shown in Figure 5.13. Both throughput and latency results are really close to the ones predicted by the analytical model shown in Figure 5.11.

The same results for a network with concentration ($c = 2$) are shown in Figure 5.14. In this case, a 16×16 toroidal application is mapped onto a 16×8 network. For both throughput and minimum latency, results are very similar to the ones predicted in Figure 5.12 with the analytical model.

Real Applications Traffic

For a more realistic analysis, the traces of the NPB benchmarks showed in Section 5.2 have been simulated.

In all cases the performance is evaluated with all the mapping algorithms studied, considering all the logical tasks as an array of consecutive nodes.

The usage of the network limits the maximum performance differences between configurations. The traffic load of each application can be easily measured by simulation.

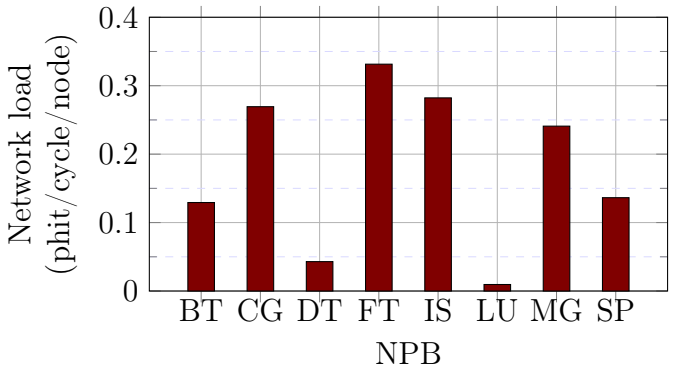


Figure 5.15: Network load for 64 tasks mapped onto a RT(4) with $c = 2$

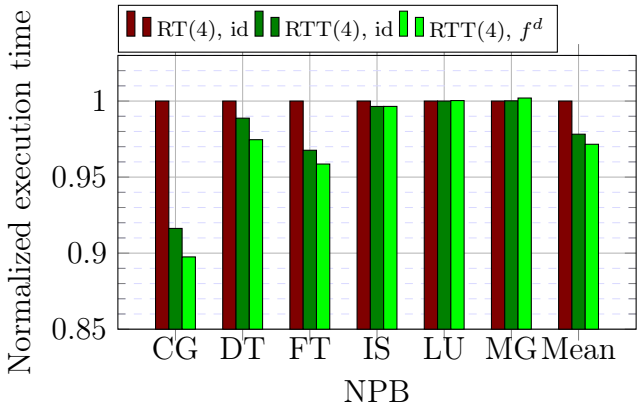


Figure 5.16: Execution time for 32 processors mapped onto a RT(4) or RTT(4) with $c = 1$

Figure 5.15 shows the load measured when running on different networks with 64 tasks and concentration $c = 2$. Results for 32 or 128 tasks are similar. It is observed that CG, FT, IS and MG are the applications with the highest network load. Therefore, the theoretical throughput results obtained by our model should be applicable to them.

Besides the network load, the performance can be limited by latency when there are multiple dependency chains among messages. Thus, a low average network load does not imply that the network is irrelevant: it can be either inactive (this occurs in the EP benchmark, omitted for this reason) or limited by latency. This is the case, for example, of DT: although having a low throughput (below 5%), the performance increase obtained by the RTT is above this 5%. Specifically, DT is a data traffic benchmark with large amounts of messages sent between nodes according to a certain pattern (black hole was employed), what introduces a large amount of dependencies in the traffic traces.

In general BT, LU, and SP use mainly near-neighbor communications (α close to 1), while DT, EP, FT and IS use more global communications (α closer to 0). The communication in CG occurs between certain pairs of nodes, not necessarily neighbors (remote messages). Finally, the consecutive node labelling and task mapping on a 2D network does not preserve adjacency of the 3D torus of MG.

First, the non-concentrated scenario is considered. Figure 5.16 shows the performance obtained when running NAS benchmarks of 32 processes on RT(4) and RTT(4). The

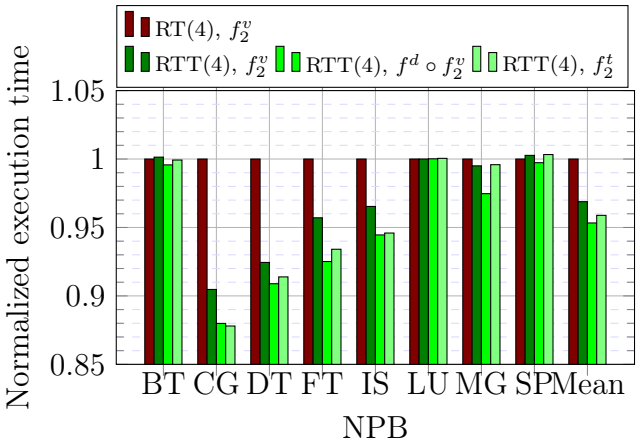


Figure 5.17: Execution time for 64 processes mapped onto a RT(4) or RTT(4) with $c = 2$

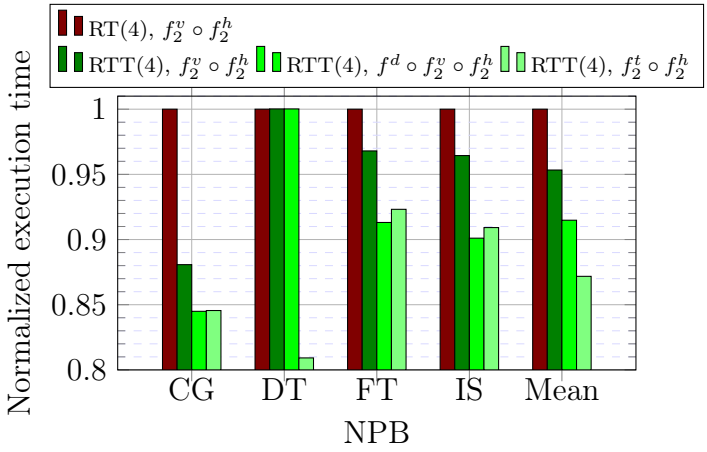


Figure 5.18: Execution time for 128 processes mapped onto a RT(4) or RTT(4) with concentration $c = 4$

simulation is restricted to those benchmarks that allow a number of processes which is not a square number. It is observed that RTT always performs equal or better than the RT counterpart except just a slight loss in one case; CG and FT are the applications in which the performance improvement is higher, saving up to 10% of the execution time. Not surprisingly, these applications contain a large amount of global communications. LU and MG are the ones with the worse performance, without any improvement or even a slight loss of less than 0.1%. On average, the use of the RTT improves execution time in 2.2%, and f^d behaves slightly better than id for the RTT.

Next, several concentration techniques are evaluated. Figure 5.17 shows the execution time of NAS benchmarks running with 64 processes, mapped onto a RT(4) or RTT(4) with two compute nodes per network router. On average the RTT outperforms the RT. Interestingly, the f_2^t concentration function, which does not concentrate neighbor tasks, provides one of the best results thanks to the arrangement of tasks in relation to peripheral links, similar to $f^d \circ f_2^v$. Note that BT and SP, which employ 2D logical torus, do not vary significantly from the base case when using an RTT. On average, the applications running on the RTT save between 3.0% and 4.6% of the overall execution time.

Finally, Figure 5.18 presents the results of non-square applications with 128 tasks

(8×16) running on a 4×8 network[1]. The combinations employed for concentration 4 are $f_2^v \circ f_2^h$ and $f_2^t \circ f_2^h$. Again, the RTT outperforms RT. The execution time savings of the RTT range from 8.7% (FT) to 19.1% (CG) using the best combination. On average, the best performance is obtained with the $id \circ f_2^t \circ f_2^h$ combination for the RTT, with an speedup of 12.8%.

The results are coherent with our analytical model. DT, FT and IS employ a large amount of collective communications, translating into a larger performance on RTT, especially on concentrated networks. CG sends remote messages and is also benefited by the distance reduction in the RTT. By contrast, BT, LU and SP employ local communications, and thus the performance on RTT is similar to RT. Regarding concentrated networks, it is observed that in RTT the use of either the diagonal-shift mapping f^d or the twisted concentration f_2^t improves performance.

5.4.3 Conclusions

This section makes a first exploration to mapping functions for rectangular torus topologies with peripheral twists. As it has been proved, in non concentrated topologies the performance gain depends on the local traffic amount. On the other hand, with concentration RT always improves performance if the mapping technique is correctly chosen. Particularly, it has been done a theoretical study that shows that simple mapping algorithms obtain the maximum performance, with speedups ranging from -10% to 50% depending on the locality of the communications and the application topology. When concentrated tori are employed, proper concentration and mapping functions prevent this performance loss by compensating the effect of the twisted peripheral links.

Those numbers reflect the performance of applications bounded by the network. However, real applications alternate computation and different communication patterns on different phases. When simulating real applications (from the NPB benchmarks) the topological advantages of the RTT translate to average performance gains of 2.2-13.2% depending on the specific configuration.

5.5 Evaluation of Lattice Graphs Compared to Topologies of Current Supercomputers

Most evaluations of big networks have relied on measuring their behavior when managing synthetic traffic loads. Typical experiments are based on simulation. Notwithstanding, the work presented in [CEH+12] evaluates different routing algorithms reporting maximum achievable loads on a real IBM Blue Gene/Q system. They make runs on machines whose topologies are the tori $T(8,8,8,4,2)$ and $T(16,8,8,8,2)$. The last dimension of size 2 will be ignored and treat the networks as four dimensional ones; the last small dimension comes from the inside of computing nodes, fixed by computer technology. The simulated networks are the same tori plus symmetric lattice graphs of the same sizes. Thus, in the evaluation, $4D\text{-}BCC(4)$ is compared to $T(8,8,8,4)$ and $4D\text{-}FCC(8)$ compared to $T(16,8,8,8)$.

The torus $T(8,8,8,4)$ contains 2048 vertices, has diameter 14 and an average distance of 7.0. On the other hand, $4D\text{-}BCC(4)$ has the same number of vertices, but it has diameter 8 and average distance 6.1. In addition the torus is not symmetric while the body-centered

[1]LU and MG, are omitted because of technical difficulties. Results should be similar to the ones presented in Figure 5.16.

Injectors	6
Packet size	16 phits
Queues	4 packets
Deadlock avoidance	Bubble
Virtual Channels	3
flow control	Virtual Cut-through
Routing Mechanisms	DOR
Arbitration mechanism	random

Table 5.3: Simulation parameters used in experiments in the evaluation of lattice graphs.

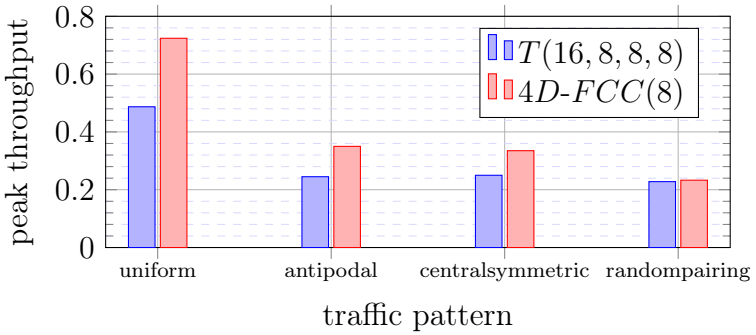


Figure 5.19: Throughput peak in $T(16,8,8,8)$ and $4D\text{-}FCC(8)$ under several synthetic traffics.

is, thus it is expected an increase of more than $7.0/6.1 = 1.15$ for uniform loads. For the large size, $T(16,8,8,8)$ contains 8192 vertices with diameter 20 and average distance 10.0; while $4D\text{-}FCC(8)$ has diameter 16 and average distance 8.8. In this subsection simulation results show that better distance properties plus symmetry translate into a better performance of symmetric networks.

The synthetic traffic patterns used are some of the used in [CEH+12]: uniform, antipodal, centralsymmetric and randompairings. Simulation parameters are shown in Table 5.3. The simulation starts with a network warmup, followed by 100,000 cycles in which statistics are collected. At least 5 simulations are averaged for each point.

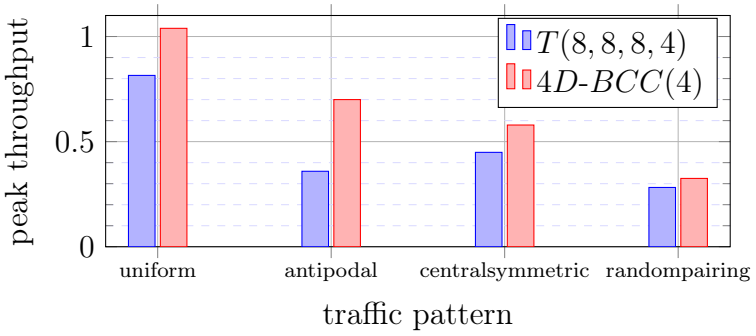


Figure 5.20: Throughput peak in $T(8,8,8,4)$ and $4D\text{-}BCC(4)$ under several synthetic traffics.

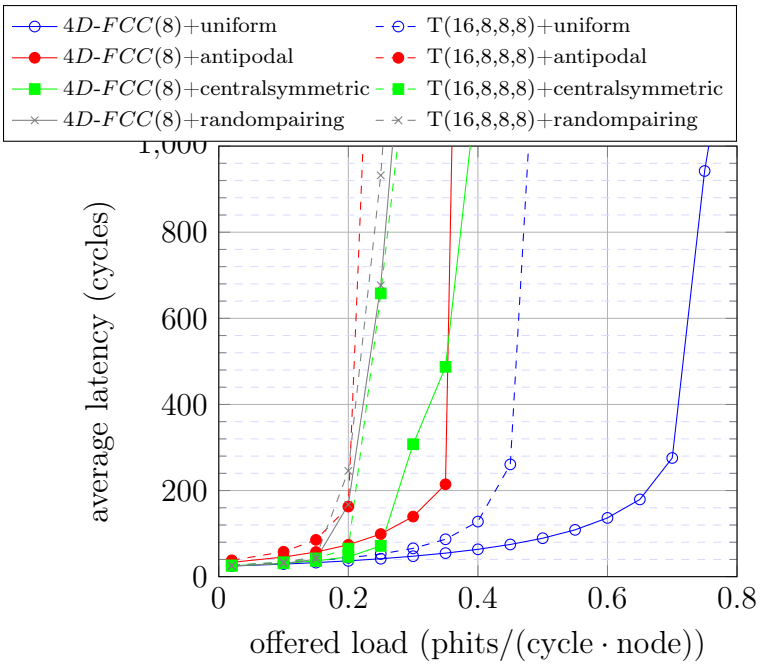


Figure 5.21: Packet delays in T(16,8,8,8) and 4*D-FCC*(8) under several synthetic traffics.

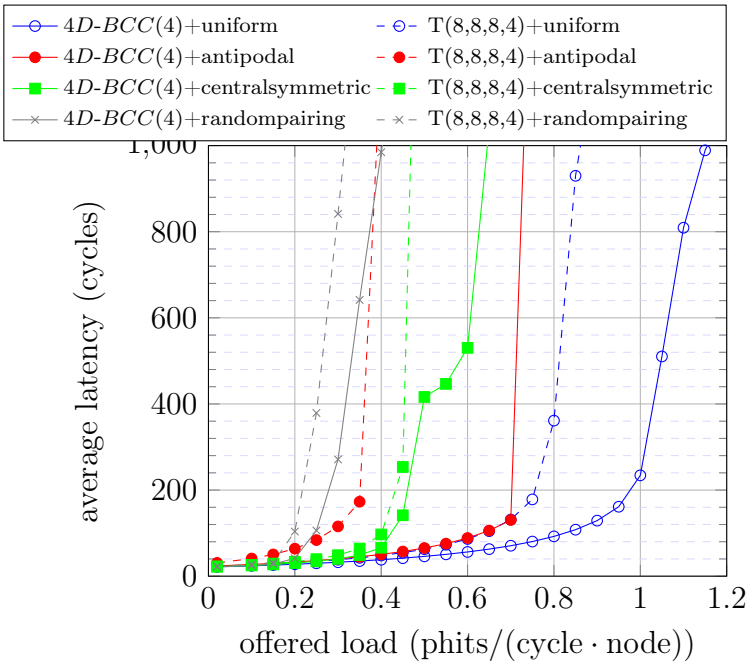


Figure 5.22: Packet delays in T(8,8,8,4) and 4*D-BCC*(4) under several synthetic traffics.

Figures 5.19 and 5.20 show results of accepted load in the four networks. Under uniform traffic, results exhibit gains of 27% in the small case (4*D-BCC*) and 49% in the large one (4*D-FCC*). In random pairings, the throughput is consistently higher, with gains of 15% and 2% respectively. The other two traffic patterns show congestion at high loads for all the networks considered. Nevertheless, the peak throughput for the antipodal traffic improves by 95% and 43% respectively. Under central symmetric traffic, gains are 29%

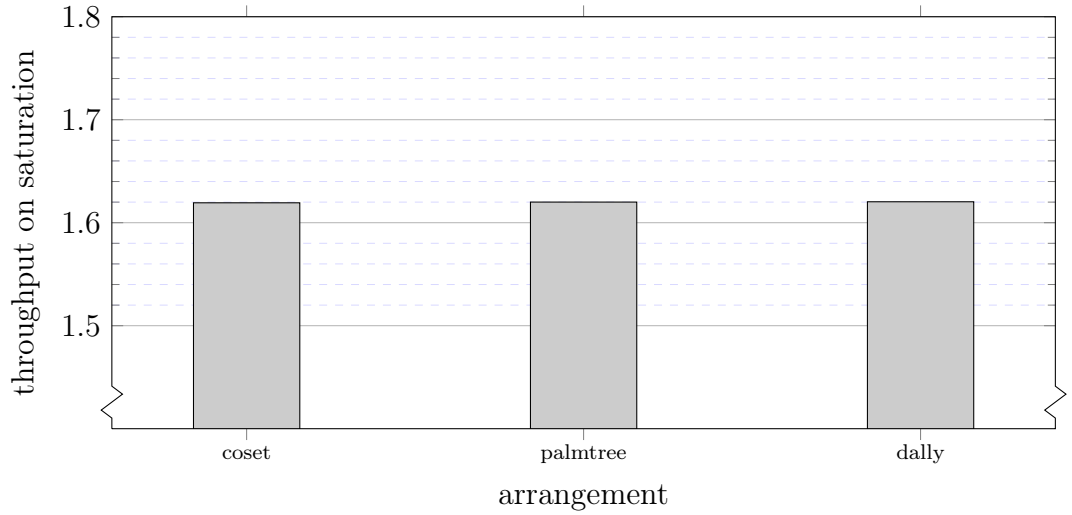


Figure 5.23: The null effect of symmetry on dragonflies of 9 groups.

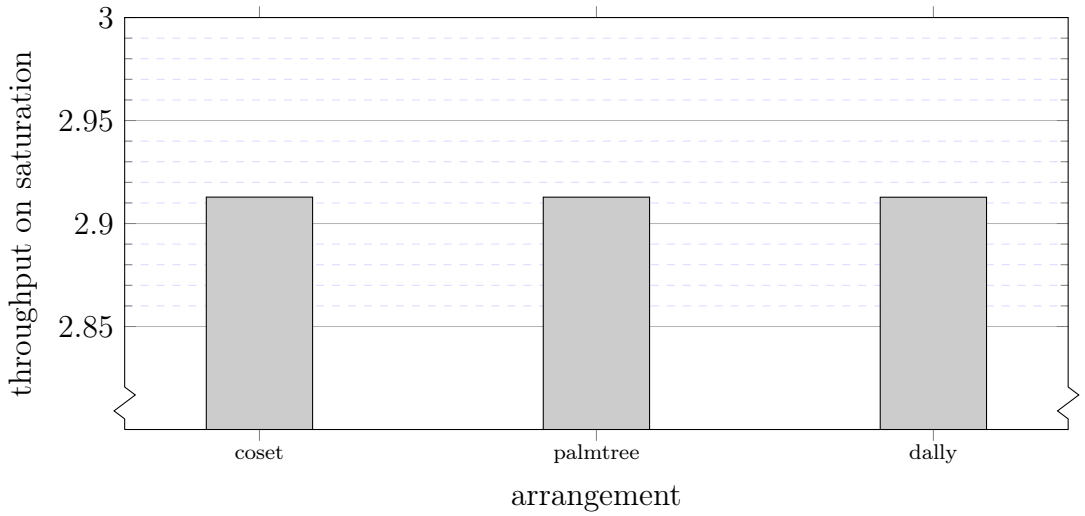


Figure 5.24: The null effect of symmetry on dragonflies of 73 groups.

in the small case and 34% in the large one. Figures 5.21 and 5.22 show average packet latencies. The different curves demonstrate the superior behavior of lattice topologies. Gain values are coherent with the topological analysis presented in Section 2.2.

5.6 Evaluation of the Symmetry in Dragonflies

In previous sections it has been shown that symmetry has a lot of impact in the performance of lattice graphs. This section studies the same question for the case of dragonflies. Note that there is never (except with very large trunking) an automorphism between local edges and global edges, so they are never edge-transitive. As seen in Section 3.4.1, there are global link arrangements for which the resulting dragonfly is vertex-transitive, while others give no symmetry at all, or some middle case between them. Among the vertex-transitive it is possible that there are also automorphisms that map a global edge to any other

Group size	$a = 24$ routers, 312 nodes	Link latency (local/global)	10/100 cycles
Number of groups	$b = 79$ groups	Packet size	8 phits
Network size	24,648 computing nodes	Buffer capacity(local/global)	32/256 phits
Router size (ports)	$R = 49$	Virtual channels per port	3 (except Valiant, 4)
	($\Delta_0 = 13$, $\Delta_1 = 23$, $\Delta_2 = 13$)	Switching policy	Virtual Cut-Through
Global link arrangement	Extended palmtree	Router arbitration	Random
Router model	input-queued	Router speedup	No

Table 5.4: Simulation parameters used in the evaluation of the routing in dragonfly networks.

global edge, and local edges to any other local edge, which would be a relaxed form of edge-transitivity. Nevertheless, expansion properties do not depend almost of the arrangement. The diameter is always 3 and the average distance can vary very little. Thus, in a asymmetric dragonfly, although there are no automorphism between edges, the load is quite well distributed among all edges of the same type; and among all the edges if the balancing condition (3.5) is hold.

Figures 5.23 and 5.24 shows the throughput of dragonflies with arrangements of different levels of symmetry for uniform traffic. The differences are lesser than the random variations in a simulation. The first one for dragonflies with 9 groups of 4 routers and the second one for dragonflies with 73 groups of 12 routers. From these results it is clear that symmetry is not required for dragonflies to have good performance, although it can allow for special mechanism as the one seen in Section 3.6.

5.7 Evaluation of the Deadlock-free Adaptive Routing for Dragonflies with Global Trunking

This section shows the performance of the routing mechanisms proposed in Chapter 3. The simulated network has input-buffered routers with $a = 24$ routers per group and global trunking $t = 4$ using the *extended palmtree* arrangement. The number of groups has been rounded to $b = 79$, what provides a balanced topology according to equation (3.4), requiring routers with $R = 49$ ports and leading to a total of 24,648 computing nodes. The complete set of parameters is presented in Table 5.4 and the routing mechanisms characteristics in Table 5.5. The following oblivious routing mechanisms have been implemented:

- *Minimal*: Hierarchical routing first to the destination group and then to the destination router, as described in [KDSA08]. The global link of the path is selected as follows: if the source router has a direct link to the destination group, select it; otherwise, select an available link to any random router with a direct link to the destination. This mechanism is the reference for uniform traffic, although it only exploits 2/1 VCs, therefore suffering from more HoLB.

- *Valiant* [Val82]: Nonminimal routing composed of two parts: Minimal to a random intermediate router and then minimal to the destination, as defined before. This is the reference mechanism for adversarial traffic.

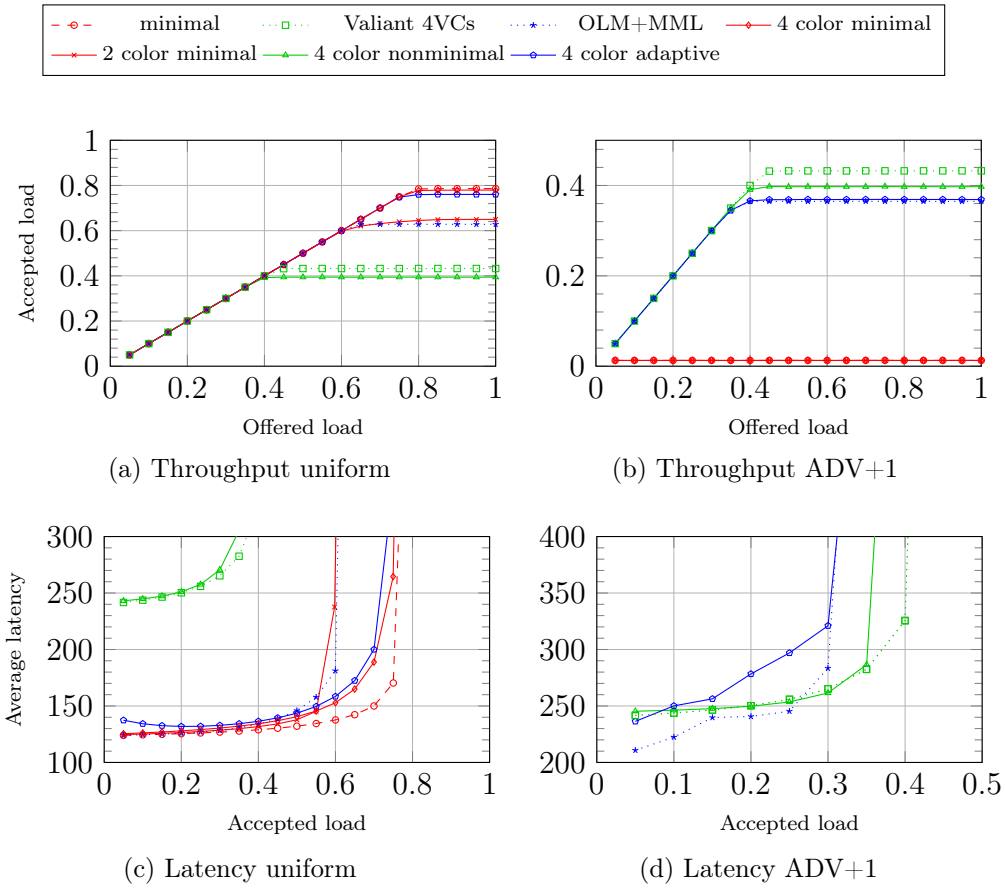
- *2-color Minimal*, *4-color Minimal* and *Nonminimal*: As described in Section 3.6.

Additionally, two adaptive mechanisms have been implemented:

Routing	Adaptive	Min VCs (local/global)	Min. trunking
Minimal	No	2/1	1
Valiant	No	4/2	1
2-color minimal	No	1/1	2
4-color minimal/nonminimal	No	1/1	4
OLM	Yes	3/2	1
4-color adaptive	Yes	1/1	4

Table 5.5: Parameters of each routing mechanism.

- *OLM-MML*: An in-transit adaptive routing mechanism described in [GVB+13c], using the MM+L global misrouting policy from [GVB+13b]. This mechanism has been selected because it provides similar or better performance than the naïve PAR6/2 from the same paper, while requiring less virtual channels.
- *4-color adaptive*: The 4-color routing presented in Section 3.6.2, implementing Piggybacking [JKD09] to adaptively select between minimal (*lgl*) or nonminimal (*lgllgl*) paths at injection time, using information from the neighbour routers.



(a) Throughput uniform

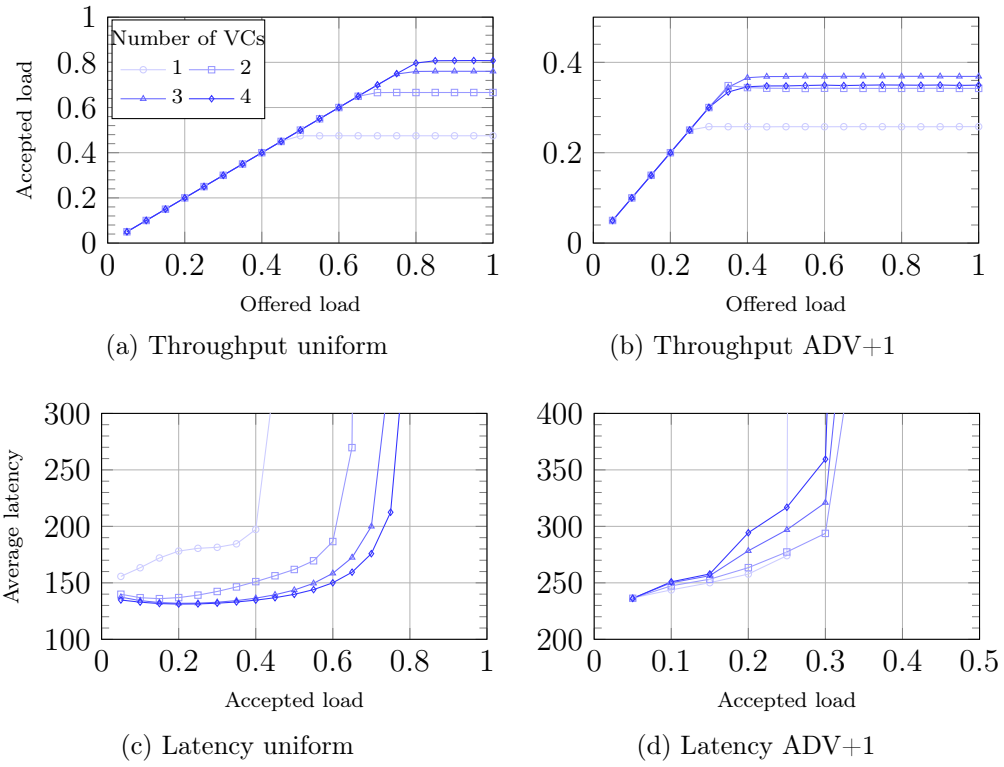
(b) Throughput ADV+1

(c) Latency uniform

(d) Latency ADV+1

Figure 5.25: Throughput and average latency for uniform and ADV+1 traffic.

Figure 5.25 shows latency and throughput results under minimal and adverse traffic patterns. As expected, minimal routings give the best results for uniform traffic but accept



(a) Throughput uniform

(b) Throughput ADV+1

(c) Latency uniform

(d) Latency ADV+1

Figure 5.26: Throughput and average latency for uniform and ADV+1 traffics varying the number of virtual channels

an insignificant amount of the adverse traffic. 2 and 4 colors minimal routings obtain approximately the same latency with a slight advantage for the 2-color implementation at medium loads, but higher throughput for the 4-color variant. The lower throughput of the 2-color variant comes from some groups using local links of type l_{+0} in most of the first local hops, especially in those with a low group index: l_{+0} is used in the first local hop both when the source and destination colors are the same, and when they differ and the destination group has a higher index. An alternative more balanced ordering between each pair of groups (without introducing cyclic dependencies) would mitigate this effect. Interestingly, the 4-color adaptive mechanism throughput is close to the maximum, despite being adaptive (and thus making some erroneous decisions).

The 4-color nonminimal implementation is relatively close to the performance of Valiant, despite its route restrictions and the use of one VC less for HoLB, which explain the lower throughput. The adaptive variant reacts very well to the traffic pattern, almost reaching the accepted load of its respective oblivious minimal and non-minimal counterparts. It also accepts more load than the OLM reference for adaptive routing. However latency is higher than OLM and Valiant especially for adverse traffic; this comes from worse decisions using source-routing in the 4-color mechanism, compared to in-transit adaptivity of OLM, which can save intermediate hops when no congestion is detected (for ADV+1 traffic, one hop in the intermediate group).

Figure 5.26 shows the performance of 4-color adaptive using different number of virtual channels. As the mechanism does not restrict VCs at all, any number of buffers can be employed, unlike the fixed requirement imposed by other policies. Furthermore, there is freedom in the way to use the VCs; in these simulations the VC is selected randomly

among those available. The HoLB problem is seen to have a great impact especially when not using VCs (*i.e.* 1 VC), but the performance with just 2 VCs noticeably improves. It is interesting to see that for adversarial traffic the delay increases with the number of VCs; this is explained by the higher capacity of the network buffers before the bottleneck, which increases the number of stored packets. When the buffer count is low, explicit mechanisms could be employed to mitigate the HoLB performance issues, such as internal router speedup or virtual-output queueing [TF88]. Such mechanisms are widely known and they have not been explored in this work.

Chapter 6

Conclusions

This final chapter shows some conclusions about the results and describes some interesting future lines of work. It ends with a list of publications that were written during the realization of this thesis, both related and non-related to this thesis.

6.1 About the Results

Tori are used in many of the top supercomputers. In many cases, that implementations have mixed-radix, which as seen, causes a notable performance lose. This can be solved by doing the appropriate twist with negligible increase of cost. Indeed, the greatest cost seems to understand the related concepts, but we think anyone can grasp them with some dedication. Thus, we do not see any impediment to the adoption of these topologies. A little more hard would be to give to a Blue Gene like system the capability of partitioning into multitude of different topologies. We have seen that it is possible with a few hardware modifications. Of course, it also would require work to implement the software handling the many different possible modes of the link chips.

When we read about the dragonfly we saw that it had a lot of potential; however its formal definition was very vague, which got us a little peeved, since we can actually fit any graph in that definition. Hence, effort has been dedicated to make a more proper definition trying to keep the original spirit. We also read about the flattened butterfly topology, which resulted to be the same as the Hamming graph or the generalized hypercube, and again made us displeased, this time for the lack of good references. We hope to have at least cleared up these aspects and to have not omitted ourselves something important. From a more practical viewpoint, the dragonflies that are actually implemented have some global trunking, which contrasts with the original study, which focused on dragonflies without any trunking. Together with the view of Hamming graphs as dragonflies with a lot of trunking this suggested that those implementations of dragonflies could have more potential. We have shown this to be true by providing a deadlock-free routing algorithm that does not require virtual channels when some trunking is present; this conforms with the existence of DOR in Hamming graphs.

Using the techniques learn when developing the lattice graphs we have been able to give some results about coding theory. We have shown the equivalence with topologies, although is not obvious how to implement systems based on them, in a similar way as how is not obvious to implement systems based on other families of graphs that attack the degree-diameter problem. Anyway, from the coding theory perspective the results seem rather important. The Golomb–Welch conjecture says that the only perfect Lee codes

for dimension $n \geq 3$ have radius (correction) 1. Recently, there have been advances only for low dimension, up to $n \leq 6$ in general or up to $n \leq 12$ with some restrictions. The last results for arbitrarily large dimension were before 1982. Those results showed that there are not perfect Lee codes where the radius is greater than $\sqrt{n}$ modified by some constants. Indeed, if the radius is very large, then the problem can be approximated to the problem of tiling $\mathbb{R}^n$ with cross-polytopes, as shown by Golomb and Welch [GW70]. However, that said nothing at all about low radius. We have shown that for radius 2 and arbitrarily large dimension there are quasi-perfect Lee codes with approximately half the density that perfect Lee codes would have if they exist. Moreover, there is no reason to think that the codes we have obtained are the best ones; indeed, we have found a bunch of better ones for low dimensions. What is clear is that more advanced techniques are required to attack the conjecture.

6.2 Ongoing and Future Work

Dragonfly networks have a good degree-diameter relation (4/27 of the bound in its more expanded form) and possess good properties to make easy its implementation. Nevertheless, there are other topologies with much better degree-diameter relation, some of them asymptotically reaching the bound. However, these graphs do not possess the implementation benefits of the dragonfly. Nevertheless, there are currently some proposals of them as interconnection topologies [BBC13, BH14]. One of the major problems of these families reaching asymptotically the Moore bound is that the graph only exists when some parameter is a power of a prime number—to be working in a finite field—which can seem preposterous to any system provider that have clients requesting different sizes. Thus, an important problem is: do these families can be generalized to contain graphs for every number of vertices? This problem can be hard, since it implies getting out of the comfortable finite fields (and of their projective spaces). This also links with the quasi-perfect Lee codes that we have constructed. It is possible to extend them to many sizes? How should be a network system based on them?

When trying to find good codes, many ideas have appeared. A few codes have been found that are better than the infinite construction. Spectral computation has shown that most of the graphs in the construction are Ramanujan graph. Weil's conjectures have appeared, first with an use of the Hasse–Weil theorem to get a bound on the number of solutions of a polynomial and later they have seemed to be implicated in the family being Ramanujan graphs. We have not introduced weighing matrices and the Cayley Dickson's construction; they give good codes, although we have not found a conclusive way of using them. Thus, there is not shortage of ideas to continue attacking Golomb–Welch conjecture.

6.3 Publications During the Realization of this Thesis

Along the development of this thesis, several works have been published in journals and conferences. Here is the list this publications that form part of this thesis or are directly related to it:

1. [MCB10] Carmen Martínez, **Cristóbal Camarero**, and Ramón Beivide. Perfect graph codes over two dimensional lattices. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1047–1051, June 2010.

2. [CMB10] **Cristóbal Camarero**, Carmen Martínez, and Ramón Beivide. Symmetric L-networks. In *2010 International Workshop on Optimal Network Topologies*, 2010.

3. [CMB13] **Cristóbal Camarero**, Carmen Martínez, and Ramón Beivide. L-networks: A topological model for regular 2D interconnection networks. *Computers, IEEE Transactions on*, 62(7):1362–1375, July 2013.

4. [CVM+13] **Cristóbal Camarero**, Enrique Vallejo, Carmen Martínez, Miquel Moretó, and Ramón Beivide. Task mapping in rectangular twisted tori. In *Proceedings of the High Performance Computing Symposium*, HPC '13, pages 15:1–15:11, San Diego, CA, USA, 2013. Society for Computer Simulation International.

5. [QCMPJ13] Cátia Quilles Queiroz, **Cristóbal Camarero**, Carmen Martínez, and Reginaldo Palazzo Jr. Quasi-perfect codes from Cayley graphs over integer rings. *Information Theory, IEEE Transactions on*, 59(9):5905–5916, September 2013.

6. [CMB14] **Cristóbal Camarero**, Carmen Martínez, and Ramón Beivide. Lattice graphs for high-scale interconnection topologies. *Parallel and Distributed Systems, IEEE Transactions on*, 2014.

7. [CVB14] **Cristóbal Camarero**, Enrique Vallejo, and Ramón Beivide. Topological characterization of Hamming and dragonfly networks and its implications on routing. *ACM Trans. Archit. Code Optim.*, 11(4):39:1–39:25, December 2014.

Other publications are less related to the present thesis. They are briefly described before listing them. Some works on *king* topologies [SBM+10, SCV+12, CSV+12]; the king torus of side a is the graph $Cay(\mathbb{Z}_a^2; \{\pm\mathbf{e}_1, \pm\mathbf{e}_2, \pm(\mathbf{e}_1+\mathbf{e}_2), \pm(\mathbf{e}_1-\mathbf{e}_2)\})$, so they are actually lattice graphs. Then the king mesh is to the king torus as the mesh is to the torus. The king mesh and king torus will be the main theme of Esteban Stafford's thesis, which hopefully will be written soon. Some works on dragonflies have been a collaboration on Marina García's thesis: [GVB+12a, GVB+12b, GVB+13b, GVB+15]. They mainly consist on strategies to adapt well to terrible traffic patterns. A work about identifying codes is in [CMB11, CMB15]. A collaboration with Emilio Castillo *et al.* on a routing mechanism for tori [CCS+13]. Another collaboration with Emilio Castillo *et al.* where we won a challenge posed by the Santander bank called "Métodos Numéricos alternativos a Montecarlo" and has become a publication on the Journal of Supercomputing as [CCBB15].

1. [SBM+10] Esteban Stafford, Jose Luis Bosque, Carmen Martínez, Fernando Vallejo, Ramón Beivide, and **Cristóbal Camarero**. A first approach to king topologies for on-chip networks. In *Proceedings of the 16th international Euro-Par conference on Parallel processing: Part II*, Euro-Par'10, pages 428–439, Berlin, Heidelberg, 2010. Springer-Verlag.

2. [CMB11] **Cristóbal Camarero**, Carmen Martínez, and Ramón Beivide. Identifying codes over L-graphs. In *3rd International Castle Meeting on Coding Theory and Applications*, 3ICMTA, pages 81–87, Barcelona, SPAIN, 2011. UB Servei de Publicacions.

3. [SCV+12] Esteban Stafford, Emilio Castillo, Fernando Vallejo, José Luis Bosque, Carmen Martínez, **Cristóbal Camarero**, and Ramón Beivide. King topologies for fault tolerance. In *High Performance Computing and Communication & 2012 IEEE*

9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on, pages 608–616. IEEE, June 2012.

4. [CSV+12] Emilio Castillo, Esteban Stafford, Fernando Vallejo, Jose Luis Bosque, Carmen Martínez, **Cristóbal Camarero**, and Ramón Beivide. Study of fault tolerance for king topologies. In *jornadas sarteco*, September 2012.

5. [GVB+12a] Marina García, Enrique Vallejo, Ramón Beivide, Miguel Odriozola, **Cristóbal Camarero**, Mateo Valero, Germán Rodríguez, Jesús Labarta, and Cyriel Minkenberg. Bubble flow control in high-radix hierarchical networks. In *jornadas sarteco*, September 2012.

6. [GVB+12b] Marina García, Enrique Vallejo, Ramón Beivide, Miguel Odriozola, **Cristóbal Camarero**, Mateo Valero, Germán Rodríguez, Jesús Labarta, and Cyriel Minkenberg. On-the-fly adaptive routing in high-radix hierarchical networks. In *The 41st International Conference on Parallel Processing (ICPP)*, pages 279–288, September 2012.

7. [GVB+13b] Marina García, Enrique Vallejo, Ramón Beivide, Miguel Odriozola, **Cristóbal Camarero**, Mateo Valero, J. Labarta, and G. Rodríguez. Global misrouting policies in two-level hierarchical networks. In *Proceedings of the 2013 Interconnection Network Architecture: On-Chip, Multi-Chip*, IMA-OCMC '13, pages 13–16, New York, NY, USA, 2013. ACM.

8. [CCS+13] Emilio Castillo, **Cristóbal Camarero**, Esteban Stafford, Fernando Vallejo, Jose Luis Bosque, and Ramón Beivide. Advanced switching mechanisms for forthcoming on-chip networks. In *Digital System Design (DSD), 2013 Euromicro Conference on*, pages 598–605. IEEE, September 2013.

9. [CCBB15] Emilio Castillo, **Cristóbal Camarero**, Ana Borrego, and Jose Luis Bosque. Financial applications on multi-CPU and multi-GPU architectures. *The Journal of Supercomputing*, 71(2):729–739, 2015.

10. [CMB15] **Cristóbal Camarero**, Carmen Martínez, and Ramón Beivide. Identifying codes of degree 4 Cayley graphs over Abelian groups. Accepted for publication in *Advances in Mathematics of Communications*, 2015.

11. [GVB+15] Marina García, Enrique Vallejo, Ramón Beivide, **Cristóbal Camarero**, Mateo Valero, Germán Rodríguez, and Cyriel Minkenberg. On-the-fly adaptive routing for dragonfly interconnection networks. *The Journal of Supercomputing*, 71(3):1116–1142, 2015.

Appendix A

Classes of Symmetric Lattice Graphs of Degrees 4 and 6

This appendix is devoted to characterize the family of symmetric undirected lattice graphs for degrees 4 and 6.

A.1 Introduction

The present appendix is devoted to characterize the symmetric members of the family of undirected lattice graphs for degrees 4 and 6. Since these graphs are known to be vertex-transitive [AK89], the characterization will be done by determining those being edge-transitive.

Definition A.1.1. *Let $M_1, M_2 \in \mathbb{Z}^{n\times n}$. Then, M_1 is right equivalent to M_2, denoted by $M_1 \cong M_2$, if and only if there exists a unit matrix $P \in \mathbb{Z}^{n\times n}$ such that $M_1 = M_2 P$.*

Theorem A.1.2. *[Fio95]. If a pair of matrices $M_1, M_2 \in \mathbb{Z}^{n\times n}$ are right-equivalent, then $\mathcal{G}(M_1)$ and $\mathcal{G}(M_2)$ are isomorphic graphs.*

Definition A.1.3. *A signed permutation matrix is a matrix with entries in $\{-1, 0, 1\}$ which has exactly one ± 1 in each row and column.*

Note that in $\mathbb{Z}^{n\times n}$ the signed permutation matrices are exactly the unitary matrices, this is, the matrices U such $UU^t = I$. They are related to permutations in the way that for each permutation $\sigma \in \Sigma_n$ there is a unique signed permutation matrix P_σ such that

$$P_\sigma \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} \pm v_{\sigma(1)} \\ \vdots \\ \pm v_{\sigma(n)} \end{pmatrix}.$$

Theorem A.1.4. *[Fio95] If P is a signed permutation matrix then $\mathcal{G}(PM) \cong \mathcal{G}(M)$.*

In this appendix we will find matrices such that $\mathcal{G}(M)$ is edge-transitive for dimensions 2 and 3. In the first case, the characterization will be complete, that is we will find all $\mathcal{G}(M)$ being symmetric. In the case of dimension 3, we will only consider those being edge-transitive by means of linear automorphism, as explained later.

With this aim, in Section A.2 we will consider some properties of isomorphisms between lattice graphs, their automorphisms and the implications of containing cycles of length 4.

In Section A.3, we give some general results for the characterization of $\mathcal{G}(M)$ graphs of any dimension which are symmetric by means of linear automorphisms. In Section A.4 we give the full characterization of symmetric $\mathcal{G}(M)$ graphs of dimension 2 (degree 4). In Section A.5 we give the full characterization of symmetric by linear automorphisms $\mathcal{G}(M)$ graphs of dimension 3 (degree 6).

A.2 Linear Automorphisms of Lattice Graphs and 4-cycles

Given a graph G, $Aut(G)$ denotes its automorphisms group. G is said to be *vertex-transitive* if, for any pair of vertices $\mathbf{v}_1, \mathbf{v}_2 \in V$ there exists $f \in Aut(G)$ such that $f(\mathbf{v}_1) = \mathbf{v}_2$. Similarly, G is said to be *edge-transitive* if for any pair of edges $a = \{\mathbf{v}_1, \mathbf{v}_2\}, b \in E$ there exists $f \in Aut(G)$ such that $f(a) = \{f(\mathbf{v}_1), f(\mathbf{v}_2)\} = b$. Then, if G is both vertex and edge transitive, then it is called *symmetric*. The subgroup of $Aut(G)$ of elements which fix some element $\mathbf{x} \in V$ is denoted as $Aut(G, \mathbf{x})$ (also known as *stabilizer*).

All Cayley graphs graphs are vertex-transitive [AK89]. The linear automorphisms of a lattice graph $\mathcal{G}(M)$ form a group $LAut(\mathcal{G}(M))$. This group usually coincides with the full automorphism group $Aut(\mathcal{G}(M))$, except in a few cases that we consider separately. We also consider the group of linear automorphisms which fixes $\mathbf{0}$, $LAut(\mathcal{G}(M), \mathbf{0})$.

Definition A.2.1. *$\mathcal{G}(M)$ is said linearly edge-transitive if for every i there exists $f \in LAut(\mathcal{G}(M), \mathbf{0})$ such that $f(\mathbf{e}_1) = \pm\mathbf{e}_i$.*

Clearly, a linearly edge-transitive lattice graph $\mathcal{G}(M)$ is symmetric. Therefore, in this section we study the automorphism group of $\mathcal{G}(M)$ graphs. A basic question is determining when there are nonlinear automorphisms; which is very related to the problem of determining *Ádám isomorphy* [Ádá67, DFM92]. A pair of graphs $\mathcal{G}(M_1)$ and $\mathcal{G}(M_2)$ are *Ádám isomorphic* if there exists an isomorphism between their groups of vertices such that it sends the set of generators of one graph into the generators of the other graph. It is clear that any Ádám isomorphic graphs are isomorphic, but the opposite it is not always true.

In [DFM92] it was proved that any pair of isomorphic lattice multigraphs of degree four are Ádám isomorphic unless the pair is (up to Ádám isomorphy) $\begin{pmatrix} 2k+1 & 2 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 2k & 2 \\ 0 & 2 \end{pmatrix}$ for some integer k. Using the nonlinear isomorphism between them one can build a nonlinear automorphism in each; hence they will appear in our study of the nonlinear automorphisms of dimension 2. However the reverse is not true, as there are a few graphs with nonlinear automorphisms which do not have a pairing non-Ádám isomorphic graph.

Definition A.2.2. *The neighborhood of a vertex v in the graph $G = (V, E)$ is defined as $N(v) = \{w \mid \{v, w\} \in E\}$. Then, the common neighborhood of a list of vertices $v_1, \ldots, v_n \in V$ as $N(v_1, \ldots, v_n) = \bigcap_{i=1}^{n} N(v_i)$.*

Theorem A.2.3. *The neighborhood is preserved in graph isomorphisms. That is, if f is a graph isomorphism, then*

$$N(f(v_1), \ldots, f(v_n)) = \{f(w) \mid w \in N(v_1, \ldots, v_n)\}.$$

Proof. Let f be a graph isomorphism from $G = (V, E)$ into $G' = (V', E')$. We have that $f(w) \in N(f(v_1), \ldots, f(v_n))$ if only if $\forall i, f(w) \in N(f(v_i))$, that is $\forall i, \{f(w), f(v_i)\} \in E'$. Since f is an isomorphism we have that this is equivalent to $\forall i, \{w, v_i\} \in E$ so $w \in N(v_1, \ldots, v_n)$. □

Next, we analyze which isomorphisms between lattice graphs are linear mappings. This is related to the following concept.

Definition A.2.4. *We say that* $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \pm\mathcal{B}_n$ *form a 4-cycle in* $\mathcal{G}(M)$ *if* $\mathbf{0} \equiv \mathbf{a}+\mathbf{b}+\mathbf{c}+\mathbf{d} \pmod{M}$[1]. *If we have* $\mathbf{a} \in \{-\mathbf{b}, -\mathbf{c}, -\mathbf{d}\}$ *then we call the cycle trivial. Then, we say that* $\mathcal{G}(M)$ *has not nontrivial 4-cycles if all its 4-cycles are trivial.*

Theorem A.2.5. *If* $\mathcal{G}(M)$ *is has not nontrivial 4-cycles, then for all* $\mathbf{a}, \mathbf{b} \in \pm\mathcal{B}_n$ *with* $\mathbf{a} \neq \mathbf{b}$

$$N(\mathbf{a}, \mathbf{b}) = \{\mathbf{0}, \mathbf{a} + \mathbf{b}\}.$$

Proof. If $\mathbf{v} \in N(\mathbf{a}, \mathbf{b})$ then $\exists \mathbf{x}, \mathbf{y} \in \pm\mathcal{B}_n$ such that $\mathbf{v} = \mathbf{a} + \mathbf{x} = \mathbf{b} + \mathbf{y}$. Since we have $\mathbf{a} - \mathbf{b} + \mathbf{x} - \mathbf{y} = \mathbf{0}$ and $\mathcal{G}(M)$ has not nontrivial 4-cycles, it must be fulfilled one of the following expressions:

- $\mathbf{a} = \mathbf{b}$ contradicting the hypothesis,
- $\mathbf{a} = -\mathbf{x}$ and thus $\mathbf{v} = \mathbf{a} - \mathbf{a} = \mathbf{0}$,
- $\mathbf{a} = \mathbf{y}$ and thus $\mathbf{v} = \mathbf{b} + \mathbf{y} = \mathbf{a} + \mathbf{b}$.

□

Lemma A.2.6. *If* f *is an automorphism of* $\mathcal{G}(M)$*, then for any* $\mathbf{t} \in \mathbb{Z}^n/M\mathbb{Z}^n$*,* $f_\mathbf{t} : \mathbf{x} \mapsto f(\mathbf{t} + \mathbf{x}) - f(\mathbf{t})$ *is an automorphism of* $\mathcal{G}(M)$ *with* $f_\mathbf{t}(\mathbf{0}) = \mathbf{0}$.

Proof. We have $f_\mathbf{t}(\mathbf{0}) = f(\mathbf{t} + \mathbf{0}) - f(\mathbf{t}) = \mathbf{0}$, thus $f_\mathbf{t}$ fixes $\mathbf{0}$. Now if $\mathbf{x} \in \mathbb{Z}^n/M\mathbb{Z}^n$ is adjacent to $\mathbf{y} \in \mathbb{Z}^n/M\mathbb{Z}^n$ then $\mathbf{t}+\mathbf{x}$ is adjacent to $\mathbf{t}+\mathbf{y}$ and then as f is an automorphism we have that $f(\mathbf{t} + \mathbf{x})$ is adjacent to $f(\mathbf{t} + \mathbf{y})$. Hence $f_\mathbf{t}(\mathbf{x})$ is adjacent to $f_\mathbf{t}(\mathbf{y})$. □

Lemma A.2.7. *Let* $\mathcal{G}(M)$ *be such that it has not nontrivial 4-cycles. Then for any* $f \in Aut(\mathcal{G}(M), \mathbf{0})$ *we have that* $f(\mathbf{a} + \mathbf{b}) = f(\mathbf{a}) + f(\mathbf{b})$ *for any* $\mathbf{a}, \mathbf{b} \in \pm\mathcal{B}_n$.

Proof. Let $\mathbf{a}, \mathbf{b} \in \pm\mathcal{B}_n$. First we prove the lemma for $\mathbf{a} \neq \mathbf{b}$. From Theorem A.2.5 we get that $N(\mathbf{a}, \mathbf{b}) = \{\mathbf{0}, \mathbf{a} + \mathbf{b}\}$, hence by Theorem A.2.3 $N(f(\mathbf{a}), f(\mathbf{b})) = \{f(\mathbf{0}), f(\mathbf{a} + \mathbf{b})\} = \{\mathbf{0}, f(\mathbf{a}) + f(\mathbf{b})\}$. As $f(\mathbf{0}) = \mathbf{0}$ we have that $f(\mathbf{a} + \mathbf{b}) = f(\mathbf{a}) + f(\mathbf{b})$.

Now note that since for any $\mathbf{a} \in \pm\mathcal{B}_n$, $\mathbf{a} \neq -\mathbf{a}$ we have that for any $f \in Aut(\mathcal{G}(M), \mathbf{0})$, $f(-\mathbf{a}) = -f(\mathbf{a})$.

It remains to prove that $f(2\mathbf{a}) = 2f(\mathbf{a})$. Consider the automorphism f' defined by $f'(\mathbf{v}) = f(\mathbf{a}+\mathbf{v})-f(\mathbf{a})$ (it is an automorphism by Lemma A.2.6). We have $f'(-\mathbf{a}) = -f'(\mathbf{a})$, hence $f(\mathbf{a} - \mathbf{a}) - f(\mathbf{a}) = -(f(\mathbf{a} + \mathbf{a}) - f(\mathbf{a}))$. Rearranging terms we obtain the desired $f(2\mathbf{a}) = 2f(\mathbf{a})$. □

Lemma A.2.8. *If* $\forall \mathbf{a}, \mathbf{b} \in \pm\mathcal{B}_n$, $f \in Aut(\mathcal{G}(M), \mathbf{0})$, $f(\mathbf{a} + \mathbf{b}) = f(\mathbf{a}) + f(\mathbf{b})$ *then every* $f \in Aut(\mathcal{G}(M), \mathbf{0})$ *is a group automorphism of* $\mathbb{Z}^n/M\mathbb{Z}^n$.

[1] each of $\{(\mathbf{v}, \mathbf{v}+\mathbf{a}, \mathbf{v}+\mathbf{a}+\mathbf{b}, \mathbf{v}+\mathbf{a}+\mathbf{b}+\mathbf{c}, \mathbf{v}+\mathbf{a}+\mathbf{b}+\mathbf{c}+\mathbf{d}) \mid \mathbf{v} \in \mathbb{Z}^n/M\mathbb{Z}^n)\}$ is a cycle of length 4.

Proof. First we prove that for all $f \in Aut\big(\mathcal{G}(M), \mathbf{0}\big)$ we have that

$$\forall \mathbf{t} \in \mathcal{G}(M),\ f(\mathbf{t}+\mathbf{a}+\mathbf{b}) = f(\mathbf{t}+\mathbf{a}) + f(\mathbf{t}+\mathbf{b}) - f(\mathbf{t}).$$

Let $\mathbf{t} \in \mathcal{G}(M)$. We define $f_{\mathbf{t}}(\mathbf{v}) = f(\mathbf{t}+\mathbf{v}) - f(\mathbf{t})$, by Lemma A.2.6 $f_{\mathbf{t}} \in Aut(\mathcal{G}(M), \mathbf{0})$. By hypothesis, we have $\forall \mathbf{t} \in \mathcal{G}(M),\ f_{\mathbf{t}}(\mathbf{a}+\mathbf{b}) = f_{\mathbf{t}}(\mathbf{a}) + f_{\mathbf{t}}(\mathbf{b})$, which implies $\forall \mathbf{t} \in \mathcal{G}(M),\ f(\mathbf{t}+\mathbf{a}+\mathbf{b}) - f(\mathbf{t}) = f(\mathbf{t}+\mathbf{a}) - f(\mathbf{t}) + f(\mathbf{t}+\mathbf{b}) - f(\mathbf{t})$.

We need to prove $\forall n_i \in \mathbb{N},\ f(\sum_i n_i\mathbf{e}_i) = \sum_i n_i f(\mathbf{e}_i)$. We proceed by induction in $N = \sum_i n_i$; for $N = 0, 1$ it is immediate. Now let $\mathbf{v} = \sum_i n_i\mathbf{e}_i$ and $\sum_i n_i = N+1$. Let u, v be any positive integers such that $n_u + n_v \geq 2$. Now, because of the first claim, $f(\mathbf{v}) = f((\mathbf{v}-\mathbf{e}_u-\mathbf{e}_v)+\mathbf{e}_u+\mathbf{e}_v) = f((\mathbf{v}-\mathbf{e}_u-\mathbf{e}_v)+\mathbf{e}_u)+f((\mathbf{v}-\mathbf{e}_u-\mathbf{e}_v)+\mathbf{e}_v)-f(\mathbf{v}-\mathbf{e}_u-\mathbf{e}_v)$. Applying the induction hypothesis we have that: $f(\mathbf{v}) = \big(f(\mathbf{v}-\mathbf{e}_u-\mathbf{e}_v)+f(\mathbf{e}_u)\big) + \big(f(\mathbf{v}-\mathbf{e}_u-\mathbf{e}_v)+f(\mathbf{e}_v)\big) - f(\mathbf{v}-\mathbf{e}_u-\mathbf{e}_v) = f(\mathbf{v}-\mathbf{e}_u-\mathbf{e}_v)+f(\mathbf{e}_u)+f(\mathbf{e}_v)$. Then as $f(\mathbf{v}-\mathbf{e}_u-\mathbf{e}_v) = f(\sum_i n_i\mathbf{e}_i - \mathbf{e}_u - \mathbf{e}_v) = \sum_i n_i f(\mathbf{e}_i) - f(\mathbf{e}_u) - f(\mathbf{e}_v)$ we have that $f(\mathbf{v}) = \sum_i n_i f(\mathbf{e}_i)$. □

Theorem A.2.9. *If the graph $\mathcal{G}(M)$ has not nontrivial 4-cycles then any graph automorphism with $f(\mathbf{0}) = \mathbf{0}$ is a group automorphism of $\mathbb{Z}^n/M\mathbb{Z}^n$.*

Proof. If there are not nontrivial 4-cycles then by Lemma A.2.7 we have for any $f \in Aut(\mathcal{G}(M), \mathbf{0})$ that $f(\mathbf{a}+\mathbf{b}) = f(\mathbf{a}) + f(\mathbf{b})$ for any $\mathbf{a}, \mathbf{b} \in \pm\mathcal{B}_n$. Now we have linearity by Lemma A.2.8. □

A.3 Edge-Transitivity of Lattice Graphs by Linear Automorphisms

In this section we will consider those graphs $\mathcal{G}(M)$ such that any of its automorphisms is a linear mapping.

Theorem A.3.1. *For any $f \in LAut(\mathcal{G}(M), \mathbf{0})$ there exists a signed permutation matrix P such that $\forall \mathbf{a} \in \mathbb{Z}^n/M\mathbb{Z}^n,\ f(\mathbf{a}) = P\mathbf{a}$.*

Proof. We define P as:

$$P_{i,j} = \begin{cases} 1 & \text{if } f(\mathbf{e}_j) = \mathbf{e}_i \\ -1 & \text{if } f(\mathbf{e}_j) = -\mathbf{e}_i \\ 0 & \text{otherwise} \end{cases}$$

having $f(\mathbf{e}_i) = P\mathbf{e}_i$. Let $\mathbf{a} = \sum n_i\mathbf{e}_i$. Now,

$$f(\mathbf{a}) = \sum_i n_i f(\mathbf{e}_i) = \sum_i n_i P\mathbf{e}_i = P\sum_i n_i\mathbf{e}_i = P\mathbf{a}.$$

□

Theorem A.3.2. *For any $M \in \mathbb{Z}^{n\times n}$ the mapping $f(\mathbf{x}) = P\mathbf{x}$ is a linear automorphism of $\mathcal{G}(M)$ if only if there exists $Q \in \mathbb{Z}^{n\times n}$ such that $PM = MQ$.*

Proof. We prove first the left to right implication. As f must be well-defined, for all i, $\mathbf{0} = P\mathbf{0} \equiv PM\mathbf{e}_i \pmod{M}$. And then exists $\mathbf{q}_i$ such that $PM\mathbf{e}_i = M\mathbf{q}_i$, gathering all is together

$$PM = [PM\mathbf{e}_1, \ldots, PM\mathbf{e}_n] = M[\mathbf{q}_1, \ldots, \mathbf{q}_n] = MQ.$$

For the right to left implication; by Theorem A.1.4 f is an isomorphism from $\mathcal{G}(M)$ into $\mathcal{G}(PM) = \mathcal{G}(MQ)$. Then by Theorem A.1.2 f is an automorphism of $\mathcal{G}(M)$. □

To know if $\mathcal{G}(M)$ is linearly edge-transitive we need to look to the multiplicative group of the signed permutation matrices P such $PM = MQ$. It is clear that, if a matrix representing a cycle of length n (even if it changes signs) is in the group then by composing it with itself, we can map $\mathbf{e}_1$ to every $\mathbf{e}_i$ making the graph edge-transitive. In these cases we have that $LAut(\mathcal{G}(M), \mathbf{0})$ is a cyclic group. The smallest dimension for which we found $LAut(\mathcal{G}(M), \mathbf{0})$ to be noncyclic is for $n = 4$ with the Klein four-group. That situation occurs for example for Lipschitz graphs, which were introduced in in [MBG09]. Since we just consider dimensions 2 and 3, this will not suppose any problem.

Definition A.3.3. *Two matrices $A, B \in \mathbb{Z}^{n\times n}$ are similar, denoted by $A \sim B$, if there exists a unit matrix $U \in \mathbb{Z}^{n\times n}$ such that $AU = UB$.*

Lemma A.3.4. *Let $PM = MQ$ and $PM' = M'Q'$. Then, $M \cong M'$ if and only if $Q \sim Q'$.*

Proof. Since $PM = MQ$ and $M = M'U$ then $PM'U = M'UQ$ and $PM' = M'(UQU^{-1}) = M'Q'$ with $Q' \sim Q$. Reciprocally, we know that if $PM = MQ$ and $Q' = UQU^{-1}$ then $M' = MU$ produces $PM' = M'Q'$ and $M' \cong M$. □

Since right equivalences leave the group invariant (Theorem A.1.2), we know that for a given P we only need to see how many Q there are modulo similarity. Then, knowing P and Q we can solve for M. In [New72] the next theorems are stated, which will be very helpful in the determination of Q in the following Sections A.4 and A.5.

Theorem A.3.5 ([New72], Theorem III.12, page 50)**.** *Given a matrix A we can find a similar matrix, made of blocks, which is block upper triangular and moreover, that the blocks of the diagonal all have characteristic polynomial irreducible over $\mathbb{Q}$.*

Theorem A.3.6 ([New72], Theorem III.14, pag 53, The theorem of Lattimer and MacDuffee)**.** *If we have a matrix with irreducible characteristic polynomial, like the produced by the previous theorem then the number of matrices modulo similarity is the class number of $\mathbb{Z}[\theta]$ where θ is a root of the polynomial.*

A.4 Characterization of Symmetric Lattice Graphs of Dimension 2

The complete characterization of symmetric $\mathcal{G}(M)$ graphs with $M \in \mathbb{Z}^{2\times 2}$ will be done in this section. Firstly, we will consider those which are edge-transitive by means of linear automorphism. Later, we will consider those cases involving non-linear automorphisms.

By Theorem A.3.1 a graph $\mathcal{G}(M)$ is linearly edge-transitive if there is an automorphism f with $f(\mathbf{e}_1) = \pm\mathbf{e}_2$ and $f(\mathbf{e}_2) = \pm\mathbf{e}_1$. Such automorphism is associated to one of the matrices: $\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}, \begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}, \begin{pmatrix}0 & -1\\ -1 & 0\end{pmatrix}$.

Since $\begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}^3 = \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}$ there is only need to check $\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}$ and $\pm\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$.

Theorem A.4.1. *Let $M \in \mathbb{Z}^{2\times 2}$ be non-singular. Then, $\mathcal{G}(M)$ is linearly edge-transitive if and only if, for some $a, b \in \mathbb{Z}$, M is right equivalent to one of the following matrices:*

$$M_1 = \begin{pmatrix}a & b\\ b & a\end{pmatrix},\ M_2 = \begin{pmatrix}a & -b\\ b & a\end{pmatrix},\ M_3 = \begin{pmatrix}a & -b\\ a & b\end{pmatrix}.$$

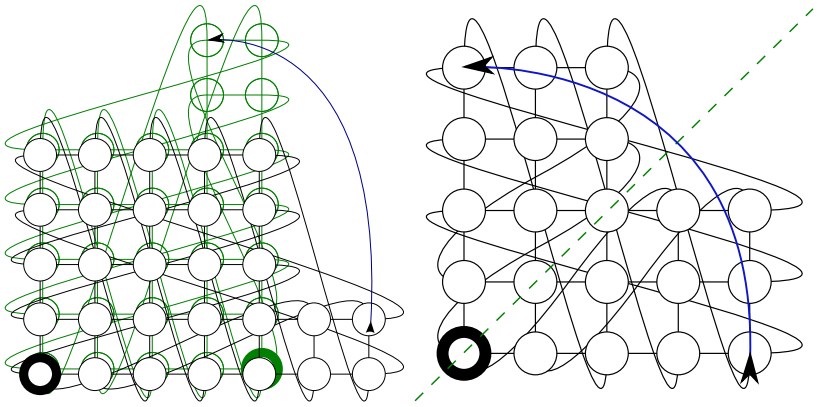


Figure A.1: Linear automorphisms of lattice graphs of dimension 2.

Proof. Let $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. We will determine Q and solve the system $PM = MQ$; which by Theorem A.3.2 is a necessary and sufficient condition to be linearly edge-transitive. The characteristic polynomial of $\pm\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ is $\lambda^2 - 1$, and the one of $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ is $\lambda^2 + 1$. As $PM = MQ$ it must be the characteristic polynomial of both P and Q. By Lemma A.3.4 we can choose any matrix similar to Q and obtain a matrix right-equivalent to M. Therefore, we have two cases:

- $P = \pm\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\lambda^2 - 1 = (\lambda + 1)(\lambda - 1)$, being reducible over $\mathbb{Q}$, by Theorem A.3.5 Q must be similar to a matrix $Q' = \begin{pmatrix} 1 & p \\ 0 & -1 \end{pmatrix}$, which is similar to either $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ or to $\begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix} \sim \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. In the first case, depending on P we obtain M equal to $\begin{pmatrix} a & b \\ b & a \end{pmatrix}$ or to $\begin{pmatrix} a & b \\ -b & -a \end{pmatrix} \cong \begin{pmatrix} a & -b \\ -b & a \end{pmatrix}$; which are the same under the variable change $b \mapsto -b$. In the second case, the same happens for the possible matrices $\begin{pmatrix} a & -b \\ a & b \end{pmatrix}$ and $\begin{pmatrix} a & b \\ -a & b \end{pmatrix} \cong \begin{pmatrix} b & -a \\ b & a \end{pmatrix}$ and the variable change $a \mapsto b$, $b \mapsto a$.

- $P = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$, $\lambda^2 + 1$ which is irreducible over $\mathbb{Q}$ and the class number of $\mathbb{Z}[i]$ is 1, so by Theorem A.3.6 Q must be similar to P. The only possible solutions are $\begin{pmatrix} a & -b \\ b & a \end{pmatrix}$.

□

The first two cases of Theorem A.4.1, $\mathcal{G}(M_1)$ and $\mathcal{G}(M_2)$, are depicted in Figure A.1. As it was proved in [MCB10], $\mathcal{G}(M_1)$ and $\mathcal{G}(M_3)$ are isomorphic to the Kronecker product of two cycles. Furthermore $\mathcal{G}(M_2)$ is isomorphic to the Gaussian graph introduced in [MBS+08].

A.4.1 Edge-Transitive Lattice Graphs of Dimension 2 by Nonlinear Automorphisms

In this subsection we focus on those lattice graphs with nontrivial 4-cycles, and hence, according to Theorem A.2.9 their group of automorphisms could contain nonlinear automorphisms. Clearly, if there is a nontrivial 4-cycle then there exist $\mathbf{a}, \mathbf{b} \in \pm\mathcal{B}_n$ which fulfill:

i) $4\mathbf{a} \equiv \mathbf{0} \pmod{M}$

ii) $3\mathbf{a} + \mathbf{b} \equiv \mathbf{0} \pmod{M}$

iii) $2\mathbf{a} + 2\mathbf{b} \equiv \mathbf{0} \pmod{M}$

If we consider $u\mathbf{a} + v\mathbf{b} \equiv \mathbf{0} \pmod{M}$ it means that there exists $\mathbf{x} \in \mathbb{Z}^2$ such that $\mathbf{k} = \begin{pmatrix} u \\ v \end{pmatrix} = M\mathbf{x}$. Now, let $\gcd(\mathbf{x}) = \gcd(x_1, \ldots, x_n)$, $\mathbf{x}' = \frac{\mathbf{x}}{\gcd \mathbf{x}}$ and $\mathbf{k}' = \frac{\mathbf{k}}{\gcd \mathbf{x}}$, having $\mathbf{k}' = M\mathbf{x}'$. As $\gcd \mathbf{x}' = 1$ we can build a unit matrix U with $\mathbf{x}'$ as one of its columns, and therefore $M' = MU$ has $\mathbf{k}'$ as a column. In addition, Theorem A.1.4 allows to choose each component positive.

We will begin with item (iii). In this case we obtain the matrix $M = \begin{pmatrix} u & 2 \\ v & 2 \end{pmatrix}$. If $v = 2k$ we have that $\begin{pmatrix} u & 2 \\ 2k & 2 \end{pmatrix}$ is right equivalent to $\begin{pmatrix} u-v & 2 \\ 0 & 2 \end{pmatrix}$. On the other hand, if $v = 2k+1$ then $\begin{pmatrix} u & 2 \\ 2k+1 & 2 \end{pmatrix}$ is right equivalent to $\begin{pmatrix} u-v+1 & 2 \\ 1 & 2 \end{pmatrix}$. Both matrices generate the same graph and there is a nonlinear isomorphism between them. In addition note that if the first column has odd weight then $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 2k+1 & 2 \\ 0 & 2 \end{pmatrix} \cong \begin{pmatrix} 2k+2 & 2 \\ 1 & 2 \end{pmatrix}$, so there is a linear isomorphism in addition to the nonlinear one. Hence, the only pairs non Ádám isomorphic are $(\begin{pmatrix} 2k+1 & 2 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 2k & 2 \\ 0 & 2 \end{pmatrix})$, which correspond with the ones in [DFM92]. Furthermore, the matrices of these non Ádám isomorphic graphs satisfy $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} M \cong M$, thus by Theorem A.3.2 they are actually linearly edge-transitive.

For the items (i) and (ii) we begin proving that if there is exactly one nontrivial 4-cycle, then all automorphisms are linear. Furthermore note that these results are also valid in any number of dimensions.

Lemma A.4.2. *Let $f \in Aut(\mathcal{G}(M), \mathbf{0})$ and $\mathbf{a} \in \pm\mathcal{B}_n$. If $\forall \mathbf{b} \in \pm\mathcal{B}_n \setminus \{\mathbf{a}, -\mathbf{a}\}$, $f(-\mathbf{b}) = -f(\mathbf{b})$ then $f^{-1}(-\mathbf{a}) = -f^{-1}(\mathbf{a})$.*

Proof. We check three cases.

- Case $f^{-1}(\mathbf{a}) \neq \pm\mathbf{a}$. Applying the hypothesis we get $\mathbf{a} = f(f^{-1}(\mathbf{a})) = -f(-f^{-1}(\mathbf{a}))$. Then $f^{-1}(-\mathbf{a}) = f^{-1}(f(-f^{-1}(\mathbf{a}))) = -f^{-1}(\mathbf{a})$.
- Case $f^{-1}(-\mathbf{a}) \neq \pm\mathbf{a}$. Applying the hypothesis we get $-\mathbf{a} = f(f^{-1}(-\mathbf{a})) = -f($ $-f^{-1}(-\mathbf{a}))$. Then $f^{-1}(\mathbf{a}) = f^{-1}(f(-f^{-1}(-\mathbf{a}))) = -f^{-1}(-\mathbf{a})$.
- Case $\{f^{-1}(\mathbf{a}), f^{-1}(-\mathbf{a})\} \subseteq \{\pm\mathbf{a}\}$. As f^{-1} is a bijection we have the equality $\{f^{-1}(\mathbf{a}), f^{-1}(-\mathbf{a})\} = \{\pm\mathbf{a}\}$. Now $f^{-1}(\mathbf{a}) + f^{-1}(-\mathbf{a}) = \mathbf{a} + (-\mathbf{a}) = \mathbf{0}$.

□

Theorem A.4.3. *If the only nontrivial 4-cycle is* $4\mathbf{a} \equiv \mathbf{0} \pmod{M}$ *or* $3\mathbf{a}+\mathbf{b} \equiv \mathbf{0} \pmod{M}$ *then*

$$Aut(\mathcal{G}(M)) = LAut(\mathcal{G}(M)).$$

Proof. We proceed proving several claims iteratively.

i) For all $\mathbf{x}, \mathbf{y} \in \pm\mathcal{B}_n \setminus \{\mathbf{a}, -\mathbf{a}\}$, $\mathbf{x} \neq \mathbf{y}$, $N(\mathbf{x}, \mathbf{y}) = \{\mathbf{0}, \mathbf{x}+\mathbf{y}\}$.

We have $N(\mathbf{x}, \mathbf{y}) = \{\mathbf{v} \mid \mathbf{v} = \mathbf{x}+\mathbf{p} = \mathbf{y}+\mathbf{q},\ \mathbf{p}, \mathbf{q} \in \pm\mathcal{B}_n\}$. That is, we look for 4-cycles $\mathbf{x}+\mathbf{p}-\mathbf{y}-\mathbf{q} = \mathbf{0}$. The trivial ones are $\mathbf{x} = -\mathbf{p}$ and $\mathbf{x} = \mathbf{q}$ which respectively give $\mathbf{v} = \mathbf{0}$ and $\mathbf{v} = \mathbf{x}+\mathbf{y}$. If it is the nontrivial 4-cycle $4\mathbf{a} = \mathbf{0}$ then we have $\{\mathbf{x}, \mathbf{y}\} = \{\mathbf{a}, -\mathbf{a}\}$, contradicting the hypothesis. If it is the nontrivial 4-cycle $3\mathbf{a}+\mathbf{b} = \mathbf{0}$, then at least one of $\mathbf{x}$ or $\mathbf{y}$ is $\pm\mathbf{a}$.

ii) For all $\mathbf{x} \in \pm\mathcal{B}_n \setminus \{\mathbf{a}, -\mathbf{a}\}$, $N(\mathbf{a}, \mathbf{x}) \subseteq \{\mathbf{0}, \mathbf{a}+\mathbf{x}, 2\mathbf{a}\}$.

In this case we look for nontrivial 4-cycles $\mathbf{a}+\mathbf{p}-\mathbf{x}-\mathbf{q} = \mathbf{0}$. As $\mathbf{x} \notin \{\pm\mathbf{a}\}$, we have $\mathbf{p} = -\mathbf{q} = \mathbf{a}$ and then $\mathbf{v} = \mathbf{a}+\mathbf{p} = 2\mathbf{a}$. Note that if we have the cycle $4\mathbf{a} = \mathbf{0}$ then we only have the trivial solutions.

iii) $N(\mathbf{a}, -\mathbf{a}) = \{\mathbf{0}, \pm 2\mathbf{a}\}$.

In this case we look for nontrivial 4-cycles $\mathbf{a}+\mathbf{p}+\mathbf{a}-\mathbf{q} = \mathbf{0}$. At least one of $\mathbf{p}, -\mathbf{q}$ is equal to $\mathbf{a}$. If $\mathbf{p} = \mathbf{a}$ then $\mathbf{v} = \mathbf{a}+\mathbf{p} = 2\mathbf{a}$. If $-\mathbf{q} = \mathbf{a}$ then $\mathbf{v} = -\mathbf{a}+\mathbf{q} = -2\mathbf{a}$.

iv) For all $\mathbf{x} \in \pm\mathcal{B}_n \setminus \{\pm\mathbf{a}\}$, $f \in Aut(\mathcal{G}(M), \mathbf{0})$, $f(-\mathbf{x}) = -f(\mathbf{x})$.

We have $4\mathbf{x} \neq \mathbf{0}$, since it would be another nontrivial 4-cycle. Hence $\mathbf{x} \neq -\mathbf{x}$ and by item (i) $N(\mathbf{x}, -\mathbf{x}) = \{\mathbf{0}\}$. By Theorem A.2.3 we have $N(f(\mathbf{x}), f(-\mathbf{x})) = \{\mathbf{0}\}$, thus $f(\mathbf{x}) + f(-\mathbf{x}) = \mathbf{0}$.

v) For all $\mathbf{x} \in \pm\mathcal{B}_n$, $f \in Aut(\mathcal{G}(M), \mathbf{0})$, $f(-\mathbf{x}) = -f(\mathbf{x})$ and $f(2\mathbf{x}) = 2f(\mathbf{x})$.

First apply Lemma A.4.2 together item (iv) to f^{-1} to get $\forall f \in Aut(\mathcal{G}(M), \mathbf{0})$, $f(-\mathbf{a}) = -f(\mathbf{a})$. Then considering the automorphism f' defined by $f'(\mathbf{v}) = f(\mathbf{x}+\mathbf{v}) - f(\mathbf{x})$ like in the proof of Lemma A.2.7 we obtain that $f(2\mathbf{x}) = 2f(\mathbf{x})$.

vi) For all $f \in Aut(\mathcal{G}(M), \mathbf{0})$, $f(\pm\mathbf{a}) = \pm\mathbf{a}$.

From item (iii) we have $N(\mathbf{a}, -\mathbf{a}) = \{\mathbf{0}, \pm 2\mathbf{a}\}$ with $\mathbf{0} \neq \pm 2\mathbf{a}$. By Theorem A.2.3 we get $N(f(\mathbf{a}), f(-\mathbf{a})) = \{\mathbf{0}, f(\pm 2\mathbf{a})\}$ with $\mathbf{0} \neq f(\pm 2\mathbf{a})$. By items (i, ii, iii) we get $\{f(\mathbf{a}), f(-\mathbf{a})\} = \{\pm\mathbf{a}\}$.

vii) For all $f \in Aut(\mathcal{G}(M), \mathbf{0})$, $\mathbf{x}, \mathbf{y} \in \pm\mathcal{B}_n$, $f(\mathbf{x}+\mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$.

If $\mathbf{x} = \mathbf{y}$ it is item (v). Otherwise if neither of $\mathbf{x}, \mathbf{y}$ is in $\{\pm\mathbf{a}\}$ we proceed like the first step of the proof of Lemma A.2.7; from item (i) we get $N(\mathbf{x}, \mathbf{y}) = \{\mathbf{0}, \mathbf{x}+\mathbf{y}\}$, hence by Theorem A.2.3 $N(f(\mathbf{x}), f(\mathbf{y})) = \{f(\mathbf{0}), f(\mathbf{x}+\mathbf{y})\} = \{\mathbf{0}, f(\mathbf{x})+f(\mathbf{y})\}$. As $f(\mathbf{0}) = \mathbf{0}$ we have that $f(\mathbf{x}+\mathbf{y}) = f(\mathbf{x})+f(\mathbf{y})$. Now if some is in $\{\pm\mathbf{a}\}$, we assume without loss of generality that $\mathbf{y} = \mathbf{a}$ and $\mathbf{x} \notin \{\pm\mathbf{a}\}$. From item (ii) we have $N(\mathbf{a}, \mathbf{x}) \subseteq \{\mathbf{0}, \mathbf{a}+\mathbf{x}, 2\mathbf{a}\}$. And by Theorem A.2.3 that $N(f(\mathbf{a}), f(\mathbf{x})) \subseteq \{\mathbf{0}, f(\mathbf{a}+\mathbf{x}), f(2\mathbf{a})\}$. By item (vi) we have $N(f(\mathbf{a}), f(\mathbf{x})) \subseteq \{\mathbf{0}, f(\mathbf{a})+f(\mathbf{x}), 2f(\mathbf{a})\}$. As $f(2\mathbf{a}) = 2f(\mathbf{a})$ (item (v)) we have that $f(\mathbf{a}+\mathbf{x}) = f(\mathbf{a})+f(\mathbf{x})$.

viii) $Aut(\mathcal{G}(M)) = LAut(\mathcal{G}(M))$.

Apply Lemma A.2.8 to item (vii).

□

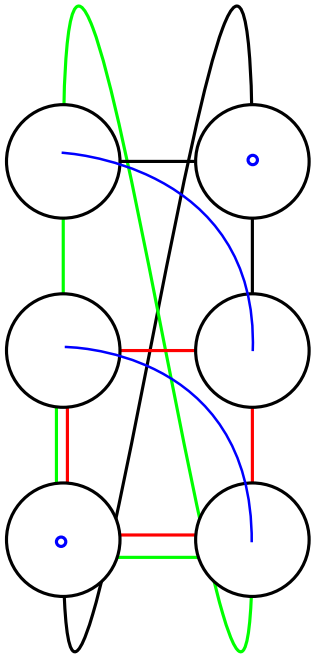


Figure A.2: A nonlinear automorphism of $\mathcal{G}(M)$, where $M = \begin{pmatrix} 2 & -1 \\ 0 & 3 \end{pmatrix}$.

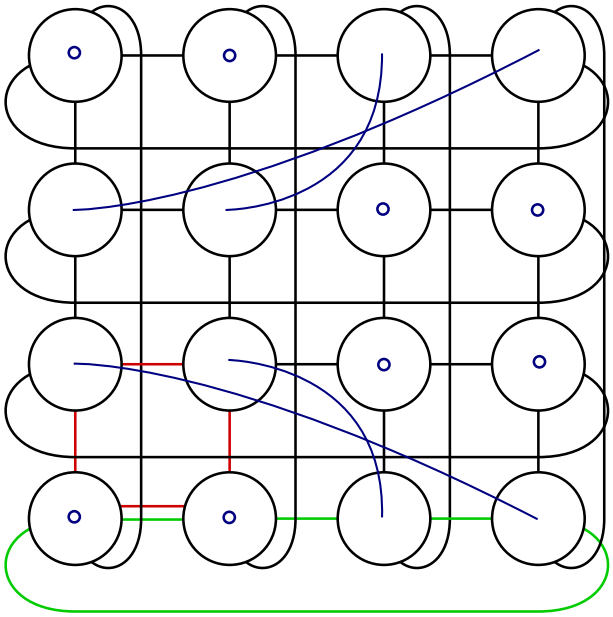


Figure A.3: A nonlinear automorphism of the square torus of side 4.

Finally, there are a few marginal cases in which the graph contains several nontrivial 4-cycles. These matrices are the matrices whose both columns correspond to nontrivial

4-cycles and their left divisors. These matrices can be built by selecting two columns in the set:

$$C = \left\{ \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 0 \\ 4 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -3 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix} \right\}.$$

A complete study of the following cases, shows that most of the combinations are edge-transitive. However, there are cases that lack of a nonlinear automorphism, leading to non-edge-transitive graphs.

Up to isomorphism, the bidimensional $\mathcal{G}(M)$ graphs with 2 different nontrivial solutions for 4-cycles are:

- With nontrivial 4 cycles but without nonlinear automorphisms.

$$\begin{pmatrix} 1 & 0 \\ 3 & 2 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 4 & 3 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 4 & 1 \\ 0 & 3 \end{pmatrix}$$

- With a nonlinear automorphism, which makes them edge-transitive,

$$\begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 3 & 3 \\ 1 & -1 \end{pmatrix} \cong \begin{pmatrix} 2 & -1 \\ 0 & 3 \end{pmatrix}, \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}$$

  with an example in Figure A.2. The first two have degree 3. Their associated lattice multigraphs do not have nonlinear automorphisms. In the figure, we show in blue a nonlinear automorphism involution, which fixes two vertices and maps the nontrivial green 4-cycle into the red 4-cycle.

- With a nonlinear automorphism, but their linear automorphisms already make them edge-transitive,

$$\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}, \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}, \begin{pmatrix} 3 & -1 \\ 1 & 3 \end{pmatrix}$$

  with the torus as example in Figure A.3.

A.5 Linearly Edge-Transitive Lattice Graphs of Dimension 3

This section provides a complete characterization of those lattice graphs generated by a matrix $M \in \mathbb{Z}^{3\times 3}$ being linearly edge-transitive.

Lemma A.5.1. *Given $M \in \mathbb{Z}^{3\times 3}$, $\mathcal{G}(M)$ is linearly edge-transitive if and only if there exists a signed permutation matrix of order 3 in $LAut(\mathcal{G}(M), \mathbf{0})$.*

Proof. If such a signed permutation exists, it is clear that $\mathcal{G}(M)$ is linearly edge-transitive.

For the reciprocal, by Theorem A.3.1 the automorphism is a signed permutation matrix. We can check that signed permutations matrices of dimension 3 can have orders 1, 2, 3, 4 and 6. The identity is the only signed permutation matrix of order 1 and it does not contribute to symmetry. Moreover, the signed permutation matrices which only change signs (that is, which are diagonal matrices) do no contribute to symmetry. Any remaining signed permutation matrix of orders 2 and 4 do not provide symmetry by themselves, since they fix one of the components, and the composition of two of them generates either a sign change or a signed permutation matrix of order 3 or 6.

Hence, linear edge-transitivity implies the existence of an automorphism $f \in LAut($ $\mathcal{G}(M), \mathbf{0})$ with order 3 or 6. If f has order 3 then it satisfies the condition. Otherwise, it satisfies $f^3 = -id$ and thus, $g = f^2$ has order 3. □

Hence, if $\mathcal{G}(M)$ is linearly edge-transitive then $LAut(\mathcal{G}(M), \mathbf{0})$ contains at least one of the next four cyclic groups as a subgroup and by Theorem A.3.2 there is a matrix P such that $PM = MQ$ for some Q.

$$P_1 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad P_2 = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$$
$$P_3 = \begin{pmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \qquad P_4 = \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

These signed permutation matrices have characteristic and minimum polynomial $\lambda^3 - 1$. We can find some matrices (symbolic over 3 integer parameters) whose lattice graphs are edge-transitive by taking $Q = P$, that is, we obtain M_i such that $P_i M_i = M_i P_i$. They are:

$$M_1 = \begin{pmatrix} a & c & b \\ b & a & c \\ c & b & a \end{pmatrix}, \qquad M_2 = \begin{pmatrix} a & -c & -b \\ b & a & -c \\ c & b & a \end{pmatrix},$$
$$M_3 = \begin{pmatrix} a & -c & -b \\ b & a & c \\ c & -b & a \end{pmatrix}, \qquad M_4 = \begin{pmatrix} a & c & b \\ b & a & -c \\ c & -b & a \end{pmatrix}.$$

Next, we find the similar matrices.

Lemma A.5.2. *There are exactly 2 similarity classes with characteristic polynomial $\lambda^3 - 1$:*

$$Q_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{pmatrix} \text{ and } Q_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{pmatrix}.$$

Proof. For $\lambda^3 - 1 = (\lambda - 1)(\lambda(\lambda + 1) + 1)$ we have the following upper triangular block matrix which has it as its characteristic polynomial: $Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{pmatrix}$. We know that

$$\begin{pmatrix} 1 & v & u \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & u+2v & u-v \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 1 & -v & -u \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{pmatrix}$$

So $\forall u, v \in \mathbb{Z}$, $\begin{pmatrix} 1 & u+2v & u-v \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{pmatrix}$. Since $|\det(\begin{pmatrix} -1 & -2 \\ -1 & 1 \end{pmatrix})| = 3$, by Theorem A.3.5, we have at most 3 matrices modulo similarity, which are:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 0 & 2 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{pmatrix}.$$

We check that the first two are non-similar. If

$$\begin{pmatrix}1&0&0\\0&-1&1\\0&-1&0\end{pmatrix}\begin{pmatrix}a&b&c\\d&e&f\\g&h&i\end{pmatrix}=\begin{pmatrix}a&b&c\\d&e&f\\g&h&i\end{pmatrix}\begin{pmatrix}1&0&1\\0&-1&1\\0&-1&0\end{pmatrix}$$

then

$$\begin{pmatrix}a&b&c\\-d+g&-e+h&-f+i\\-d&-e&-f\end{pmatrix}=\begin{pmatrix}a&-b-c&a+b\\d&-e-f&d+e\\g&-h-i&g+h\end{pmatrix}.$$

Hence $d=g=0$ and $a=-3b$; and $3b$ divides the determinant, which cannot be a unit. Now we see that the last two are similar.

$$\begin{pmatrix}1&0&1\\0&-1&1\\0&-1&0\end{pmatrix}\begin{pmatrix}1&0&1\\0&0&1\\0&-1&1\end{pmatrix}=\begin{pmatrix}1&0&1\\0&0&1\\0&-1&1\end{pmatrix}\begin{pmatrix}1&0&2\\0&-1&1\\0&-1&0\end{pmatrix}$$

So we have proved that there are exactly 2 similarity classes with characteristic polynomial λ^3-1:

$$Q_1=\begin{pmatrix}1&0&0\\0&-1&1\\0&-1&0\end{pmatrix}\text{ and }Q_2=\begin{pmatrix}1&0&1\\0&-1&1\\0&-1&0\end{pmatrix}.$$

□

Finally, we explore the $4\cdot 2=8$ possible matrices from all the combinations.

Lemma A.5.3. *With the previous definitions, $P_1\sim Q_2\sim P_2\sim P_3\sim P_4$.*

Proof. First we see that $P_1\sim Q_2$.

$$\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}\begin{pmatrix}1&0&0\\1&-1&1\\1&0&1\end{pmatrix}=\begin{pmatrix}1&0&0\\1&-1&1\\1&0&1\end{pmatrix}\begin{pmatrix}1&0&1\\0&-1&1\\0&-1&0\end{pmatrix}$$

And now that $P_1\sim P_2\sim P_3\sim P_4$.

$$\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}\begin{pmatrix}-1&0&0\\0&1&0\\0&0&-1\end{pmatrix}=\begin{pmatrix}-1&0&0\\0&1&0\\0&0&-1\end{pmatrix}\begin{pmatrix}0&0&1\\-1&0&0\\0&-1&0\end{pmatrix}$$

$$\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}\begin{pmatrix}1&0&0\\0&1&0\\0&0&-1\end{pmatrix}=\begin{pmatrix}1&0&0\\0&1&0\\0&0&-1\end{pmatrix}\begin{pmatrix}0&0&-1\\1&0&0\\0&-1&0\end{pmatrix}$$

$$\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}\begin{pmatrix}1&0&0\\0&-1&0\\0&0&-1\end{pmatrix}=\begin{pmatrix}1&0&0\\0&-1&0\\0&0&-1\end{pmatrix}\begin{pmatrix}0&0&-1\\-1&0&0\\0&1&0\end{pmatrix}$$

□

Thus, the first 4 matrices with $P_iM = MQ_2$ are right equivalent to the previously calculated M_i. Therefore, we find the 4 symbolic matrices M_i' which satisfy $P_iM_i' = M_i'Q_1$.

$$M_1' = \begin{pmatrix} a & b & c \\ a & c & -b-c \\ a & -b-c & b \end{pmatrix} \qquad M_2' = \begin{pmatrix} a & b & c \\ -a & -c & b+c \\ a & -b-c & b \end{pmatrix}$$
$$M_3' = \begin{pmatrix} a & b & c \\ a & c & -b-c \\ -a & b+c & -b \end{pmatrix} \qquad M_4' = \begin{pmatrix} a & b & c \\ -a & -c & b+c \\ -a & b+c & -b \end{pmatrix}$$

Now we have all the necessary elements to enunciate the tridimensional characterization of linearly edge-transitive graphs.

Theorem A.5.4. *Let $M \in \mathbb{Z}^{3\times 3}$ be non-singular. Then, $\mathcal{G}(M)$ is linearly edge-transitive if and only if it is isomorphic to $\mathcal{G}(M_1)$ or $\mathcal{G}(M_1')$, where:*

$$M_1 = \begin{pmatrix} a & c & b \\ b & a & c \\ c & b & a \end{pmatrix} \text{ or } M_1' = \begin{pmatrix} a & b & c \\ a & c & -b-c \\ a & -b-c & b \end{pmatrix}$$

for some $a, b, c \in \mathbb{Z}$.

Proof. Let $\mathcal{G}(M)$ be linearly edge-transitive with $M \in \mathbb{Z}^{3\times 3}$. By Lemma A.5.1, P must exist with $PM = MQ$ with $P \in \{P_1, P_2, P_3, P_4\}$. By Lemmas A.3.4 and A.5.2 there exist M' and Q with $M \cong M'$, $Q \in \{Q_1, Q_2\}$ and $PM' = M'Q$.

- If $Q = Q_2$, then by Lemma A.5.3 we know $M'' \in \{M_1, M_2, M_3, M_4\}$, with $PM'' = M''P$, $M'' \cong M$. Now we want to see that the matrices M_1, M_2, M_3 and M_4 generate the same set of matrices modulo graph-isomorphism. For each M_i we find a variable change and isomorphism from M_1 into M_i:

  $$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} M_1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} -a & c & b \\ b & -a & -c \\ c & -b & -a \end{pmatrix},$$

  which is M_4 giving a the value $-a$.

  $$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} M_1 \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} -a & c & -b \\ b & -a & c \\ -c & b & -a \end{pmatrix},$$

  which is M_2 giving a the value $-a$ and c the value $-c$.

  $$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} M_1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} a & c & -b \\ b & a & -c \\ -c & -b & a \end{pmatrix},$$

  which is M_3 giving c the value $-c$.

- If $Q = Q_1$, then by Lemma A.5.3 we know $M' \in \{M'_1, M'_2, M'_3, M'_4\}$. Now we want to see that the matrices M'_1, M'_2, M'_3 and M'_4 generate the same set of matrices modulo graph-isomorphism. For each M_i we find an isomorphism from M_1 into M_i; we do not need in this case variable changes:

$$M'_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} M'_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} M'_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} M'_4.$$

□

Bibliography

[AAC+10] Baba Arimilli, Ravi Arimilli, Vicente Chung, Scott Clark, Wolfgang Denzel, Ben Drerup, Torsten Hoefler, Jody Joyner, Jerry Lewis, Jian Li, Nan Ni, and Ram Rajamony. The PERCS high-performance interconnect. In *2010 18th IEEE Symposium on High Performance Interconnects*, pages 75–82, Washington, DC, USA, 2010. IEEE, IEEE Computer Society.

[AAK01] Rudolf Ahlswede, Harout K. Aydinian, and Levon H. Khachatrian. On perfect codes and related concepts. *Designs, Codes and Cryptography*, 22(3):221–237, 2001.

[AB03a] Bader F. AlBdaiwi and Bella Bose. On resource placements in 3D tori. *J. Parallel Distrib. Comput.*, 63(9):838–845, September 2003.

[AB03b] Bader F. AlBdaiwi and Bella Bose. Quasi-perfect Lee distance codes. *Information Theory, IEEE Transactions on*, 49(6):1535–1539, June 2003.

[AB05] Bader F. AlBdaiwi and Bella Bose. Quasi-perfect resource placements for two-dimensional toroidal networks. *J. Parallel Distrib. Comput.*, 65(7):815–831, July 2005.

[ABC+05] Narasimha R. Adiga, Matthias A. Blumrich, Dong Chen, Paul Coteus, Alan Gara, Mark E. Giampapa, Philip Heidelberger, Sarabjeet Singh, Burkhard D. Steinmacher-Burow, Todd Takken, Mickey Tsao, and Pavlos Vranas. Blue Gene/L torus interconnection network. *IBM Journal of Research and Development*, 49(2.3):265–276, March 2005.

[ABC+06] Krste Asanovic, Rastilav Bodík, Bryan Christopher Catanzaro, Joseph James Gebis, Parry Husbands, Kurt Keutzer, David A. Patterson, William Lester Plishker, John Shalf, Samuel Webb Williams, and Katherine A. Yelick. The landscape of parallel computing research: A view from Berkeley. Technical report, UCB/EECS-2006-183, 2006.

[ABD+09] Jung Ho Ahn, Nathan Binkert, Al Davis, Moray McLaren, and Robert S. Schreiber. HyperX: Topology, routing, and packaging of efficient large-scale networks. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, pages 1–11, New York, NY, USA, 2009. ACM.

[ABF12] Bader Albader, Bella Bose, and Mary Flahive. Efficient communication algorithms in hexagonal mesh interconnection networks. *Parallel and Distributed Systems, IEEE Transactions on*, 23(1):69–77, January 2012.

[Ádá67] A. Ádám. Research problem 2-10. *J. Combin. Theory*, 2(393):217, 1967. Edited by Alan J. Hoffman.

[ADH14] Carlos Araújo, Italo J. Dejter, and Peter Horak. A generalization of Lee codes. *Designs, Codes and Cryptography*, 70(1-2):77–90, 2014.

[AHM09] Bader F. AlBdaiwi, Peter Horak, and Lorenzo Milazzo. Enumerating and decoding perfect linear Lee codes. *Des. Codes Cryptography*, 52(2):155–162, 2009.

[AK89] Sheldon B. Akers and Balakrishnan Krishnamurthy. A group-theoretic model for symmetric interconnection networks. *IEEE Transactions on Computers*, 38:555–566, 1989.

[AS12] Helena Astola and Stanislav Stankovic. On the use of Lee-codes for constructing multiple-valued error-correcting decision diagrams. In *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*, pages 1–6. IEEE, 2012.

[ASK06] Tarun Agarwal, Amit Sharma, and Laxmikant V. Kalé. Topology-aware task mapping for reducing communication contention on large parallel machines. In *IPDPS*, 2006.

[ASK13] Jung Ho Ahn, Young Hoon Son, and John Kim. Scalable high-radix router microarchitecture using a network switch organization. *ACM Trans. Archit. Code Optim.*, 10(3):17:1–17:25, September 2013.

[ASS09] Yuichiro Ajima, Shinji Sumimoto, and Toshiyuki Shimizu. Tofu: A 6D mesh/torus interconnect for exascale computers. *Computer*, 42:36–40, 2009.

[Ast82] Jaakko Astola. An Elias-type bound for Lee codes over large alphabets and its application to perfect codes (corresp.). *Information Theory, IEEE Transactions on*, 28(1):111–113, January 1982.

[AT13] Helena Astola and Ioan Tabus. Bounds on the size of Lee-codes. In *Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium on*, pages 471–476, September 2013.

[BA84] Laxmi N. Bhuyan and Dharma P. Agrawal. Generalized hypercube and hyperbus structures for a computer network. *Computers, IEEE Transactions on*, C-33(4):323–333, April 1984.

[Bar64] Paul Baran. On distributed communications networks. *Communications Systems, IEEE Transactions on*, 12(1):1–9, March 1964.

[BB96] Myung M. Bae and Bella Bose. Resource placement in torus-based networks. In *Parallel Processing Symposium, 1996., Proceedings of IPPS '96, The 10th International*, pages 327–331, April 1996.

[BBC13] Dhananjay Brahme, Onkar Bhardwaj, and Vipin Chaudhary. SymSig: A low latency interconnection topology for HPC clusters. In *High Performance Computing (HiPC), 2013 20th International Conference on*, pages 462–471, December 2013.

[BBK+68] George H. Barnes, Richard M. Brown, Maso Kato, David J. Kuck, Daniel L. Slotnick, and Richard A. Stokes. The Illiac IV computer. *IEEE Transactions on Computers*, C-17(8):746–757, August 1968.

[Ber68] Elwyn R. Berlekamp. *Algebraic coding theory.* McGraw-Hill, 1968.

[BGJK11] Abhinav Bhatele, William D. Gropp, Nikhil Jain, and Laxmikant V. Kale. Avoiding hot-spots on two-level direct networks. In *High Performance Computing, Networking, Storage and Analysis (SC), 2011 International Conference for*, pages 1–11, New York, NY, USA, November 2011. ACM.

[BH14] Maciej Besta and Torsten Hoefler. Slim Fly: A cost effective low-diameter network topology. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '14, pages 348–359, Piscataway, NJ, USA, 2014. IEEE Press.

[Bha11] Abhinav Bhatele. Topology aware task mapping. In David Padua, editor, *Encyclopedia of Parallel Computing*, pages 2057–2062. Springer US, 2011.

[BHBA91] Ramón Beivide, Enrique Herrada, José L. Balcázar, and Agustin Arruabarrena. Optimal distance networks of low degree for parallel computers. *IEEE Trans. Comput.*, 40(10):1109–1124, 1991.

[BHS+95] David Bailey, Tim Harris, William Saphir, Rob van der Wijngaart, Alex Woo, and Maurice Yarrow. The NAS parallel benchmarks 2.0. Technical report, NAS-95-020, NASA Ames Research Center, 1995.

[BKM+10] Tom Budnik, Brant Knudson, Mark Megerian, Sam Miller, Mike Mundy, and Will Stockdell. Blue Gene/Q resource management architecture. In *Many-Task Computing on Grids and Supercomputers (MTAGS), 2010 IEEE Workshop on*, pages 1–5, November 2010.

[Bla09] Buddy Bland. Jaguar: Powering and cooling the beast. `http://www.cse.ohio-state.edu/~panda/875/class_slides/cray-jaguar.pdf`, 2009.

[Bok81] Shahid H. Bokhari. On the mapping problem. *IEEE Trans. Comput.*, 30(3):207–214, March 1981.

[Bro66] William G Brown. On graphs that do not contain a Thomsen graph. *Canad. Math. Bull*, 9(2):1–2, 1966.

[BW85] Francis T. Boesch and Jhing-Fa Wang. Reliable circulant networks with minimum transmission delay. *Circuits and Systems, IEEE Transactions on*, 32(12):1286–1291, December 1985.

[Cam10] Cristóbal Camarero Coterillo. Toroidal L-graphs and applications. Proyecto Fin de Carrera, Universidad de Cantabria., 2010.

[CBC+05] Paul Coteus, H. Randall Bickford, Thomas M. Cipolla, Paul G. Crumley, Alan Gara, Shawn A. Hall, Gerard V. Kopcsay, Alphonso P. Lanzetta, Lawrence S. Mok, Rick A. Rand, Richard A. Swetz, Todd Takken, Paul La Rocca, Christopher Marroquin, Philip R. Germann, and Mark J. Jeanson. Packaging the Blue Gene/L supercomputer. *IBM Journal of Research and Development*, 49(2/3):213–248, March 2005.

[CBGV97] Carmen Carrion, Ramón Beivide, José Ángel Gregorio, and Fernando Vallejo. A flow control mechanism to avoid message deadlock in k-ary n-cube networks. In *High-Performance Computing, 1997. Proceedings. Fourth International Conference on*, pages 322–329, December 1997.

[CCBB15] Emilio Castillo, Cristóbal Camarero, Ana Borrego, and Jose Luis Bosque. Financial applications on multi-CPU and multi-GPU architectures. *The Journal of Supercomputing*, 71(2):729–739, 2015.

[CCS+13] Emilio Castillo, Cristóbal Camarero, Esteban Stafford, Fernando Vallejo, Jose Luis Bosque, and Ramón Beivide. Advanced switching mechanisms for forthcoming on-chip networks. In *Digital System Design (DSD), 2013 Euromicro Conference on*, pages 598–605. IEEE, September 2013.

[CEH+11] Dong Chen, Noel A. Eisley, Philip Heidelberger, Robert M. Senger, Yutaka Sugawara, Sameer Kumar, Valentina Salapura, David L. Satterfield, Burkhard Steinmacher-Burow, and Jeffrey J. Parker. The IBM Blue Gene/Q interconnection network and message unit. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 1–10, New York, NY, USA, 2011. IEEE, ACM.

[CEH+12] Dong Chen, Noel Eisley, Philip Heidelberger, Sameer Kumar, Amith Mamidala, Fabrizio Petrini, Robert Senger, Yutaka Sugawara, Robert Walkup, Burkhard Steinmacher-Burow, Anamitra Choudhury, Yogish Sabharwal, Swati Singhal, and Jeffrey J. Parker. Looking under the hood of the IBM Blue Gene/Q network. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '12, pages 69:1–69:12, Los Alamitos, CA, USA, 2012. IEEE Computer Society Press.

[CHM+99] Jin-yi Cai, George Havas, Bernard Mans, Ajay Nerurkar, Jean-Pierre Seifert, and Igor Shparlinski. On routing in circulant graphs. In *COCOON*, pages 360–369, 1999.

[CMAPJ04] Sueli I. R. Costa, Marcelo Muniz, Edson Agustini, and Reginaldo Palazzo Jr. Graphs, tessellations, and perfect codes on flat tori. *Information Theory, IEEE Transactions on*, 50(10):2363–2377, October 2004.

[CMB10] Cristóbal Camarero, Carmen Martínez, and Ramón Beivide. Symmetric L-networks. In *2010 International Workshop on Optimal Network Topologies*, 2010.

[CMB11] Cristóbal Camarero, Carmen Martínez, and Ramón Beivide. Identifying codes over L-graphs. In *3rd International Castle Meeting on Coding Theory and Applications*, 3ICMTA, pages 81–87, Barcelona, SPAIN, 2011. UB Servei de Publicacions.

[CMB13] Cristóbal Camarero, Carmen Martínez, and Ramón Beivide. L-networks: A topological model for regular 2D interconnection networks. *Computers, IEEE Transactions on*, 62(7):1362–1375, July 2013.

[CMB14] Cristóbal Camarero, Carmen Martínez, and Ramón Beivide. Lattice graphs for high-scale interconnection topologies. *Parallel and Distributed Systems, IEEE Transactions on*, 2014.

[CMB15] Cristóbal Camarero, Carmen Martínez, and Ramón Beivide. Identifying codes of degree 4 Cayley graphs over Abelian groups. Accepted for publication in *Advances in Mathematics of Communications*, 2015.

[CMV+07] José M. Cámara, Miquel Moretó, Enrique Vallejo, Ramón Beivide, José Miguel-Alonso, Carmen Martínez, and Javier Navaridas. Mixed-radix twisted torus interconnection networks. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1–10, March 2007.

[CMV+10] Jose M. Cámara, Miquel Moretó, Enrique Vallejo, Ramón Beivide, José Miguel-Alonso, Carmen Martínez, and Javier Navaridas. Twisted torus topologies for enhanced interconnection networks. *IEEE Transactions on Parallel and Distributed Systems*, 21(12):1765–1778, December 2010.

[CO93] Israel Cidon and Yoram Ofek. MetaRing—a full-duplex ring with fairness and spatial reuse. *Communications, IEEE Transactions on*, 41(1):110 –120, January 1993.

[Coh07] Henri Cohen. *Number Theory: Volume I: Tools and Diophantine Equations*, volume 1. Springer, 2007.

[Col04] Phillip Colella. Defining software requirements for scientific computing. slide of 2004 presentation included in David Patterson's 2005 talk, 2004.

[Cra] Cray XE6 brochure. `http://www.cray.com/Products/XE/Technology.aspx`.

[CS11] Yawen Chen and Hong Shen. Embedding meshes and tori on double-loop networks of the same size. *IEEE Trans. Comput.*, 60(8):1157–1168, August 2011.

[CSV+12] Emilio Castillo, Esteban Stafford, Fernando Vallejo, Jose Luis Bosque, Carmen Martínez, Cristóbal Camarero, and Ramón Beivide. Study of fault tolerance for king topologies. In *jornadas sarteco*, September 2012.

[CVB14] Cristóbal Camarero, Enrique Vallejo, and Ramón Beivide. Topological characterization of Hamming and dragonfly networks and its implications on routing. *ACM Trans. Archit. Code Optim.*, 11(4):39:1–39:25, December 2014.

[CVM+13] Cristóbal Camarero, Enrique Vallejo, Carmen Martínez, Miquel Moretó, and Ramón Beivide. Task mapping in rectangular twisted tori. In *Proceedings of the High Performance Computing Symposium*, HPC '13, pages 15:1–15:11, San Diego, CA, USA, 2013. Society for Computer Simulation International.

[Del85] Charles Delorme. Grands graphes de degré et diamètre donnés. *European Journal of Combinatorics*, 6(4):291–302, 1985.

[DFM92] Charles Delorme, O. Favaron, and M. Mahéo. Isomorphisms of Cayley multigraphs of degree 4 on finite Abelian groups. *Eur. J. Comb.*, 13(1):59–61, 1992.

[DSV03] Giuliana Davidoff, Peter Sarnak, and Alain Valette. *Elementary number theory, group theory and Ramanujan graphs*, volume 55. Cambridge University Press, 2003.

[DT03] William Dally and Brian Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[DV12] Daniel Dadush and Santosh S. Vempala. Deterministic construction of an approximate M-ellipsoid and its applications to derandomizing lattice algorithms. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1445–1456. SIAM, 2012.

[DV13] Daniel Dadush and Santosh S. Vempala. Near-optimal deterministic algorithms for volume computation via M-ellipsoids. *Proceedings of the National Academy of Sciences*, 2013.

[Etz11] Tuvi Etzion. Product constructions for perfect Lee codes. *Information Theory, IEEE Transactions on*, 57(11):7473–7481, November 2011.

[EVY13] Tuvi Etzion, Alexander Vardy, and Eitan Yaakobi. Coding for the Lee and Manhattan metrics with weighing matrices. *Information Theory, IEEE Transactions on*, 59(10):6712–6723, October 2013.

[Ext] Extrae MPI profiling tool. `http://www.bsc.es/ssl/apps/performanceTools/`.

[FB10] Mary Flahive and Bella Bose. The topology of Gaussian and Eisenstein-Jacobi interconnection networks. *IEEE Trans. Parallel Distrib. Syst.*, 21(8):1132–1142, August 2010.

[FBR+12] Greg Faanes, Abdulla Bataineh, Duncan Roweth, Tom Court, Edwin Froese, Bob Alverson, Tim Johnson, Joe Kopnick, Mike Higgins, and James Reinhard. Cray Cascade: a scalable HPC system based on a dragonfly network. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '12, pages 1–9, Los Alamitos, CA, USA, November 2012. IEEE Computer Society Press.

[FG04] Yun Fan and Ying Gao. Codes over algebraic integer rings of cyclotomic fields. *IEEE Transactions on Information Theory*, 50(1):194–200, January 2004.

[Fio87] Miguel Angel Fiol. Congruences in $\mathbb{Z}^n$, finite Abelian groups and the Chinese remainder theorem. *Discrete Math.*, 67:101–105, October 1987.

[Fio95] Miguel Angel Fiol. On congruence in $\mathbb{Z}^n$ and the dimension of a multidimensional circulant. *Discrete Math*, 141:1–3, 1995.

[FJ91] G. David Forney Jr. Geometrically uniform codes. *Information Theory, IEEE Transactions on*, 37(5):1241–1260, September 1991.

[FYAV87] Miguel Angel Fiol, José Luis Andrés Yebra, Ignacio Alegre, and Mateo Valero. Discrete optimization problem in local networks and data alignment. *IEEE Trans. Comput.*, 36(6):702–713, 1987.

[GGI+05] Domingo Gómez, Jaime Gutierrez, Álvar Ibeas, Carmen Martínez, and Ramón Beivide. On finding a shortest path in circulant graphs with two jumps. In *COCOON*, pages 777–786, 2005.

[GMP98] Sylvain Gravier, Michel Mollard, and Charles Payan. On the non-existence of 3-dimensional tiling in the Lee metric. *European Journal of Combinatorics*, 19(5):567–572, 1998.

[GN92] Christopher J. Glass and Lionel M. Ni. The turn model for adaptive routing. In *Proceedings of the 19th Annual International Symposium on Computer Architecture*, ISCA '92, pages 278–287, New York, NY, USA, 1992. ACM.

[Gün81] Klaus D. Günther. Prevention of deadlocks in packet-switched data transport systems. *Communications, IEEE Transactions on*, 29(4):512–524, April 1981.

[GVB+12a] Marina García, Enrique Vallejo, Ramón Beivide, Miguel Odriozola, Cristóbal Camarero, Mateo Valero, Germán Rodríguez, Jesús Labarta, and Cyriel Minkenberg. Bubble flow control in high-radix hierarchical networks. In *jornadas sarteco*, September 2012.

[GVB+12b] Marina García, Enrique Vallejo, Ramón Beivide, Miguel Odriozola, Cristóbal Camarero, Mateo Valero, Germán Rodríguez, Jesús Labarta, and Cyriel Minkenberg. On-the-fly adaptive routing in high-radix hierarchical networks. In *The 41st International Conference on Parallel Processing (ICPP)*, pages 279–288, September 2012.

[GVB+13a] Marina García, Enrique Vallejo, Ramón Beivide, Mateo Valero, and Germán Rodríguez. OFAR-CM: Efficient dragonfly networks with simple congestion management. In *High-Performance Interconnects (HOTI), 2013 IEEE 21st Annual Symposium on*, pages 55–62, 2013.

[GVB+13b] Marina García, Enrique Vallejo, Ramón Beivide, Miguel Odriozola, Cristóbal Camarero, Mateo Valero, J. Labarta, and G. Rodríguez. Global misrouting policies in two-level hierarchical networks. In *Proceedings of the 2013 Interconnection Network Architecture: On-Chip, Multi-Chip*, IMA-OCMC '13, pages 13–16, New York, NY, USA, 2013. ACM.

[GVB+13c] Marina García, Enrique Vallejo, Ramón Beivide, Miguel Odriozola, and Mateo Valero. Efficient routing mechanisms for dragonfly networks. In *Parallel Processing (ICPP), 2013 42nd International Conference on*, pages 582–592, October 2013.

[GVB+15] Marina García, Enrique Vallejo, Ramón Beivide, Cristóbal Camarero, Mateo Valero, Germán Rodríguez, and Cyriel Minkenberg. On-the-fly adaptive routing for dragonfly interconnection networks. *The Journal of Supercomputing*, 71(3):1116–1142, 2015.

[GW70] Solomon W. Golomb and Lloyd R. Welch. Perfect codes in the Lee metric and the packing of polyominoes. *SIAM Journal on Applied Mathematics*, 18(2):302–317, 1970.

[HA06] Peter Horak and Bader F. Albdaiwi. Fast decoding of quasi-perfect Lee distance codes. *Des. Codes Cryptography*, 40(3):357–367, September 2006.

[HA12] Peter Horak and Bader F. AlBdaiwi. Diameter perfect Lee codes. *Information Theory, IEEE Transactions on*, 58(8):5490–5499, August 2012.

[Haz14] Raj Hazra. Accelerating insights... in the technical computing transformation, June 2014. keynote at the International Supercomputing Conference (ISC'14).

[HG14] Peter Horak and Otokar Grošek. A new approach towards the Golomb–Welch conjecture. *European Journal of Combinatorics*, 38:12–22, 2014.

[Hil84] Anthony J. W. Hilton. Hamiltonian decompositions of complete graphs. *Journal of Combinatorial Theory, Series B*, 36(2):125–134, 1984.

[Hor09a] Peter Horak. On perfect Lee codes. *Discrete Mathematics*, 309(18):5551–5561, 2009. Combinatorics 2006, A Meeting in Celebration of Pavol Hell's 60th Birthday (May 1–5, 2006).

[Hor09b] Peter Horak. Tilings in Lee metric. *European Journal of Combinatorics*, 30(2):480–489, 2009.

[HS60] Alan J. Hoffman and Robert R. Singleton. On Moore graphs with diameters 2 and 3. *IBM Journal of Research and Development*, 4(5):497–504, November 1960.

[Hub93] Klaus Huber. Codes over Eisenstein-Jacobi integers. *Finite fields: theory, applications, and algorithms (Las Vegas, NV, 1993)*, pages 165–179, 1993.

[Hub94] Klaus Huber. Codes over Gaussian integers. *IEEE Transactions on Information Theory*, 40(1):207–216, January 1994.

[Hun74] Thomas W. Hungerford. Algebra. 1974. *Grad. Texts in Math*, 1974.

[HW79] Godfrey Harold Hardy and Edward Maitland Wright. *An introduction to the theory of numbers*, volume 4. Oxford University Press, fourth edition, 1979.

[IEE89] IEEE standards for local area networks: Token ring access method and physical layer specifications. *IEEE Std 802.5-1989*, 1989.

[IEE91] IEEE standards for local and metropolitan area networks: Distributed queue dual bus (DQDB) subnetwork of a metropolitan area network (MAN). *IEEE Std 802.6-1990*, 1991.

[IK00] Wilfried Imrich and Sandi Klavžar. *Product Graphs: Structure and Recognition*. Wiley-Interscience, 2000.

[Jan73] Ted Janssen. *Crystallographic Groups*. American Elsevier, 1973.

[JKD09] Nan Jiang, John Kim, and William J Dally. Indirect adaptive routing on large scale interconnection networks. In *ISCA '09: 36th International Symposium on Computer Architecture*, pages 220–231, 2009.

[JSB10] Anxiao Jiang, Moshe Schwartz, and Jehoshua Bruck. Correcting charge-constrained errors in the rank-modulation scheme. *Information Theory, IEEE Transactions on*, 56(5):2112–2120, 2010.

[KDA07] John Kim, William J. Dally, and Dennis Abts. Flattened butterfly: a cost-efficient topology for high-radix networks. In *Proceedings of the 34th annual international symposium on Computer architecture*, ISCA '07, pages 126–137, New York, NY, USA, 2007. ACM.

[KDSA08] John Kim, William J. Dally, Steve Scott, and Dennis Abts. Technology-driven, highly-scalable dragonfly topology. In *Proceedings of the 35th Annual International Symposium on Computer Architecture*, pages 77–88. IEEE Computer Society, 2008.

[KDTG05] John Kim, William J. Dally, Brian Towles, and Amit K. Gupta. Microarchitecture of a high-radix router. In *Proceedings of the 32th annual international symposium on Computer architecture*, volume 33 of *ISCA '05*, pages 420–431. IEEE Computer Society, 2005.

[KK79] Parviz Kermani and Leonard Kleinrock. Virtual cut-through: a new computer communication switching technique. *Computer Networks*, 3(4):267–286, 1979.

[KL98] JunSeong Kim and David J. Lilja. Characterization of communication patterns in message-passing parallel scientific application programs. *Network-Based Parallel Computing Communication, Architecture, and Applications*, pages 202–216, 1998.

[KN84] Hironori Kasahara and Seinosuke Narita. Practical multiprocessor scheduling algorithms for efficient parallel processing. *IEEE Trans. Comput.*, 33(11):1023–1029, November 1984.

[KS96] Michael Kaib and Claus P. Schnorr. The generalized Gauss reduction algorithm. *J. Algorithms*, 21(3):565–578, 1996.

[Lee09] Ingyu Lee. Characterizing communication patterns of NAS-MPI benchmark programs. In *Southeastcon*, pages 158–163, 2009.

[Lei85] Charles E. Leiserson. Fat-trees: Universal networks for hardware-efficient supercomputing. *Computers, IEEE Transactions on*, C-34(10):892–901, October 1985.

[Lep81] Timo Lepistö. A modification of the Elias-bound and nonexistence theorems for perfect codes in the Lee-metric. *Information and Control*, 49(2):109–124, 1981.

[LH] Antoine Le Hyaric. Converting the NAS benchmarks from MPI to BSP. `http://www.ann.jussieu.fr/~lehyaric/NASfromMPItoBSP/`.

[LK12] Gary Lakner and Brant Knudson. *IBM System Blue Gene Solution: Blue Gene/Q Hardware Installation and Maintenance Guide*. IBM Redbooks, April 2012.

[LKF03] Laurence E. LaForge, Kirk F. Korver, and M. Sami Fadali. What designers of bus and network architectures should know about hypercubes. *Computers, IEEE Transactions on*, 52(4):525–544, April 2003.

[LPW+15] Xiang-Ke Liao, Zheng-Bin Pang, Ke-Fei Wang, Yu-Tong Lu, Min Xie, Jun Xia, De-Zun Dong, and Guang Suo. High performance interconnect network for Tianhe system. *Journal of Computer Science and Technology*, 30(2):259–272, 2015.

[LY10] Lin Liu and Yuanyuan Yang. Energy-aware routing in hybrid optical network-on-chip for future multi-processor system-on-chip. In *Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ANCS '10, pages 18:1–18:9, New York, NY, USA, 2010. ACM.

[Mar81] Alain J. Martin. The torus: An exercise in constructing a processing surface. *Proceedings of the VLSI Conference*, 1981.

[Mar07] Carmen Martínez. *Codes and Graphs over Complex Integer Rings*. PhD thesis, University of Cantabria, 2007.

[MBG07] Carmen Martínez, Ramón Beivide, and Ernst M. Gabidulin. Perfect codes for metrics induced by circulant graphs. *IEEE Transactions on Information Theory*, 53(9):3042–3052, 2007.

[MBG09] Carmen Martínez, Ramón Beivide, and Ernst M. Gabidulin. Perfect codes from Cayley graphs over Lipschitz integers. *Information Theory, IEEE Transactions on*, 55(8):3552–3562, August 2009.

[MBGG05] Carmen Martínez, Ramón Beivide, Jaime Gutierrez, and Ernst Gabidulin. On the perfect t-dominating set problem in circulant graphs and codes over Gaussian integers. In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pages 254–258, September 2005.

[MBS+08] Carmen Martínez, Ramón Beivide, Esteban Stafford, Miquel Moretó, and Ernst M. Gabidulin. Modeling toroidal networks with the Gaussian integers. *IEEE Transactions on Computers*, 57:1046–1056, 2008.

[MCB10] Carmen Martínez, Cristóbal Camarero, and Ramón Beivide. Perfect graph codes over two dimensional lattices. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1047–1051, June 2010.

[Mil12] James Milano. *IBM System Blue Gene Solution: Blue Gene/Q Hardware Installation and Maintenance Guide*. IBM Redbooks, April 2012. In progress.

[MMB06] Carmen Martínez, Miquel Moretó, and Ramón Beivide. A generalization of perfect Lee codes over Gaussian integers. In *Information Theory, 2006 IEEE International Symposium on*, pages 1070–1074, July 2006.

[Mol71] Emil Molnár. Sui mosaici dello spazio di dimensionen. *Atti Accad. Naz. Lincei, VIII. Ser., Rend., Cl. Sci. Fis. Mat. Nat*, 51:177–185, 1971.

[MŠ13] Mirka Miller and Jozef Širáň. Moore graphs and beyond: A survey of the degree/diameter problem (2nd ed). *The Electronic Journal of Combinatorics*, 5 2013.

[MSB+08] Carmen Martínez, Esteban Stafford, Ramón Beivide, Cristóbal Camarero, Fernando Vallejo, and Ernst Gabidulin. Graph-base metrics over QAM constellations. In *2008 IEEE International Symposium on Information Theory*, pages 2494–2498, July 2008.

[MSBG08] Carmen Martínez, Esteban Stafford, Ramón Beivide, and Ernst. M. Gabidulin. Modeling hexagonal constellations with Eisenstein-Jacobi graphs. *Probl. Inf. Transm.*, 44:1–11, March 2008.

[MŠŠ12] Heather Macbeth, Jana Šiagiová, and Jozef Širáň. Cayley graphs of given degree and diameter for cyclic, Abelian, and metacyclic groups. *Discrete Mathematics*, 312(1):94–99, 2012. Algebraic Graph Theory – A Volume Dedicated to Gert Sabidussi on the Occasion of His 80th Birthday.

[Mul82] Henry Martyn Mulder. Interval-regular graphs. *Discrete Mathematics*, 41(3):253–269, 1982.

[New72] Morris Newman. *Integral matrices*. Academic Press, New York,, 1972.

[NH08] Shigeto Nishimura and Toyokazu Hiramatsu. A generalization of the Lee distance and error correcting codes. *Discrete Applied Mathematics*, 156(5):588–595, 2008.

[Nie05] Michael A Nielsen. Introduction to expander graphs, 2005.

[NMPR11] Javier Navaridas, José Miguel-Alonso, Jose Antonio Pascual, and Francisco Javier Ridruejo. Simulating and evaluating interconnection networks with INSEE. *Simulation Modelling Practice and Theory*, 19(1):494–515, 2011.

[NS10] Michael L. Norman and Allan Snavely. Accelerating data-intensive science with Gordon and Dash. In *Proceedings of the 2010 TeraGrid Conference*, TG '10, pages 14:1–14:7, New York, NY, USA, 2010. ACM.

[PBB+10] Cheolmin Park, Roy Badeau, Larry Biro, Jonathan Chang, Tejpal Singh, Jim Vash, Bo Wang, and Tom Wang. A 1.2 TB/s on-chip ring interconnect for 45nm 8-core enterprise Xeon® processor. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pages 180–181, February 2010.

[PD01] Li-Shiuan Peh and William J. Dally. A delay model and speculative architecture for pipelined routers. In *Proceedings of the 7th International Symposium on High-Performance Computer Architecture*, HPCA '01, pages 255–266, Washington, DC, USA, 2001. IEEE Computer Society.

[Pea96] Barak A. Pearlmutter. Doing the twist: Diagonal meshes are isomorphic to twisted toroidal meshes. *IEEE Transactions on Computers*, 45:766–767, 1996.

[Pos75] Karel A. Post. Nonexistence theorems on perfect Lee codes over large alphabets. *Information and Control*, 29(4):369–380, 1975.

[PRG+14] Bogdan Prisacari, Germán Rodríguez, Marina García, Enrique Vallejo, Ramón Beivide, and Cyriel Minkenberg. Performance implications of remote-only load balancing under adversarial traffic in dragonflies. In *Proceedings of the 8th International Workshop on Interconnection Network Architecture: On-Chip, Multi-Chip*, INA-OCMC '14, pages 5:1–5:4, New York, NY, USA, 2014. ACM.

[QCMPJ13] Cátia Quilles Queiroz, Cristóbal Camarero, Carmen Martínez, and Reginaldo Palazzo Jr. Quasi-perfect codes from Cayley graphs over integer rings. *Information Theory, IEEE Transactions on*, 59(9):5905–5916, September 2013.

[QPJ10] Cátia Quilles Queiroz and Reginaldo Palazzo Jr. Quasi-perfect geometrically uniform codes derived from graphs over Gaussian integer rings. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1158–1162, June 2010.

[QPJ11] Cátia Quilles Queiroz and Reginaldo Palazzo Jr. Geometrically uniform quasi-perfect codes derived from graphs over integer rings. In *3rd International Castle Meeting on Coding Theory and Applications*, volume 5, page 237. Univ. Autònoma de Barcelona, 2011.

[Rie06] Rolf Riesen. Communication patterns. In *Proceedings of the 20th International Conference on Parallel and Distributed Processing*, IPDPS'06, pages 275–232, Washington, DC, USA, 2006. IEEE Computer Society.

[Rob96] Borut Robic. Optimal routing in 2-jump circulant networks. Technical report, University of Cambridge Computer Laboratory, TR397, 1996.

[RPM05] Francisco Javier Ridruejo Pérez and José Miguel-Alonso. INSEE: An interconnection network simulation and evaluation environment. In *Euro-Par 2005, Parallel Processing, 11th International Euro-Par Conference, Lisbon, Portugal*, pages 1014–1023. Springer, 2005.

[RS94] Ron M. Roth and Paul H. Siegel. Lee-metric BCH codes and their application to constrained and partial-response channels. *Information Theory, IEEE Transactions on*, 40(4):1083–1096, July 1994.

[Sam67] Pierre Samuel. *Algebraic Theory of Numbers*. Hermann, 1967.

[SBM62] Daniel L. Slotnick, W. Carl Borck, and Robert C. McReynolds. The SOLOMON computer. In *AFIPS '62 (Fall): Proceedings of the December 4-6, 1962, fall joint computer conference*, pages 97–107, New York, NY, USA, 1962. ACM.

[SBM+10] Esteban Stafford, Jose Luis Bosque, Carmen Martínez, Fernando Vallejo, Ramón Beivide, and Cristóbal Camarero. A first approach to king topologies for on-chip networks. In *Proceedings of the 16th international Euro-Par conference on Parallel processing: Part II*, Euro-Par'10, pages 428–439, Berlin, Heidelberg, 2010. Springer-Verlag.

[SCS+08] Larry Seiler, Doug Carmean, Eric Sprangle, Tom Forsyth, Michael Abrash, Pradeep Dubey, Stephen Junkins, Adam Lake, Jeremy Sugerman, Robert Cavin, Roger Espasa, Ed Grochowski, Toni Juan, and Pat Hanrahan. Larrabee: a many-core x86 architecture for visual computing. *ACM Trans. Graph.*, 27:18:1–18:15, August 2008.

[SCV+12] Esteban Stafford, Emilio Castillo, Fernando Vallejo, José Luis Bosque, Carmen Martínez, Cristóbal Camarero, and Ramón Beivide. King topologies for fault tolerance. In *High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on*, pages 608–616. IEEE, June 2012.

[Seq81] Carlo H. Sequin. Doubly twisted torus networks for VLSI processor arrays. In *ISCA '81: Proceedings of the 8th annual symposium on Computer Architecture*, pages 471–480, Los Alamitos, CA, USA, 1981. IEEE Computer Society Press.

[Sin05] Arjun Singh. *Load-Balanced Routing in Interconnection Networks*. PhD thesis, Stanford University, 2005.

[SKS+11] Balaram Sinharoy, Ronald N. Kalla, William J. Starke, Hung Le, Robert Cargnoni, James Van Norstrand, B. J. Ronchetti, J. Stuecheli, Jens Leenstra, G. L. Guthrie, D. Q. Nguyen, Bart Blaner, C. F. Marino, E. Retter, and Phillip G. Williams. IBM POWER7 multicore server processor. *IBM Journal of Research and Development*, 55(3):1:1–1:29, May–June 2011.

[Špa07] Simon Špacapan. Nonexistence of face-to-face four-dimensional tilings in the Lee metric. *European Journal of Combinatorics*, 28(1):127–133, 2007.

[TF88] Y. Tamir and G. L. Frazier. High-performance multi-queue buffers for VLSI communications switches. In *Proceedings of the 15th Annual International Symposium on Computer Architecture*, ISCA '88, pages 343–354, Los Alamitos, CA, USA, 1988. IEEE Computer Society Press.

[Tho79] Clark D. Thompson. Area–time complexity for VLSI. In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 81–88. ACM, 1979.

[TP94] K. Wendy Tang and Sanjay A. Padubidri. Diagonal and toroidal mesh networks. *IEEE Trans. Comput.*, 43(7):815–826, 1994.

[Val82] Leslie G. Valiant. A scheme for fast parallel communication. *SIAM Journal on Computing*, 11(2):350–361, 1982.

[Vet13] Tomáš Vetrík. Abelian Cayley graphs of given degree and diameter 2 and 3. *Graphs and Combinatorics*, pages 1–5, 2013.

[VMMB11] Enrique Vallejo, Miquel Moretó, Carmen Martínez, and Ramón Beivide. Peripheral twists for torus topologies with arbitrary aspect ratio. In *Actas XXII Jornadas de Paralelismo*, pages 421–426, 2011.

[WCP13] Ruisheng Wang, Lizhong Chen, and Timothy Mark Pinkston. Bubble coloring: Avoiding routing- and protocol-induced deadlocks with minimal virtual channel requirement. In *Proceedings of the 27th International ACM Conference on International Conference on Supercomputing*, ICS '13, pages 193–202, New York, NY, USA, 2013. ACM.

[Wei62] Paul M. Weichsel. The Kronecker product of graphs. *Proceedings of the American Mathematical Society*, 13(1):47–52, 1962.

[YCM06] Hao Yu, I-Hsin Chung, and José E. Moreira. Topology mapping for Blue Gene/L supercomputer. In *SC 2006 Conference, Proceedings of the ACM/IEEE*, November 2006.

[YFJ+01] Yulu Yang, Akira Funahashi, Akiya Jouraku, Hiroaki Nishi, Hideharu Amano, and Toshinori Sueyoshi. Recursive diagonal torus: An interconnection network for massively parallel computers. *IEEE Transactions on Parallel and Distributed Systems*, 12:701–715, 2001.